

**Козюра В. Д., Скачек Л. М., Ткач Ю. М.,
Хорошко В. О., Шелест М. Є., Мехед Д. Б.**

**ПРОЦЕСИ ТА ТЕХНОЛОГІЇ В
ІНФОРМАЦІЙНИХ СИСТЕМАХ**

Навчальний посібник

Чернігів
2020

УДК 004(075.8)

П 84

Рецензент:

В.М. Базилевич – завідувач кафедри інформаційних та комп'ютерних систем, к.е.н., доцент

О.В. Криворучко – д.т.н., проф. Київський національний торговельно-економічний університет

(Рішення вченої ради ЧНТУ №5 від 30.06.2020 року)

Козюра В. Д., Скачек Л. М., Ткач Ю. М., Хорошко В. О., Шелест М. Є., Мехед Д. Б.

П-84 Процеси та технології в інформаційних системах: навчальний посібник. – Ніжин: ФОП Лук'яненко В.В., ТПК «Орхідея», 2020. – 278 с.

ISBN 978-617-7609-58-1

У навчальному посібнику докладно розглянуті основні поняття інформаційного процесу, сучасні інформаційні системи, їх процеси і технології. Розкрито положення та методологічні принципи сучасних інформаційних систем управління. Проаналізовано принципи побудови інтегрованих корпоративних інформаційних систем. Визначено поняття життєвого циклу інформаційної системи і супроводжуючих його процесів. Розглянуто методи оцінки ефективності інформаційних систем.

Навчальний посібник містить контрольні запитання до кожного розділу та лабораторні роботи для самостійного виконання здобувачами вищої освіти і може бути використаний під час вивчення дисциплін, пов'язаних з інформаційними технологіями та їх керуванням. Запропонований матеріал може стати корисним також для широкого кола фахівців, що займаються впровадженням інформаційних систем на підприємствах. Зміст даного навчального посібника передбачає, що читач освоїв курси інформатики та інформаційних технологій, володіє теорією баз даних і має уявлення про економіку і управління підприємством.

УДК 004(075.8)

ISBN 978-617-7609-58-1

**Козюра В. Д., Скачек Л. М., Ткач Ю. М.,
Хорошко В. О., Шелест М. Є., Мехед Д. Б.**

ПРОЦЕСИ ТА ТЕХНОЛОГІЇ В ІНФОРМАЦІЙНИХ СИСТЕМАХ

Навчальний посібник

В авторській редакції

Відповідальний за випуск – *Лук'яненко В.В.*

Підписано до друку 27.11.2020 р.
Формат 60x 84/16. Папір офсетний. Друк числовий.
Гарнітура Times New Roman. Обл.-вид. арк. 18,35.
Ум. друк. арк. 16,16. Тираж 300 прим.
Зам. № 607.

Віддруковано з оригінал-макету замовника

Видавець - ФОП Лук'яненко В.В. ТПК «Орхідея»

*Свідоцтво про внесення суб'єкта видавничої справи
до державного реєстру видавців, виготівників
і розповсюджувачів видавничої продукції
серія ДК № 3020 від 02.11.2007 р.*

16600, Чернігівська обл., м. Ніжин, вул. Небесної Сотні, 13 а.
Тел.: 068 815 06 60
E-mail: holdingvv@gmail.com

ЗМІСТ

ВСТУП.....	6
РОЗДІЛ 1. ІНФОРМАЦІЙНІ СИСТЕМИ. ОСНОВНІ ПОНЯТТЯ ТА ВИЗНАЧЕННЯ.....	7
1.1. Інформаційні системи як складові системи управління.....	7
1.2. Корпоративні інформаційні системи.....	10
1.3. Класифікація ІС. Рівні ІС в організації.....	16
РОЗДІЛ 2. СТРУКТУРИ ІНФОРМАЦІЙНОГО ПРОЦЕСУ.....	19
2.1. Параметри інформаційного процесу і зв'язок між ними.....	19
2.2. Варіаційна матриця інформаційного процесу.....	21
2.3. Граф інформаційного процесу і його особливості.....	26
2.4. Паралельні форми інформаційного процесу.....	27
2.5. Введення у функціональне дослідження структури.....	31
РОЗДІЛ 3. ФОРМАЛІЗАЦІЯ ІНФОРМАЦІЙНОГО ПРОЦЕСУ.....	33
3.1. Неперервний інформаційний процес.....	34
3.2. Дискретний інформаційний процес.....	38
3.3. Матричне представлення перетворення інформаційної системи.....	39
РОЗДІЛ 4. VISUAL BASIC FOR APPLICATION.....	45
4.1. Загальний опис VBA.....	45
4.2. Процедури і функції VBA.....	47
4.3. Константи.....	49
4.4. Змінні.....	50
4.4.1. Типи змінних.....	51
4.4.2. Область видимості змінних.....	51
4.5. Масиви.....	52
4.6. Оператор присвоєння.....	53
4.6.1. Арифметичні вирази.....	54
4.6.2. Стандартні математичні функції.....	55
4.6.3. Нестандартні математичні функції.....	55
4.7. Введення і вивід інформації.....	56
РОЗДІЛ 5. СУБД MICROSOFT ACCESS.....	58
5.1 Об'єкти бази даних.....	59
5.2 Робота з таблицями.....	59
5.2.1 Створення таблиць за допомогою Майстра.....	60
5.2.2 Ключові поля.....	62
5.2.3 Режим таблиць.....	63
5.2.4 Введення зовнішніх таблиць.....	65
5.2.5 Міжтабличні зв'язки.....	66

5.3 Запити	68
5.3.1 Запити на вибірку	68
5.3.2 Запити з обчислювальними полями	70
5.3.3 Запити для впорядкування записів в таблиці	71
5.3.4 Групувальні запити	72
5.3.5 Запити з умовами вибору (параметрами)	73
5.3.6 Запити на створення таблиці	75
5.3.7 Перехресні запити	77
5.3.8 Запити на вилучення	78
5.3.9 Запити на оновлення записів у таблицях	79
5.3.10 Запити на додавання записів	80
5.3.11 Запити, що виконують вибірку записів, які повторюються	81
5.4 Форми в MS ACCESS	83
5.4.1 Створення нової форми	83
5.4.2 Режим Конструктора	84
5.5 Приклади роботи з MS ACCESS	89
РОЗДІЛ 6. ПРОГРАММИ, ТОПОЛОГІЇ ТА ТЕХНОЛОГІЇ	125
6.1 Розподілені програми	127
6.1.1. Основні програмні і апаратні компоненти мережі	128
6.1.2. Механізм взаємодії комп'ютера з периферійними пристроями	128
6.1.3. Найпростіший випадок взаємодії двох комп'ютерів	129
6.2 Еталонна модель взаємодії відкритих систем	132
6.3 Топології фізичних зв'язків	139
6.4 Базові технології локальних мереж	142
6.4.1. Технологія Ethernet	142
6.4.2. Технологія Token Ring	145
6.5 Протоколи TCP/IP та IP-адресування	147
6.5.1 Протокол IPv4, формат IP-адреси. Класи IP-адрес. Зарезервовані IP-адреси	147
6.5.2 Маска підмережі	149
6.5.3 Визначення адреси призначення пакету. Шлюз за замовчуванням	150
6.5.4 Встановлення і налаштування стеку протоколів TCP/IP на хості	151
6.5.5 Протокол IPv6, формат IP-адреси	152
6.5.6 Основні утиліти для конфігурування та діагностування TCP/IP	152
6.6 Мережа INTERNET	154
6.6.1. Основи Internet	154
6.6.2 Доступ до Internet	155
6.6.3 Розвиток засобів доступу	156

37. Лищук В.А. Математическая теория кровообращения / В.А. Лищук. — М.: Медицина, 1991. — 256с.

38. Саати Т. Принятие решений. Метод анализа иерархий / Т.Саати; пер.с англ. В.Н.Веселова. — М.: Радио и связь, 1993. — 278 с.

39. Гантмахер Ф.Р. Теория матриц / Ф.Р.Гантмахер. — М.: Наука, 1988. — 552 с.

40. Иванов Б.Н. Дискретная математика. Алгоритмы и программы / Б.Н.Иванов. — М.: Лаборатория Базовых Знаний, 2001. — 288с.

41. Грибунин В.Г. Цифровая стеганография / В.Г.Грибунин, И.Н.Оков, И.В.Туринцев. — М.: Солон-Пресс, 2002. — 272с.

42. Кобозева А.А. Анализ защищености информационных систем / А.А.Кобозева, І.О.Мачалін, В.О.Хорошко. - К.: Вид. ДУІКТ, 2010. – 316 с.

43. Кобозева А.А. Применение сингулярного и спектрального разложения матриц в стеганографических алгоритмах / А.А.Кобозева // Вісник Східноукр-го нац-го ун-ту ім. В.Дала. — 2006. — №9(103), ч.1. — С.74—82.

44. Конахович Г.Ф. Компьютерная стеганография. Теория и практика / Г.Ф.Конахович, А.Ю.Пузыренко. — К.: МК — Пресс, 2006. — 288 с.

45. Кобозева А.А. Учет свойств нормального спектрального разложения матрицы контейнера при обеспечении надежности восприятия стегосообщения / А.А.Кобозева, Е.А.Трифенова // Вестник НТУ «ХПИ». — 2007. — №18. — С.81—93.

46. Кобозева А.А. Оценка чувствительности стегосообщения к возмущающим воздействиям / А.А.Кобозева, Е.В.Нариманова // Системні дослідження та інформаційні технології. — 2008. — №3. — С. 52—65.

47. Топології комп'ютерних мереж. Базові технології локальних мереж: Інструкція до лабораторної роботи 1 із курсу "Основи телекомунікаційних систем" для студентів базового напрямку "Управління інформацією" / Укл. А.З. Піскозуб, І.Р.Опирський - Львів: Національний університет "Львівська політехніка", 2010. - 14 с.

48. Т-эффективные алгоритмы приближенного разв'язання задач обчислювальної та прикладної математики / [В.К. Задірака, М.Д. Бабич, А.І. Березовський та ін.]. — Київ, 2003 р. — 261 с.

49. Воеводин В.В. Вычислительные основы линейной алгебры / В.В.Воеводин. — М.: Наука. Гл.ред.физ.-мат.лит., 1977. — 304 с.

50. Свами М. Графы, сети и алгоритмы / М. Свами, К. Тхуласираман. — М.: Мир, 1984. — 456 с.

6.6.4 Служби Internet. Сумісність на рівні даних	162
6.7 Основи функціонування WWW.....	167
6.7.1 Основні поняття.....	167
6.7.2 Інфраструктура World Wide Web та інтерактивний перегляд	170
6.7.3 Подання документів	171
6.7.4. Взаємодія типу "клієнт-сервер"	171
6.7.5 Архітектура програмного забезпечення броузера	172
РОЗДІЛ 7. HTML	174
7.1. Основні терміни.....	174
7.2. Структура Web-сторінки.....	175
7.4. Розміщення тексту і заголовки.....	179
7.5. Управління кольором	181
7.6. Шрифти.....	182
7.7. Списки.....	183
7.8. Гіперпосилання	197
7.9. Фрейми.....	202
7.10. Відтворення звуку і рухомих об'єктів.....	207
РОЗДІЛ 8. ЛАБОРАТОРНИЙ ПРАКТИКУМ	209
8.1. Лабораторний практикум з VBA	209
8.2 Лабораторний практикум з СУБД MICROSOFT ACCESS	237
ЛІТЕРАТУРА:.....	274

ВСТУП

Сьогодні інформація розглядається як один з основних інструментів розвитку суспільства, а інформаційні системи і технології як засіб для підвищення продуктивності, ефективності та поліпшення якості роботи співробітників.

Інформаційні системи і технології широко використовуються в суспільстві в різних сферах: управлінської, фінансової, виробничої і т. д. У зв'язку з цим набуття знання та розуміння процесів та технологій в сучасних інформаційних системах, і технологій у професійній діяльності людини набуває важливого значення.

Під інформаційною технологією розуміється упорядкована сукупність формалізованих методів переробки, зміни стану, властивостей і якісної форми прояву інформації, а також методів тиражування, поширення, зберігання та використання інформації, що здійснюються в процесі цілеспрямованої суспільно виробничої діяльності [2].

Будь-який користувач повинен вміти отримувати доступ до необхідної йому інформації, а також переробити її в таку форму, з якої йому можна буде виконати свою професійну функцію, отже, необхідно навчати майбутніх фахівців використанню інформаційних систем і технологій вже в вищому навчальному закладі.

Процеси та технології в сучасних інформаційних системах постійно розвиваються, що викликає необхідність модернізації навчальних матеріалів необхідних для формування актуальних знань і умінь і в плані їх змістовної складової, і в плані використання засобів і методів навчання.

У навчальному посібнику розглянуті класифікація і структура інформаційних систем, інформаційні ресурси, технології та пов'язані з ними поняття, визначення і технології. Описано характеристики і базові функції сучасних інформаційних систем, а також викладена стратегія їх розвитку. Кожен розділ підручника включає контрольні питання для самостійної роботи, що дозволяють студенту закріпити вивчений матеріал.

Підручник може бути використаний студентами при вивченні дисциплін, пов'язаних з інформаційними технологіями та керуванням. Запропонований матеріал може стати корисним також для широкого кола фахівців, що займаються впровадженням інформаційних систем на підприємствах. Зміст даного посібника передбачає, що читач освоїв курси інформатики та інформаційних технологій, володіє теорією баз даних і має уявлення про економіку і управління підприємством.

20. Новиков Ф.А. Дискретная математика для программистов / Ф.А.Новиков. — СПб.: Питер, 2006. — 364 с.

21. Бахвалов Н.С. Численные методы / Н.С.Бахвалов, Н.П.Жидков, Г.М.Кобельков. — М.: БИНОМ. Лаборатория знаний, 2006. — 636 с.

22. Гонсалес Р. Цифровая обработка изображений / Р.Гонсалес, Р.Вудс; пер. с англ. под ред. П.А.Чочиа. — М.: Техносфера, 2005. — 1072 с.

23. Малюк А.А. Информационная безопасность: концептуальные и методологические основы защиты информации / Малюк А.А. — М.: Горячая линия — Телеком, 2004. — 280с.

24. Мамиконов А.Г. Достоверность, защита и резервирование информации в АСУ / А.Г.Мамиконов, В.В.Кульба, А.Б.Шелков. — М.: Энергоатомиздат, 1986. — 254 с.

25. Куприянов А.И. Основы защиты информации / А.И.Куприянов, А.В.Сахаров, В.А.Шевцов. — М.: Издательский центр «Академия», 2006. — 256 с.

26. Герасименко В.А. Основы защиты информации / В.А.Герасименко, А.А.Малюк. — М.: МИФИ, 1997. — 212 с.

27. Грушо А.А. Теоретические основы защиты информации / А.А.Грушо, Е.Е.Тимонина. — М.: Яхтмен, 1996. — 67 с.

28. Герасименко В.А. Защита информации в автоматизированных системах обработки данных / Герасименко В.А. — М.: Энергоатомиздат, 1994. — 453 с.

29. Нечеткие множества и теория возможностей / Под ред. Р.Р.Ячера. — М.: Радио и связь, 1986. — 408 с.

30. Танака К. Итоги рассмотрения факторов неопределенности и неясности в инженерном искусстве / Танака К. — М.: Радио и связь, 1986. — 408 с.

31. Жданов А.А. О методе автономного адаптивного управления. Лекции по нейроинформатике. Часть 2 / Жданов А.А. — М.: МИФИ, 2004. — 200 с.

32. Винер Н. Кибернетика или управление и связь в животном и в машине / Винер Н. — М.: Наука, 1983. — 344с.

33. Терейковский И.А. Нейронні мережі в засобах захисту комп'ютерної інформації / І.А.Терейковский. — К.: ТОВ «ПоліграфКонсалтинг», 2007. — 209с.

34. Маслов В.П. Асимптотические методы и теория возмущений / Маслов В.П. — М.: Наука. Гл.ред.физ.-мат.лит., 1988. — 312 с.

35. Каханер Д. Численные методы и программное обеспечение / Д.Каханер, К.Моулера, С.Нэш; пер. с англ. Х.Д.Икрамова. — М.: Мир, 2001. — 575 с.

36. Парлетт Б. Симметричная проблема собственных значений. Численные методы / Парлетт Б.; пер. с англ. Х.Д.Икрамова и Ю.А.Кузнецова. — М.: Мир, 1983. — 384с.

ЛІТЕРАТУРА:

1. Домарев В.В. Безопасность информационных технологий. Методология создания систем защиты / Домарев В.В. — Изд-во: ТИД «ДС», 2001. — 688с.
2. Домарев В.В. Безопасность информационных технологий. Системный подход / Домарев В.В. — Изд-во: ТИД «ДС», 2004. — 992с.
3. Хорошко В.А. Методы и средства защиты информации / В.А. Хорошко, А.А. Чекатков. — К.: Юниор, 2003. — 501 с.
4. Ленков С.В. Методы и средства защиты информации: в 2 т. / С.В. Ленков, Д.А. Перегудов, В.А. Хорошко. — К.: Арий, 2008 — . —
5. Т.2: Информационная безопасность. — 2008. — 344 с.
6. Хорошко В.А. Основы компьютерной стеганографии: навчальний посібник для студентів і аспірантів / В.А.Хорошко, О.Д.Азаров, М.С.Шелест, Ю.Є.Яремчук. — Вінниця: ВДТУ, 2003. —143 с.
7. Стрельцов А.А. Обеспечение информационной безопасности России. Теоретические и методологические основы / Стрельцов А.А., Под ред. В.А.Садовниченко и В.П.Шерстюка. — М.: МЦНМО, 2002. — 296 с.
8. Чумарин И.Г. Тайна предприятия: что и как защищать / Чумарин И.Г. — СПб.: Изд-во ДНК, 2001. — 72 с.
9. Хорев П.Б. Методы и средства защиты информации в компьютерных системах / Хорев П.Б. — Изд-во «Академия», 2005. — 256 с.
10. Хорев П.Б. Способы и средства защиты информации / Хорев П.Б. — М.: МО РФ, 2000. — 316 с.
11. Степанов Е.А. Информационная безопасность и защита информации / Е.А.Степанов, И.К.Корнеев. — М.: ИНФРА-М, 2001. — 303 с.
12. Кобозева А.А. Анализ информационной безопасности / А.А.Кобозева, В.А.Хорошко. — К.: Изд. ГУИКТ, 2009. — 251 с.
13. Круглов В.В. Нечеткая логика и искусственные нейронные сети: учеб.пособие для студ. вузов / В.В.Круглов,М.И. Дли, Р.Ю.Голунов. — М.:Физматлит, 2001. — 224с.
14. Фихтенгольц Г.М. Курс дифференциального и интегрального исчисления / Г.М. Фихтенгольц. — М.: Наука, 1969. — 608 с.
15. Деммель Дж. Вычислительная линейная алгебра / Дж.Деммель; пер.с англ. Х.Д.Икрамова. — М.: Мир, 2001. — 430 с.
16. Ломов С.А. Введение в общую теорию сингулярных возмущений / С.А.Ломов. — М.: Наука. Гл.ред.физ.-мат.лит., 1981. — 400с.
17. Воеводин В.В. Параллельные вычисления / В.В.Воеводин, В.Л.Воеводин. — СПб.: БХВ-Петербург, 2002. — 608 с.
18. Макконнелл Дж. Основы современных алгоритмов. 2-е дополненное издание / Дж. Макконнелл. — М.: Техносфера, 2004. — 368 с.
19. Джордж А. Численное решение больших разреженных систем уравнений / А.Джордж, Дж.Лю; пер. с англ. Х.Д.Икрамова. — М., Мир, 1984. — 333с.

РОЗДІЛ 1. ІНФОРМАЦІЙНІ СИСТЕМИ. ОСНОВНІ ПОНЯТТЯ ТА ВИЗНАЧЕННЯ

1.1. Інформаційні системи як складові системи управління

Управлінська інформаційна система, яку зазвичай називають УІС - це фраза, що складається з трьох слів: управління, інформація та система. Дивлячись на ці три слова, легко визначити інформаційні системи управління як системи, що надають інформацію управлінню. Це просте визначення УІС, яке загалом підсумовує, що таке управлінська інформаційна система та що вона повинна робити. Однак його роль та вплив на безперебійну роботу компанії ніколи не можна переоцінити. Саме тому кожна успішна компанія так чи інакше використовує ці системи.

Причина того, чому інформаційні системи управління є дуже важливими у повсякденній роботі компаній, полягає в тому, що ці системи працюють з людьми, організаціями, технологіями та взаємовідносинами між людьми та організаціями, що впливають на компанію. Це означає, що при належній реалізації інформаційні системи управління допомагають досягти високого рівня ефективності в управлінських операціях компанії.

Інформаційна система управління – це широкий термін, що включає багато спеціалізованих систем. Основні типи систем включають наступне:

- Виконавча інформаційна система. Керівники використовують виконавчу інформаційну систему для прийняття рішень, які впливають на всю організацію. Керівникам потрібні дані високого рівня з можливістю детального аналізу за необхідності.
- Маркетингова інформаційна система. Маркетингові групи використовують даний вид систем управління для звітування про ефективність минулих та поточних кампаній та використовують отримані дані для планування майбутніх кампаній.
- Система бізнес-аналітики. Використовуються в організаціях для прийняття управлінських та стратегічних рішень на основі збору, інтеграції та аналізу зібраних даних та інформації. Ця система схожа на виконавчу інформаційну систему, але нею користуються як менеджери нижчого рівня, так і керівники.
- Система управління взаємовідносинами з клієнтами. Дана система зберігає ключову інформацію про клієнтів, включаючи попередні продажі, контактну інформацію та можливості продаж. Команди маркетингу, обслуговування клієнтів, продаж та розвитку бізнесу часто використовують даний вид інформаційних систем.
- Система автоматизації продаж. Спеціалізований компонент систем управління взаємовідносинами з клієнтами, який автоматизує багато завдань, які виконує команда продаж. Це може включати управління контактами, відстеження та генерацію потенційних клієнтів та управління замовленнями.

- Система обробки транзакцій. Інформаційна система, яка завершує продаж та управляє супутніми деталями. На базовому рівні система обробки транзакцій може бути системою торгових точок або системою, яка дозволяє подорожуючим шукати готель та вивчати варіанти номерів, такі як ціновий діапазон, тип та кількість ліжок та ін., а потім дає можливість вибрати та забронювати його. Співробітники можуть використовувати створені дані для звітування про тенденції використання та відстеження продажів з часом.
- Система управління знаннями - служба обслуговування клієнтів яка може використовувати систему управління для відповіді на запитання та вирішення проблем.
- Система фінансового обліку: Ця інформаційна система характерна для підрозділів, що займаються фінансами та бухгалтерським обліком, таких як кредиторська та дебіторська заборгованість.
- Система управління людськими ресурсами. Ця система відстежує результати роботи працівників та дані про заробітну плату.
- Система управління ланцюгами поставок. Виробничі компанії використовують даний вид інформаційних систем для відстеження потоку ресурсів, матеріалів та послуг від закупівлі до відвантаження кінцевої продукції.

Типи звітів інформаційних систем управління.

За своєю суттю існують інформаційні системи управління для зберігання даних та створення звітів, які професіонали бізнесу можуть використовувати для аналізу та прийняття рішень. Існує три основних типи звітів:

За розкладом. Ці звіти, що створюються регулярно, використовують правила, надані запитувачем, для збирання та упорядкування даних. Заплановані звіти дозволяють компаніям аналізувати дані протягом часу (наприклад, авіакомпанія може бачити відсоток втраченого багажу за місяцями), місцезнаходження (наприклад, торгова мережа може порівнювати показники продажів з різних магазинів) або інші параметри.

Довільні. Це одноразові звіти, які користувач створює для відповіді на запитання. Якщо звіти корисні, користувач може перетворити спеціальні звіти в заплановані звіти.

В режимі реального часу. Цей тип звіту управлінських інформаційних систем дозволяє користувачу відстежувати зміни, коли вони відбуваються. Наприклад, менеджер колл-центру може побачити несподіваний стрибок обсягу дзвінків і знайти спосіб підвищити продуктивність або надіслати деякі дзвінки в інше місце.

Окрім потреби залишатися конкурентоспроможною, є кілька ключових переваг ефективного використання інформаційних систем управління:

1. Керівництво може отримати звіт своєї діяльності.

Лабораторна робота 13. HTML. Створення Web-сторінки приватного підприємства.

Виконати в лабораторії.

Проект Web-сторінки приватного підприємства, створити логотип (графічний об'єкт).

Розробити структуру Web-сторінки приватного підприємства.

Розробити файли підтримки фреймової структури Web-сторінки приватного підприємства.

Застосувати відомі Вам анімаційні ефекти.

Переслати електронною поштою HTML-код створеної Web-сторінки приватного підприємства викладачеві.

Лабораторна робота 10. HTML. Гіперпосилання.

Перед виконанням лабораторної роботи необхідно вивчити такі питання.

Формати гіперпосилань.

Файли підтримки гіперпосилань.

Виконати в лабораторії.

Виконати індивідуальне завдання.

Контрольні запитання

Яка структура дескриптора, призначеного для формування гіперпосилання у вигляді тексту?

Яка структура дескриптора, призначеного для формування гіперпосилання у вигляді графічного об'єкту?

Де зберігаються файли підтримки гіперпосилань?

Які типи файлів використовуються у гіперпосиланнях у вигляді графічних об'єктів?

Лабораторна робота 11. HTML. Фрейми.

Перед виконанням лабораторної роботи необхідно вивчити такі питання.

Створення фреймів.

Створення Web-документу з фреймами.

Виконати в лабораторії.

Виконати індивідуальне завдання.

Контрольні запитання

Який формат запису дескриптора, призначеного для формування фреймової структури Web-документу?

Як формуються файли підтримки фреймової структури Web-документу?

Де зберігаються файли підтримки фреймової структури Web-документу?

Як організовується фреймова структура Web-документу?

Лабораторна робота 12. Створення індивідуальної Web-сторінки студента.

Виконати в лабораторії.

Проект особистої Web-сторінки.

Розробити структуру особистої Web-сторінки.

Розробити файли підтримки фреймової структури особистої Web-сторінки.

Застосувати відомі Вам анімаційні ефекти.

Переслати електронною поштою HTML-код Вашої Web-сторінки викладачеві.

2. Менеджери мають можливість отримувати відгуки про свої результати.
3. Організації можуть максимізувати вигоди від своїх інвестицій, побачивши, що працює, а що ні.
4. Менеджери можуть порівняти результати із запланованими показниками, визначивши сильні та слабкі сторони як плану, так і результативності.
5. Компанії можуть сприяти вдосконаленню робочого процесу, що призводить до кращого узгодження бізнес-процесів із потребами клієнтів.
6. Багато бізнес-рішень переміщуються з вищого керівництва на рівні організації, які ближче до місця знань та досвіду.

Останні розробки та майбутнє інформаційних систем управління

Інформаційні системи управління, як і будь-яка *ifkerm*, що включає комп'ютери та програмне забезпечення, постійно змінюється. Останні події в цій галузі включають наступне:

1. ПК можуть тепер розміщувати ІСУ: малий бізнес може отримати доступ до потужного програмного забезпечення, яке раніше було доступне лише для великих підприємств.
2. Постачальники послуг з управління програмами та управліннями: подібно до оренди хмарних сховищ, компанії можуть орендувати пакети програмного забезпечення та послуги з управління системами та розширюватись у міру зміни їх потреб.
3. Безпека: як було доведено нещодавніми порушеннями даних, безпека даних перейшла від другорядного до головного.

У майбутньому багато тих самих сил, які змінять більший світ, впливатимуть на ІСУ, але деякі матимуть більший вплив, ніж інші. Експерти ІСУ зважують тему та те, що ми можемо очікувати в майбутньому:

Однією з великих галузей розвитку інформаційних технологій є штучний інтелект (ШІ), який виходить далеко за межі роботів, які контролюють виробництво (наприклад, в автомобільній промисловості). Машини стають розумнішими в тому, що вони можуть навчитися вирішувати проблеми. Однією з таких систем є нейронна мережа, яка використовується для попередження про те, що ваша кредитна картка могла використовуватися незаконно. Ці нейронні мережі формують схему ваших витрат і, виходячи з цього, вони позначають покупки, для яких відсутні певні характерні риси, тобто коли ви отримуєте сповіщення або заморожуєте вашу кредитну картку, залежно від ситуації. Такі події, безсумнівно, впливають на ІСУ, але вони також впливають на культуру, закон, медицину, військову оборону тощо.

Оскільки на сьогоднішній день збирається та аналізується стільки великих даних, існує велика потреба в юридичному регулюванні галузі, яке допоможе вирішувати різні питання щодо того, що повинно, а що не повинно бути законним з точки зору конфіденційності.

Штучний інтелект: вузький ШІ (ШІ для конкретних завдань) зараз поширений у багатьох організаціях. Досягнення машинного навчання та глибокого навчання роблять вузький ШІ набагато ціннішим для всіх нас. Відділи ІСУ повинні намагатися не відставати від цих досягнень і вирішити, наскільки вузький ШІ можна використовувати в їх організаціях.

Інтернет речей (IoT): Швидке збільшення розміщення датчиків на всіх об'єктах (живих та неживих) веде до створення середовища відчуття та реагування. Співробітники ІСУ повинні проводити SWOT-аналіз IoT для своїх організацій.

Блокчейн: технологія розподіленої книги зараз використовується у різних галузях. Знову ж таки, співробітники МІС повинні йти в ногу з цією технологією та бачити, як вона впливає на їх організацію.

Фінансові технології (FinTech): Якщо ваша організація перебуває у фінансовому секторі, вашим працівникам ІСУ краще стежити за стартовими компаніями FinTech. Ці компанії планують зруйнувати традиційний фінансовий сектор.

Квантові обчислення: Коли закон Мура починає сповільнюватися, коли ми досягаємо фізичних меж кількості інтегральних схем, які ми можемо розмістити на мікросхемі, виникає нова парадигма, яка називається квантовими обчисленнями. У класичних обчисленнях використовуються біти, які мають або «0», або «1». Квантові обчислення використовують квантові біти (кубіти). На відміну від класичних бітів, кубіти можуть зберігати набагато більше інформації, ніж просто 1 або 0, оскільки вони можуть існувати в будь-якій суперпозиції цих значень. Квантові обчислення знаходяться в самих ранніх днях, але їх потенціал може забезпечити різке збільшення обчислювальних швидкостей. Наприклад, вчені сподіваються, що зможуть точно моделювати клімат. Ще одна програма лежить в галузі інформаційної безпеки.

1.2. Корпоративні інформаційні системи

Традиційно корпоративні інформаційні системи (ІС) будувались для підтримки конкретних ділових функцій, і тому їх слід було розробляти у повному обсязі з функціональним розподілом організації. Це призвело до того, що в рамках однієї організації було розгорнуто широкий спектр систем, кожна з яких покладається на різні технології, що слугують певній меті в межах певної функціональної галузі (наприклад, логістика, людські ресурси, управління відносинами з клієнтами, управління ланцюгами поставок тощо), і не обов'язковим є "спілкування" із сусідніми програмами. Ранні системи часто було важко налаштувати та / або розширити, і було практично неможливо замінити або розвинути їх частини. Їх залежність від платформи та технології призвела до зобов'язання організацій приймати довгострокові технологічні рішення. У випадках, коли необхідний обмін інформацією між системами, доводилося домагатися тісного зв'язку. Розробка інтерфейсів між такими системами часто була пов'язана зі значними технічними проблемами.

Виконати індивідуальне завдання.

Контрольні запитання

Яка структура Web-документу?

Які основні дескриптори призначені для форматування тексту?

За допомогою яких дескрипторів впроваджуються символи у Web-документі?

Які основні дескриптори призначені для створення і форматування ліній?

Які дескриптори призначені для кольорового оформлення тла Web-документу?

Які засоби відтворення Web-документу у вікні броузера Ви знаєте?

У чому полягає принцип формування кольорів з використанням RGB-коду?

Лабораторна робота 8. HTML. Створення і редагування списків.

Перед виконанням лабораторної роботи необхідно вивчити такі питання.

Марковані списки.

Нумеровані списки.

Списки складної структури.

Виконати в лабораторії.

Виконати індивідуальне завдання.

Контрольні запитання

Які дескриптори призначені для створення маркованих списків?

Як редагується програмний код Web-документу?

Які дескриптори призначені для створення нумерованих списків?

Як формується програмний код Web-документу зі списками складної структури?

Лабораторна робота 9. HTML. Створення таблиць.

Перед виконанням лабораторної роботи необхідно вивчити такі питання.

Створення простих таблиць.

Створення складних таблиць.

Виконати в лабораторії.

Виконати індивідуальне завдання.

Контрольні запитання

Які дескриптори для формування рядків таблиці Ви знаєте?

Які дескриптори для формування стовпців таблиці Ви знаєте?

Як задати ширину стовпця таблиці?

Як об'єднати рядки таблиці?

Як об'єднати стовпці таблиці?

Як створити рамку довкола таблиці?

Як зберегти Web-сторінку на комп'ютері користувача?
Що розуміють під пошуковою системою в мережі Internet?
Які пошукові системи Ви знаєте?
Як класифікують пошукові системи?
Які особливості використання пошукової системи Google?

Лабораторна робота 6. Робота з електронною поштою.

Перед виконанням лабораторної роботи необхідно вивчити такі питання.

Інтерфейс програми Outlook Express.

Засоби обробки повідомлень програми Outlook Express, поштове відділення, поштова скринька, адресна книга, особиста папка, конфігурація користувача.

Формування повідомлення з використанням прикладних програм інтегрованого пакету Microsoft Office.

Виконати в лабораторії.

Налаштуйте параметри вікна програми Outlook Express.

Створіть свій ідентифікаційний запис.

Підготуйте повідомлення.

Вишліть своє повідомлення одному з товаришів у Вашій групі.

Виконати індивідуальне завдання.

Контрольні запитання

Які основні елементи графічного інтерфейсу програми Outlook Express Ви знаєте?

Назвіть призначення пунктів головного меню програми Outlook Express.

За допомогою якої команди налагоджують графічний інтерфейс програми Outlook Express?

За допомогою якої команди налагоджують властивості програми Outlook Express?

Як створити запис у адресній книзі?

Як створити групи розсилань?

Які особливості використання програми Outlook Express для пересилання графічних об'єктів?

Лабораторна робота 7. Основи HTML. Робота з текстом.

Перед виконанням лабораторної роботи необхідно вивчити такі питання.

HTML. Форматування тексту і символів.

HTML. Елементи ліній. Управління кольором.

HTML. Відтворення та редагування тексту.

Створення простої Web-сторінки.

Виконати в лабораторії.

На сьогодні в більшості бізнес-секторів посилена конкуренція на ринку, а очікування споживачів створюють зростаючу потребу в:

- обслуговування клієнтів з доданою вартістю та якістю обслуговування за кількома каналами доставки;
- скорочення часу виходу на ринок нових товарів та послуг;
- стримування операційних витрат (наприклад, на технічне обслуговування та експлуатацію ІБ) та операційної неефективності (наприклад, шляхом внутрішньої реконструкції);
- швидке реагування на стратегічні пріоритети бізнесу.

Адаптація корпоративних структур та методів роботи до постійно мінливого ділового середовища, включаючи реінжиніринг у відповідь на масштабні стратегічні рішення (наприклад, злиття та поглинання).

Корпоративна інформаційна система (КІС) – це інформаційна система, яка підтримує функції управління на підприємстві (у корпорації) і постачає інформацію для підтримки прийняття управлінських рішень. У ній реалізована управлінська ідеологія, яка об'єднує бізнес-стратегію підприємства і прогресивні інформаційні технології.

Сучасним КІС притаманні такі основні характеристики.

Масштабованість. Це одна із важливих характеристик інформаційних систем такого класу, враховуючи масштаби діяльності корпорації. Масштабована ІС повинна функціонувати на програмно-апаратній платформі (сервери, ОС, системи телекомунікації, СУБД), яка надає можливість нарощування і конфігурування нових потужностей. Оскільки варіантів конфігурування базового устаткування і програмного забезпечення може бути багато, тому КІС має бути мультиплатформною.

Мультиплатформне забезпечення. У КІС виникає потреба в тому, щоб прикладна програма працювала на кількох платформах. При цьому мають бути забезпечені: однакові інтерфейс і логіка роботи на всіх платформах, маючи на увазі подібність схем екрана, елементів меню і діалогової інформації, що надається користувачеві різними платформами; інтегрованість з користувацьким операційним середовищем; однакова поведінка на різних операційних платформах; узгоджена підтримка незалежно від платформи тощо. Реалізувати прикладну програму одночасно у кількох середовищах нелегко. Тому, з'явилися інтегровані програмні середовища (frameworks), які значно полегшують перенесення прикладних програм між різними середовищами. До них належать Windows Open Systems Architecture (WOSA); загальне відкрите програмне середовище UNIX COSE і App Ware Foundation тощо.

Робота в неоднорідному обчислювальному середовищі. Важливою перевагою КІС є можливість роботи у КМ, до яких входять робочі станції (WS), що працюють під управлінням різних операційних систем або побудовані на різних обчислювальних платформах. При цьому має бути забезпечена взаємодія всіх робочих обчислювальних платформ і операційних систем, які використовуються.

Розподілені обчислення. Це один із видів роботи в клієнт-серверній архітектурі, коли дані або запити вислані з клієнтських WS розподіляються поміж кількома WS, наприклад між кількома серверами, що збільшує пропускну спроможність для користувача і дає можливість багатозадачної роботи. Це сприяє максимальному використанню обчислювальних ресурсів, зниженню витрат і підвищенню ефективності системи. Забезпечення розподіленої роботи і віддаленого доступу до документів – це обов'язкова вимога до ІС корпоративного рівня. Останніми роками невід'ємною складовою частиною цієї вимоги стала підтримка роботи в архітектурі Internet/Intranet.

Види корпоративних інформаційних систем

Корпоративні інформаційні системи поділяються на наступні класи:

ERP (Enterprise Resource Planning System) планування ресурсів підприємства (рис. 1.4.).

Сучасні ERP з'явилися у результаті майже сорокарічної еволюції управлінських та інформаційних технологій [2]. Призначені вони, головним чином, для побудови єдиного інформаційного простору підприємства (об'єднання всіх відділів і функцій), ефективного управління всіма ресурсами, пов'язаними з продажами, виробництвом, обліком замовлень. Будується ERP-система за модульним принципом і, як правило, включає у себе модуль безпеки для запобігання як внутрішніх, так і зовнішніх крадіжок інформації. Проблеми ж виникають в основному через неправильність роботи або побудови неефективного плану впровадження системи. Наприклад, урізані інвестиції у навчання персоналу для роботи у системі суттєво знижують ефективність. Тому, впроваджують ERP-системи, як правило, не відразу в повному обсязі, а окремими модулями (особливо на початковій стадії).

Функціональний склад ERP



Рис. 1.4. Функціональний склад ERP.

Порівняйте переваги та недоліки ширококомовного broadcast трафіку і unicast трафіку.

Що таке MAC-адреса? Вкажіть формат запису MAC-адреси.

Лабораторна робота 4. Програма-планувальник Microsoft Outlook.

Перед виконанням лабораторної роботи необхідно вивчити такі питання.

Технологія планування робочого часу у вікні "Календарь".

Технологія створення та керування списком контактів у вікні "Контакты".

Технологія планування і керування проектами та задачами у вікні "Задача".

Виконати в лабораторії.

Налаштуйте параметри контактів.

Створіть ділові контакти.

Призначте на встановлену дату ділову зустріч.

Створіть у щоденнику запис про ділову зустріч.

Контрольні запитання

Яке призначення програми Microsoft Outlook?

Що таке "Календарь" і які події фіксуються у вікні "Календарь"?

Як здійснюється планування робочого часу у вікні "Календарь"?

У чому полягає принцип керування об'єктами у вікні?

Лабораторна робота 5. Робота у мережі Internet.

Перед виконанням лабораторної роботи необхідно вивчити такі питання.

Робота з браузером Microsoft Internet Explorer.

Цілеспрямований пошук у Internet.

Робота з Web-сторінками.

Виконати в лабораторії.

Налаштуйте параметри вікна Internet Explorer.

У полі Адреса вікна Internet Explorer введіть адресу Web-сторінки Львівського державного університету внутрішніх справ (www.livs.lviv.ua).

Уважно ознайомтеся зі сторінкою університету, зробіть її стартовою.

Виконати індивідуальне завдання.

Контрольні запитання

Які основні елементи графічного інтерфейсу програми Internet Explorer Ви знаєте?

Назвіть призначення пунктів головного меню програми Internet Explorer.

За допомогою якої команди налагоджують графічний інтерфейс програми Internet Explorer?

За допомогою якої команди налагоджують властивості програми Internet Explorer?

Для чого використовується кросове обладнання? Яка відмінність між кабелями і патчкордами?

Які типи конекторів для UTP та оптоволоконних кабелів Ви знаєте?

Поясніть відмінності між стандартами T568A і T568B.

Що таке централізоване та колапсоване кабелювання?

Лабораторна робота 3. Протоколи TCP/IP та IP-адресація.

Перед виконанням лабораторної роботи необхідно вивчити такі питання.

Вивчити роль мережевого рівня моделі TCP/IP, протоколів IPv4 та IPv6, відзначити відмінності цих протоколів між собою.

Ознайомитись з особливостями IP-адресації протоколу IPv4.

Освоїти конфігурування і налаштування стеку протоколів TCP/IP на хості у ручному режимі.

Освоїти формати основних утиліт при роботі з TCP/IP та його діагностиці.

Виконати в лабораторії.

1. Провести тестування протоколу TCP/IP на хості. Для цього визначити: його IP-адресу (чи вона присвоєна статично, чи динамічно; її значення; кількість та тип мережевих адаптерів, встановлених на ЕОМ, значення їх MAC-адреси; тип вузла). У випадку використання протоколу DHCP визначити, чи адреси присвоюються статично, чи динамічно (якщо динамічно, тоді задати термін дії оренди), поновити оренду.

коректність функціонування протоколу TCP/IP, а саме: відсутність дублювання IP-адрес; працездатність операційної системи, працездатність мережевого адаптера.

2. Провести тестування протоколу TCP/IP у мережі. Для цього визначити:

якість зв'язку з окремим хостом у локальній мережі (час проходження тестового сигналу, наявність втрат сигналу при цьому).

провести тестування зв'язку Вашого хоста з віддаленою мережею. Для цього спочатку з'ясувати наявність такої мережі, вивчивши таблицю маршрутизації Вашого хоста за допомогою утиліти route.

визначити якість зв'язку з окремим хостом у віддаленій мережі (час проходження тестового сигналу, наявність втрат сигналу при цьому). Порівняти результати тестів з хостом у локальній та віддаленій мережах.

3. Виконати індивідуальне завдання.

Контрольні запитання

Якого класу IP-адреси застосовуються на хостах у Ваших комп'ютерних лабораторіях університету?

Поясніть алгоритм визначення приналежності хоста-отримувача повідомлення до локальної чи віддаленої мережі? Для чого він використовується?

CRM (Customer Relationship Management System) управління відносинами з клієнтами.

Управління відносинами з клієнтами – поняття що охоплює концепції, котрі використовуються компаніями для управління їхніми взаємовідносинами зі споживачами, включаючи збір, зберігання і аналіз інформації про споживачів, постачальників, партнерів та інформації про взаємовідносини з ними. Сучасна CRM спрямована на вивчення ринку і конкретних потреб клієнтів. На основі цих знань розробляються нові товари або послуги і таким способом компанія досягає поставлених цілей і покращує свої фінансові показники [2].

Існує три CRM-підходи, кожний з котрих може бути реалізованим окремо від інших:

- оперативний – автоматизування споживчих бізнес-процесів, що допомагає персоналу, який працює з клієнтами виконувати свої функції;
- співробітницький – програма взаємодії зі споживачами без участі персоналу, який працює з клієнтами;
- аналітичний – аналіз інформації про споживачів із різноманітними цілями.

Принципи CRM-систем. Наявність єдиного сховища інформації, звідки у довільний момент доступні усі відомості про усі випадки взаємодії з клієнтом, синхронізування управління множинними каналами взаємодії, постійний аналіз зібраної інформації про клієнтів та прийняття відповідних організаційних рішень – наприклад, "сортування" клієнтів на основі їхньої значимості для компанії.

Можливості CRM-систем:

- швидкий доступ до актуальної інформації про клієнтів;
- оперативність обслуговування клієнтів та проведення операцій;
- формалізування схем взаємодії з клієнтами, автоматизування документообігу;
- швидке отримання всіх необхідних звітних даних та аналітичної інформації;
- зниження операційних витрат менеджерів;
- контроль за роботою менеджерів;
- узгоджена взаємодія між працівниками і підрозділами.

Управління бізнес-процесами дозволяє автоматизувати послідовні операції, які виконуються працівниками організації;

Управління контактами, історія взаємодії з клієнтами – це єдина база даних всіх контрагентів компанії (клієнтів, постачальників, конкурентів) з внесеною раніше докладною інформацією про них, про їх працівників тощо.

Система дозволяє здійснювати швидкий пошук важливої інформації про контрагентів, отримувати всю історію зустрічей, перемовин, листування, угод тощо. Це дуже зручний інструмент для швидкої і якісної роботи з величезними масивами інформації про клієнтів. Система автоматично нагадує

про необхідність зателефонувати, про заплановані зустрічі та інші заходи. CRM дозволяє складати плани за різними показниками. За історією проектів можна визначати проблемні зони у циклах продажів. Планування і контроль виконання плану за фактом. Уся робота з клієнтом відбувається в одній системі: планування заходів; здійснення угод; підготування і виписування необхідних звітних документів.

Планування та управління закупівлями і допровадженням – у системі менеджери завжди можуть бачити наявність і кількість товарів на складі. Відповідальні працівники можуть стежити за виконанням плану закупівель.

Управління маркетингом – дозволяє управляти маркетинговими заходами і визначати їх результативність. Можливість сегментування наявних у базі клієнтів (діючих і потенційних) за певними параметрами для проведення маркетингових заходів.

Автоматизування документообігу – у систему можна ввести шаблони довільних документів, які використовуються в організації, при цьому зникає необхідність ручного складання нового документа при виникненні події. Швидко автоматичне заповнення шаблонів угод, які зберігаються в системі.

MES (Manufacturing Execution System) спеціалізоване прикладне програмне забезпечення. Системи класу MES призначені для виробничого середовища підприємства. Вони відстежують і документують весь виробничий процес, відтворюють виробничий цикл у реальному часі.

На відміну від ERP, яка не має безпосереднього впливу на процес виробництва, за допомогою MES стає можливим коригувати процес стільки разів, скільки це буде потрібно. Інакше кажучи, системи такого класу призначені для оптимізування процесу виробництва і підвищення його рентабельності. Збираючи та аналізуючи дані, одержувані, наприклад, від технологічних ліній, система надає більш детальне уявлення про виробничу діяльність підприємства (від формування замовлення до відвантаження готової продукції), покращуючи фінансові показники підприємства.

Усі головні показники, які входять в основний курс економіки галузі (віддача основних фондів, обіг грошових коштів, собівартість, прибуток і продуктивність) детально відтворюються у ході виробництва. Фахівці називають MES мостом між фінансовими операціями ERP-систем і оперативною діяльністю підприємства на рівні цеху, ділянки або лінії.

WMS (Warehouse Management System) система управління складом – система управління, що забезпечує автоматизування та оптимізування всіх процесів складської роботи підприємства.

Архітектура ІС управління складом побудована за трирівневим принципом. Перший компонент є видимою для користувача частина – інтерфейс типу "користувач"/"клієнтський додаток" за допомогою якого користувач здійснює введення, модифікування та видалення даних, створює запити на виконання операцій та запити на вибірку даних (одержання звітів).

Охарактеризувати фізичну топологію, на якій базується технологія Token Ring. Описати логіку роботи даної технології.

Визначити, який вид мережевих технологій забезпечує гарантовану перепускную здатність? Пояснити, за рахунок чого.

За допомогою діагностичних утиліт типу ipconfig визначити MAC-адресу мережевого адаптера комп'ютера, за яким Ви працюєте. Порівняти її з адресою сусіднього комп'ютера.

Пояснити формат запису MAC-адреси.

Контрольні запитання

Чому технологія Ethernet - найпоширеніша на сьогодні технологія LAN?

Що таке колізія? Що є причиною її виникнення?

Чи впливає завантаженість мережі Ethernet на її продуктивність? Чому?

Що означає правило "5-4-3"?

Скільки є рівнів пріоритетів у мережі Token Ring?

Скільки швидкостей передавання підтримує технологія Token Ring?

Які функції активного монітора за технологією Token Ring?

Лабораторна робота 2. Структуровані кабельні системи.

Перед виконанням лабораторної роботи необхідно вивчити такі питання.

Ознайомитись зі стандартами, які використовуються при проектуванні, побудові та тестуванні СКС.

Вивчити шість основних компонентів СКС.

Ознайомитись з особливостями проектування СКС. Подати свій приклад такої мережі. Для даного прикладу створити структурну схему СКС із зазначенням усіх необхідних компонент, місце їх розташування. Обґрунтувати своє рішення.

Виконати в лабораторії.

Визначити структуру кабельної системи комп'ютерного класу, в якому проводяться заняття. Вивчити структуру мережі кафедри і зобразити її на структурній схемі, зазначивши: пасивне обладнання та активне обладнання, типи LAN-технологій, які застосовані.

Для навчального корпусу, в якому Ви навчаєтесь, запропонувати свій варіант СКС.

Виконати індивідуальне завдання.

Контрольні запитання

Що таке СКС? Яка їх основна перевага? Які основні компоненти СКС Ви знаєте?

Що таке NEXT, FEXT?

Які стандарти є визначальними при формуванні СКС?

Подайте структуру СКС із зазначенням максимальних відстаней кабелів горизонтальної підсистеми, телекомунікаційної комірки, підсистеми робочого місця.

Запрос4 : запрос на выборку				
	Прізвище	Ім'я	По-батькові	Сума кредиту
	Бачинський	Назар	Романович	2 000,00 грн.
	Баків	Дмитро	Васильович	1 230,00 грн.
	Двигун	Сергій	Борисович	7 600,00 грн.
	Дідухів	Галина	Іванівна	2 300,00 грн.
	Залужна	Христина	Семенівна	3 000,00 грн.
	▶ Кожан	Лев	Маркович	2 400,00 грн.
	Пронів	Олена	Василівна	5 600,00 грн.
	*			

Рис. 8.38. Результати запиту до об'єднаної таблиці про всіх співробітників, що брали товари в кредит.

Утворіть запит про всіх неодружених чоловіків, що не брали товари в кредит.

Вилучення та утворення нових зв'язків між таблицями.

Анулюйте зв'язок між таблицями. Для цього у меню *Сервіс* виберіть команду *Схема даних*, зафіксуйте вказівник миші на лінії зв'язку і натисніть клавішу *Delete*.

Знову встановіть зв'язок між таблицями за спільним полем **Табельний номер**.

Утворіть реляційну форму, на якій розмістіть поля **Прізвище, Ім'я, По-батькові, Сімейний стан, Кількість дітей, Кредит**.

Утворіть реляційний запит: виведіть на екран всі записи про жінок, які заборгували фірмі більше, ніж 1500 грн.

8.3. Лабораторний практикум з телекомунікаційних технологій і html

Лабораторна робота 1. Топології комп'ютерних мереж. Базові технології локальних комп'ютерних мереж.

Перед виконанням лабораторної роботи необхідно вивчити такі питання.

Роль каналного рівня моделі OSI.

Основні властивості і характеристики технології Ethernet.

Основні властивості і характеристики технології Token Ring.

Особливості множинного доступу до середовища передавання даних.

Вміти порівняти між собою конфліктні та безконфліктні методи множинного доступу та технології на їх основі.

Виконати в лабораторії.

Описати можливі фізичні топології, на яких може базуватися технологія Ethernet, та порівняти їх між собою.

Навести приклади, у яких випадках відбувається співпадіння фізичної та логічної топології мережі Ethernet, а у яких не відбувається.

Описати логіку роботи методу доступу до середовища передавання даних, який використовується в технології Ethernet.

Другий компонент (прихована від користувачів частина системи) – сервер бази даних. Користувач через "клієнтський додаток" ініціює процедуру запиту на вибірку, введення, модифікування або видалення даних у БД.

Третій компонент – бізнес-логіка ("завдання" або "процеси" – спеціалізовані програми) здійснює ініційоване користувачем оброблення даних і повертає оброблені дані у БД, повідомляючи користувачеві через інтерфейс "клієнтського додатку" про завершення оброблення запиту.

Цілі впровадження:

- активне управління складом;
- збільшення швидкості набору товару;
- отримання точної інформації про знаходження товару на складі;
- ефективне управління товаром, що має обмежений термін придатності;
- отримання інструменту для підвищення ефективності і розвитку процесів з оброблення товару на складі;
- оптимізування використання складських площ.

EAM (Enterprise Asset Management) система управління основними фондами підприємства, що дозволяє раціонально використовувати устаткування, витрати на техобслуговування, ремонти і матеріально-технічне постачання [2]. Основні фонди – це засоби праці, які багаторазово беруть участь у виробничому процесі, зберігаючи при цьому свою натуральну форму, поступово зношуючись, переносять свою вартість частинами на знову створювану продукцію. У бухгалтерському та податковому обліку відтворені у грошовому вираженні основні фонди називаються основними засобами. Історично EAM-системи виникли з CMMS-систем (ще одного класу IC, управління ремонтами). Зараз модулі EAM входять до складу великих пакетів ERP-систем (таких як mySAP Business Suite, IFS Applications, Oracle E-Business Suite і подібні).

HRM (Human Resource Management) система управління персоналом – є однією з найважливіших складових частин сучасного менеджменту. Основна мета таких систем – залучення та утримання цінних для підприємства фахівців. HRM-системи вирішують два головні завдання: упорядкування всіх облікових і розрахункових процесів, пов'язаних з персоналом; зниження відсотка втрати працівників. Таким чином, HRM-системи, у певному сенсі, можна назвати CRM-системами навпаки, залучати та утримувати не покупців, а власних працівників. Зрозуміло, тут методи застосовуються зовсім інші, але загальні підходи є схожими [2].

Функції HRM-систем:

- пошук персоналу;
- оцінювання фахового рівня персоналу;
- навчання та кар'єрне зростання персоналу;
- управління корпоративною культурою;
- мотивування персоналу;
- організація праці.

1.3. Класифікація ІС. Рівні ІС в організації

ІС можуть бути класифікованими за рядом характерних ознак.

За рівнем у системі державного управління: територіальні і галузеві; міжгалузеві; підприємств.

За рівнем інтелектуалізації: інформаційно-довідкові; інформаційно-пошукові; підтримки прийняття управлінських рішень; з використанням баз знань; експертні системи.

За ступенем централізації оброблення інформації: централізовані і децентралізовані;

За принципом інтегрування: багаторівневі з інтеграцією за рівнями управління та функціями управління; однорівневі.

За видами процесів: для наукових досліджень; для автоматизованого проектування; організаційного управління; управління виробничими процесами; управління технологічними процесами; навчальні.

За сферою діяльності: культурологічні; владні; науково-технічні; соціальні; фінансово-економічні; міжнародних організацій;

За режимом оброблення інформації: в режимі реального часу; в автономному режимі.

За рівнем у системі державного управління: загальнодержавні ІС призначені для вирішення найважливіших проблем національної економіки. тощо.

Територіальні (регіональні) ІС призначені для управління адміністративно-територіальними регіонами. Сюди належать ІС області, міста, району. Ці системи здійснюють оброблення інформації, яка необхідна для реалізування функцій управління регіоном, формування звітності і подання оперативних даних органам місцевого самоврядування.

Галузеві ІС управління призначені для галузей управління підвідомчими підприємствами. Галузеві ІС діють у промисловості, енергетиці, транспорті тощо. У них розв'язуються задачі інформаційного обслуговування апарату управління відповідних міністерств.

Міжгалузеві ІС є спеціалізованими системами функціональних органів управління національною економікою (планових, фінансових, статистичних та ін.).

Централізовані ІС – накопичення і оброблення інформації здійснюють у єдиному центрі. Доступ до інформаційних ресурсів ІС може здійснюватись віддалено.

Децентралізовані ІС побудовані за автономним принципом. Кожна ІС певного рівня обслуговує певне коло користувачів.

Оскільки організації мають різноманітні інтереси і структуру, тому для їх обслуговування існують різні види ІС. Ніяка єдина система не може цілком забезпечувати потреби організації у всій інформації. На рис. 1.4 подані види ІС, які складають основу організації. На цьому рисунку організація поділена на рівні: стратегічний; управлінський; знання й експлуатаційний. Далі ІС поділена на функціональні області типу продажу і маркетингу, виробництва, фінансів, бухгалтерського обліку і людських ресурсів.

Закрийте початкові таблиці **Основні дані** та **Кредит**.

На підставі створення нової форми (див. рис. 10.37) можна отримати нову таблицю з об'єднаними полями; **Табельний номер, Прізвище, Ім'я, По-батькові, Посадовий оклад** з таблиці **Основні дані** та полем **Сума кредиту** з таблиці **Кредит** (рис. 8.36).

Табел	Прізвище	Ім'я	По-батькові	Посадовий оклад	Сума кредиту
1	Бачинський	Назар	Романович	2 100,00р.	2 000,00 грн.
2	Баків	Дмитро	Васильович	1 575,00р.	1 230,00 грн.
3	Двигун	Сергій	Борисович	1 890,00р.	7 600,00 грн.
4	Дідухів	Галина	Іванівна	1 100,00р.	2 300,00 грн.
5	Залужна	Христина	Семенівна	3 150,00р.	3 000,00 грн.
6	Кожан	Лев	Маркович	1 260,00р.	2 400,00 грн.
7	Мацігін	Григорій	Петрович	3 150,00р.	0,00 грн.
8	Мозговий	Іван	Іванович	3 675,00р.	0,00 грн.
9	Павлик	Віктор	Климович	1 575,00р.	0,00 грн.
10	Пронів	Олена	Василівна	1 365,00р.	5 600,00 грн.

Рис. 8.36. Нова таблиця з об'єднаними полями з різних таблиць.

Створіть реляційну форму, на якій будуть поля **Прізвище, Ім'я, По-батькові, Дата прийняття на роботу, Сума кредиту**.

Реляційні запити.

Утворимо запит до двох таблиць про всіх співробітників, що брали товари в кредит.

У вікні бази даних на вкладці **Запросы** натиснемо кнопку **Создать**.

Вибираємо **Конструктор, Ок**.

У вікні **Добавление таблицы** на вкладці **Таблицы** виберемо назву **Основні дані, Добавити**, потім **Кредит, Добавити, Закрити**.

У вікні **Конструктора** запитів заповнимо бланк так, як на рис. 8.37.

Рис. 8.37. Реляційний запит.

Отримаємо в результаті запиту наступну таблицю (рис. 8.38).

У вікні, що з'явилося, потрібно мишкою перетягнути поле **Табельний номер** з таблиці **Основні дані** до поля **Табельний номер** таблиці **Кредит**. Тоді таблиця **Основні дані** буде головною, а таблиця **Кредит** – підпорядкованою. У наступному вікні натисніть кнопки **Об'єднання** та **Створити**. Відкриється наступне вікно, де повинна бути включена перша радіоклавіша (рис. 8.34).

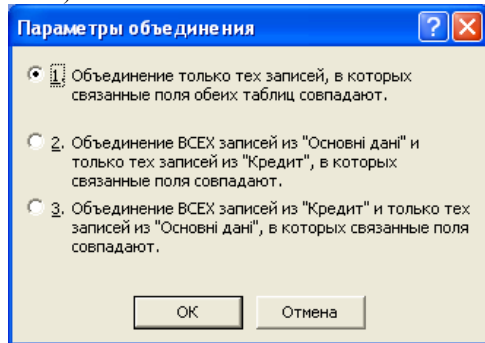


Рис. 8.34. Вікно вибору параметрів об'єднання двох таблиць.

Натисніть ОК. Закрийте вікно зі збереженням макета.

Реляційна форма.

Створимо форму, на якій розмістимо дані з двох таблиць: з першої – **Табельний номер**, **Прізвище**, **Ім'я**, **По-батькові**, **Посадовий оклад**, з другої – **Сума кредиту**.

У вікні бази даних на вкладці **Форми** натисніть кнопку **Створити**. У наступному вікні виберіть: **Мастер форм, Ок**.

У списку **Таблиці**, **Запросы** виберемо назву головної таблиці – **Основні дані**, а також (за допомогою кнопки >) поля **Табельний номер**, **Прізвище**, **Ім'я**, **По-батькові**, **Посадовий оклад**. Потім аналогічно виберемо з таблиці **Кредит** поле **Сума кредиту** і натиснемо кнопку **Далее**.

Натиснемо радіоклавішу **В один столбец** і кнопку **Далее**. Виберемо стиль форми, наприклад, **Ткань**, **Далее**.

Завершуємо створення форми натиском кнопки **Готово** (рис. 8.35).

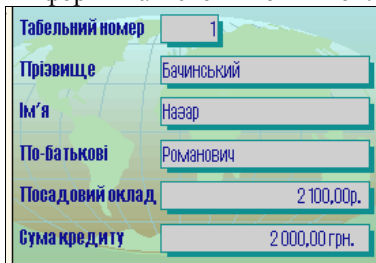


Рис. 8.35. Реляційна форма створена з таблиць **Основні дані** та **Кредит**.

Системи створюються, щоб обслуговувати ці різні організаційні інтереси. Організаційні рівні обслуговують чотири головних типи ІС: системи експлуатаційного рівня, системи рівня знань, системи управлінського рівня та стратегічні системи.

Системи експлуатаційного рівня підтримують операційних менеджерів, стежать за елементарними діями організації типу продажу, платежів, кредитування та ін. Основна мета систем на цьому рівні полягає в тому, щоб відповісти на типові питання і проводити потоки транзакцій через організацію.

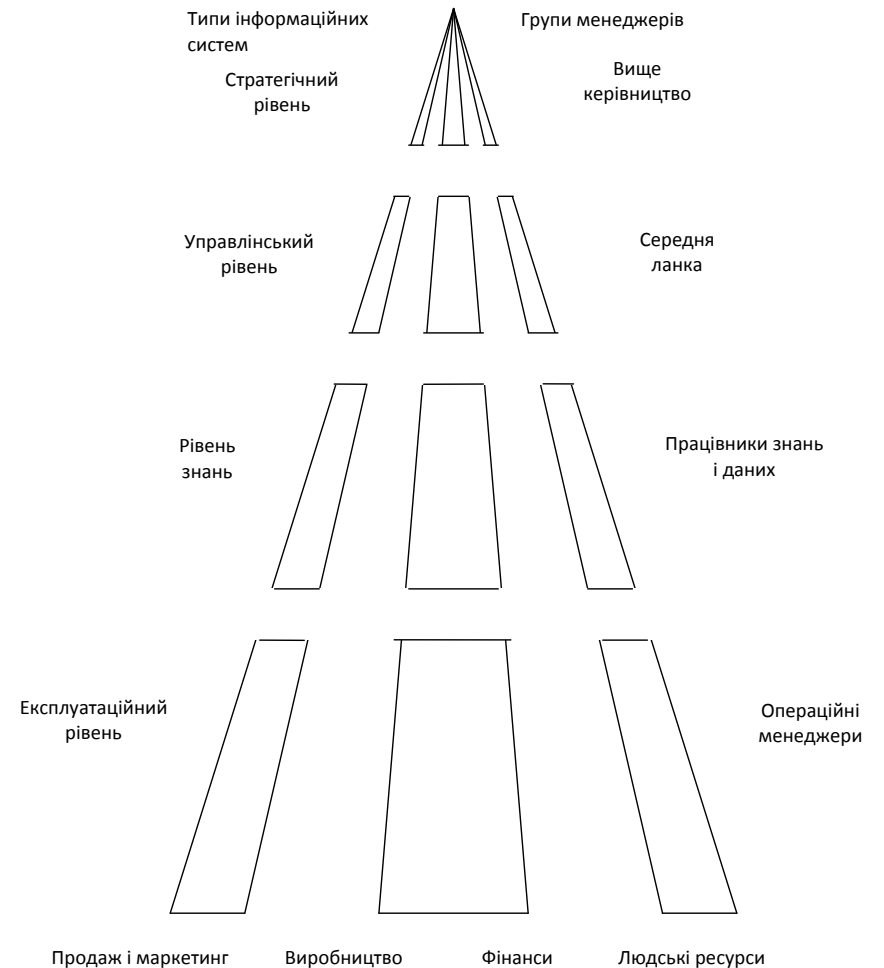


Рис. 1.4. Види ІС, які складають основу організації.

Системи рівня знань підтримують працівників знання й оброблювачів даних в організації. Мета систем рівня знань полягає в тому, щоб допомогти діловій фірмі інтегрувати нове знання в бізнес і допомагати організації керувати потоком документів.

Системи рівня знань, особливо у формі робочих станцій і офісних систем, сьогодні є найбільш швидко зростаючими додатками в бізнесі.

Системи управлінського рівня розроблені, щоб обслуговувати контроль, управління, прийняття рішень і адміністративні дії середніх менеджерів. Вони визначають, чи добре працюють об'єкти, і періодично сповіщають про це. Наприклад, система управління переміщеннями повідомляє про переміщення загальної кількості товару, рівномірність роботи торговельного відділу і відділу, що фінансує витрати для службовців у всіх філіях компанії, відзначаючи, де фактичні витрати перевищують бюджет.

Системи стратегічного рівня – це інструмент допомоги керівникам вищого рівня, що готують стратегічні дослідження і тривалі тренди у фірмі й у діловому оточенні. Їхнє основне призначення – приводити у відповідність зміни в умовах експлуатування з існуючою організаційною можливістю. Який буде рівень зайнятості через п'ять років? Які тривалі промислові фінансові тренди і де наші підйоми і спади? Які виробы ми повинні робити через п'ять років?

ІС можуть також бути диференційовані функціональним чином. Головні організаційні функції типу продажу, виробництва, фінансів, бухгалтерського обліку і людських ресурсів обслуговуються власними інформаційними системами.

Типова організація має системи різних рівнів: експлуатаційного, управлінську, знання і стратегічну для кожної функціональної області. Наприклад, комерційна функція має комерційну систему на експлуатаційному рівні, щоб робити запис щоденних комерційних даних і обробляти замовлення.

Системи управлінського рівня відслідковують щомісячні комерційні дані всіх комерційних територій і доповідають про території, де продаж перевищує очікуваний рівень або падає нижче очікуваного рівня. Система прогнозу прогнозує комерційні тренди протягом п'ятирічного періоду – обслуговує стратегічний рівень.

Табельний ном	Сума кредиту
1	2 000,00
2	1 230,00
3	7 600,00
4	2 300,00
5	3 000,00
6	3 400,00
7	0,00
8	0,00
9	0,00
10	5 600,00
	0,00

Рис. 8.31. Фрагмент другої таблиці **Кредит** бази даних **Кадри**.

Встановимо зв'язок між двома таблицями даної бази. Спочатку, перебуваючи у вікні бази даних, відкриємо меню **Сервіс** і виберемо команду **Схема даних**. Якщо зв'язки між таблицями ще не встановлювалися (як у даному випадку), то можна побачити вікно **Добавление таблицы** (рис. 8.32).

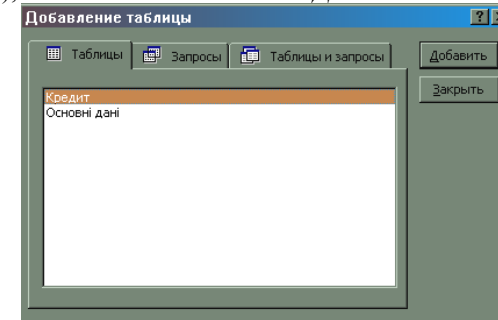


Рис. 8.32. Вікно **Добавление таблицы**.

Клацніть мишкою по назві **Основные дані**, по кнопці **Добавить**, потім по назві **Кредит**, по кнопці **Добавить** та **Закреть**. (Якщо зв'язки вже встановлювалися, замість вікна **Добавление таблицы** може з'явитися вікно **Схема даних**. Тоді потрібно у меню **Связи** (або у контекстному меню цього вікна) вибрати команду **Добавить таблицу**). Послідовність під'єднання таблиць має істотне значення. Таблиця названа першою буде головною, а інші – підпорядковані їй (рис. 8.33).

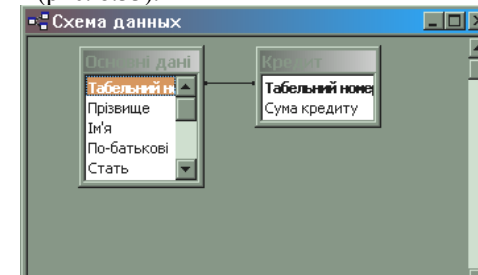


Рис. 8.33. Фрагмент вікна **Схема даних** з двома таблицями **Основные дані** та **Кредит**, зв'язаними між собою

Натисніть **Ок** і запустіть запит на виконання. Як результат відкриється вікно, що має назву **Введіть значення параметра**. У нього потрібно ввести прізвище тієї людини, інформацію про яку потрібно вибрати з бази даних. Після вводу прізвища на екран виведуться відповідні записи.

Формуючи наступний параметричний запит, у вікні команди **Запрос – Параметри** спочатку видаліть дані, що стали непотрібними.

Створіть параметричний запит для відшукування записів за місцем праці.

Створіть параметричний запит для відшукування записів за іменем.

Лабораторна робота 9. ЗВ'ЯЗОК МІЖ ТАБЛИЦЯМИ

Відкрийте вікно таблиці **Основні дані** бази даних **Кадри**, що містить інформацію про кадровий склад установи (рис. 8.30).

Табельний ном	Прізвище	Ім'я	По-батькові	Стать	Дата народжен	Сімейний стан	Кількість дітей	Дата прийняття	Посад
1	Бачинський	Назар	Романович	чол	12.01.1967	одруж	2	12.12.1987	2
2	Ваків	Дмитро	Васильович	чол	11.03.1989	неодруж	0	12.04.2009	1
3	Двигун	Сергій	Борисович	чол	12.04.1990	одруж	1	09.09.2008	1
4	Дідухів	Галина	Іванівна	жін	01.01.1991	неодруж	0	02.02.2002	2
5	Залужна	Христина	Семенівна	жін	03.03.1990	одруж	1	04.04.2008	3
6	Кожан	Лев	Маркович	чол	04.04.1990	одруж	3	05.05.2008	1
7	Мацігін	Григорій	Петрович	чол	05.05.1985	неодруж	0	06.07.2007	3
8	Мозговий	Іван	Іванович	чол	06.06.1966	одруж	2	03.05.1990	3
9	Павлик	Віктор	Климович	чол	07.07.1970	одруж	1	05.08.1992	1
10	Пронів	Олена	Василівна	жін	08.08.1960	одруж	4	09.12.1980	1

Рис. 8.30. Фрагмент таблиці **Основні дані** бази даних **Кадри**.

Створіть нову таблицю у цій самій базі даних. Записи таблиці повинні складатися лише з двох полів: **Табельний номер** та поля, що називатиметься **Сума кредиту**. Перше поле повинно мати ті ж характеристики, що й поле **Табельний номер** у першій з таблиць бази даних, а друге поле повинно мати тип **Денежний**, формат – **с разделителем разрядов** (Поле Сума кредиту міститиме суму, на яку у магазині фірми даному співробітнику відпущено товари у кредит. Вважатимемо, що борг погашається регулярними автоматичними відрахуваннями із окладу).

Зробіть перше поле (**Табельний номер**) ключовим (так само як і в таблиці **Основні дані**) і збережіть таблицю, надавши їй назву **Кредит** (рис. 8.31).

Заповніть дану таблицю записами (не менше 10). У полі **Табельний номер** таблиці **Кредит** потрібно розмістити такі ж числові значення, як і у полі **Табельний номер** таблиці **Основні дані** бази даних **Кадри**. У поле **Сума кредиту** потрібно ввести 0 або інше число.

РОЗДІЛ 2. СТРУКТУРИ ІНФОРМАЦІЙНОГО ПРОЦЕСУ

2.1. Параметри інформаційного процесу і зв'язок між ними

Як відомо, будь-який інформаційний процес визначається зміною параметрів, які його задають, або приведенням одних параметрів (вихідних — $\Phi_1, \Phi_2, \dots, \Phi_m$) у відповідність з іншими (вхідними — x_1, x_2, \dots, x_n) за законом:

$$\begin{cases} \Phi_1 = \varphi_1(x_1, \dots, x_n), \\ \Phi_2 = \varphi_2(x_1, \dots, x_n), \\ \vdots \\ \Phi_m = \varphi_m(x_1, \dots, x_n), \end{cases} \quad (2.1)$$

де $(x_1, x_2, \dots, x_n) \in D \subseteq R^n$.

Нехай $\varphi_i \in C^1(D)$, $i = \overline{1, m}$. Припустимо, що значення одного з вихідних параметрів Φ_j однозначно визначається сукупністю значень інших $\Phi_1, \dots, \Phi_{j-1}, \Phi_{j+1}, \dots, \Phi_m$, тобто, якщо $\Omega_0 \subseteq R^{m-1}$ є множина точок, що відповідають усіляким точкам $(x_1, x_2, \dots, x_n) \in D$, то в Ω_0 буде мати місце функціональна залежність:

$$\Phi_j = f(\Phi_1, \dots, \Phi_{j-1}, \Phi_{j+1}, \dots, \Phi_m), \quad (2.2)$$

де $f \in C^1(\Omega)$, $\Omega \subseteq R^{m-1}$, $\Omega \supseteq \Omega_0$, Ω — відкрита множина, а при підстановці (2.1) в (6.2) виходить тотожність відносно $(x_1, x_2, \dots, x_n) \in D$:

$$\varphi_j(x_1, \dots, x_n) \equiv f(\varphi_1(x_1, \dots, x_n), \dots, \varphi_{j-1}(x_1, \dots, x_n), \varphi_{j+1}(x_1, \dots, x_n), \dots, \varphi_m(x_1, \dots, x_n)). \quad (2.3)$$

У цьому випадку будемо казати, що функція φ_j *залежить від функцій* $\varphi_1, \dots, \varphi_{j-1}, \varphi_{j+1}, \dots, \varphi_m$ в області D [12]. У загальному випадку функції $\varphi_1, \varphi_2, \dots, \varphi_m$ називаються *залежними* в області D , якщо одна з них залежить від інших.

Якщо ані в області D , ані в якій-небудь області $E \subseteq D$ не має місця тотожність (6.3), то функції $\varphi_1, \varphi_2, \dots, \varphi_m$ називаються *незалежними* в D [12].

Визначення 2.1. Вихідні параметри інформаційного процесу будуть незалежними (залежними), якщо незалежними (залежними) в області D будуть функції (6.1), що їх визначають.

Зауваження 2.1. У випадку незалежності вихідних параметрів їх визначення відповідно до (2.1) може проводитися одночасно (паралельно), що значно скорочує при необхідності час реалізації й аналізу інформаційного процесу. Цей процес можна представити як сукупність не зв'язаних між собою

«простих» процесів, результатом кожного з яких є отримання лише одного параметра φ_i , а дослідження поданого інформаційного процесу зведеться до дослідження скінченної сукупності «простих».

Завдяки зауваженню 2.1 важливою є можливість визначення залежності (незалежності) вихідних параметрів (функцій (2.1)). Відповідь на це питання дається властивостями матриці частинних похідних функцій φ_i , $i = \overline{1, m}$, — матриці Якобі неперервного інформаційного процесу [12]:

$$\begin{pmatrix} \frac{\partial \varphi_1}{\partial x_1} & \frac{\partial \varphi_1}{\partial x_2} & \dots & \frac{\partial \varphi_1}{\partial x_n} \\ \frac{\partial \varphi_2}{\partial x_1} & \frac{\partial \varphi_2}{\partial x_2} & \dots & \frac{\partial \varphi_2}{\partial x_n} \\ \dots & \dots & \dots & \dots \\ \frac{\partial \varphi_m}{\partial x_1} & \frac{\partial \varphi_m}{\partial x_2} & \dots & \frac{\partial \varphi_m}{\partial x_n} \end{pmatrix}. \quad (2.4)$$

Нехай $n \geq m$. Якщо хоч один визначник m -го порядку, складений з елементів матриці (2.4), відмінний від нуля в області D , то в цій області функції φ_i , $i = \overline{1, m}$, а тому і вихідні параметри інформаційного процесу, незалежні. Дійсно, припустимо, що

$$\det \begin{pmatrix} \frac{\partial \varphi_1}{\partial x_1} & \frac{\partial \varphi_1}{\partial x_2} & \dots & \frac{\partial \varphi_1}{\partial x_m} \\ \frac{\partial \varphi_2}{\partial x_1} & \frac{\partial \varphi_2}{\partial x_2} & \dots & \frac{\partial \varphi_2}{\partial x_m} \\ \dots & \dots & \dots & \dots \\ \frac{\partial \varphi_m}{\partial x_1} & \frac{\partial \varphi_m}{\partial x_2} & \dots & \frac{\partial \varphi_m}{\partial x_m} \end{pmatrix} \neq 0, \quad (2.5)$$


але якась з функцій, наприклад, φ_m , визначається через інші:

$$\varphi_m = f(\varphi_1, \dots, \varphi_{m-1}) \quad (2.6)$$

хоча б у деякій області $E \subseteq D$ (якби відмінним від нуля був інший визначник, то, змінивши нумерацію змінних, можна було б знову прийти до випадку (2.5)).

Продиференціюємо (2.6) по x_i , $i = \overline{1, m}$:

$$\frac{\partial \varphi_m}{\partial x_i} = \frac{\partial \varphi_m}{\partial \varphi_1} \frac{\partial \varphi_1}{\partial x_i} + \frac{\partial \varphi_m}{\partial \varphi_2} \frac{\partial \varphi_2}{\partial x_i} + \dots + \frac{\partial \varphi_m}{\partial \varphi_{m-1}} \frac{\partial \varphi_{m-1}}{\partial x_i}, \quad i = \overline{1, m}. \quad (2.7)$$

запустіть запит на виконання **Запуск** ; підтвердіть або скасуйте створення таблиці.

Після створення нової таблиці поверніться у вікно бази даних (клавіша F11). На вкладці **Таблиці** можна побачити назву створеної таблиці.

Створіть нову таблицю на основі таблиці про кадровий склад фірми, застосувавши запит на створення. Ця нова таблиця повинна містити записи про незаміжніх жінок, що не мають дітей і молодші ніж 30 років.

Створіть ще одну таблицю, що містить записи тільки про тих, хто працює в бухгалтерії й має посадовий оклад більший ніж 1500 грн.

Параметричні запити (Запити з умовами вибору)

У розглянутих запитах для умов відбору використовувалися константи, але замість констант у виразах умов відбору можна використовувати змінні (параметри). Параметри – це довільні текстові рядки в квадратних дужках. Для того, щоб встановити умову вибірки, застосовують вирази, в яких можна використовувати операції порівняння ”>”, ”<”, ”>=”, ”<=”, ”=”, логічні операції AND, OR, а також службові слова, описані в SQL.

Розглянемо приклад.

Допустимо, база даних велика й потрібно швидко отримувати інформацію про співробітників, задаючи їхні прізвища. Тоді можна створити параметричний запит. Зробимо це так.

Створимо спочатку запит на вибірку, у якому передбачимо виведення на екран всіх полів. Після цього у стовпчик **Прізвище**, у клітину **Умовне отбора** введіть текст, який з’явиться після запуску запиту на виконання: [Прізвище ?] (рис. 8.29).

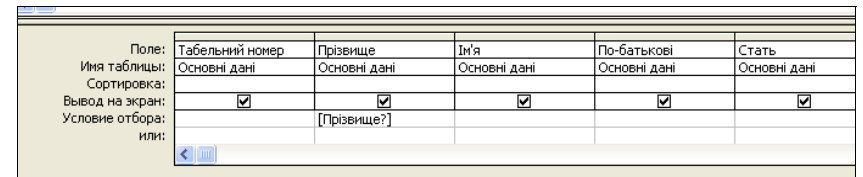


Рис. 8.29. Фрагмент конструювання запиту з інтерактивними умовами вибору.

У меню **Запрос** виберіть команду **Параметры**. У вікні, що відкриється, введіть у перший стовпчик параметр: “Прізвище ?” (тобто той самий текст, що і у клітину **Умовне отбора**, тільки без квадратних дужок), а у другий стовпчик – тип даних: **Текстовый** (рис. 8.29).

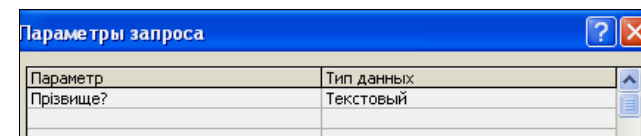


Рис. 8.29. Приклад підготовки параметрів запиту

Поновити базу даних за допомогою запиту, якщо всім, хто має від 18 до 25 років, збільшили оклад на 15 грн. (Використайте побудовувач для створення умови: кнопка **Построить, Операторы, Сравнения, Between, Вставить**. Замість слів **Выражение** введіть потрібні дати.)

Використайте запит на оновлення для збільшення значення поля **Посадовий оклад** на 30% у записах про одружених чоловіків, що молодші за 30 років і мають більше ніж 2 дітей.



Запити на створення таблиці.

За допомогою цього запиту можна утворити нову таблицю, яка складатиметься із записів іншої таблиці. При цьому усі записи задовольнятимуть задану умову.

Щоб виконати запит на створення, спочатку потрібно діяти, як під час створенні запиту на вибірку, перейти у вікно **Конструктора** запитів, заповнити бланк потрібними іменами полів та умовами, а потім у меню **Запрос** вибрати команду **Создание таблицы** і запустити запит на виконання. З'явиться вікно з повідомленням про створення таблиці. Потрібно підтвердити або скасувати задану операцію.

Створіть, наприклад, таблицю, у якій будуть тільки ті записи таблиці бази даних про кадровий склад установи, що стосуються неодружених чоловіків. Для цього виконайте наступні дії:

створіть запит на вибірку, задавши у бланку всі поля даної таблиці. Кожне поле повинно виводитись на екран. У стовпці **Стать** у клітинку **Условие отбора** введіть: "чол", а у стовпчик **Сімейний стан**, у клітинку **Условие отбора** введіть: "ложь" (без лапок);

у меню **Запрос** виберіть команду **Создание таблицы** (рис. 8.28), або за допомогою розкриття функцій групової кнопки **Тип запроса**  і вибору кнопки **Создание таблицы** .

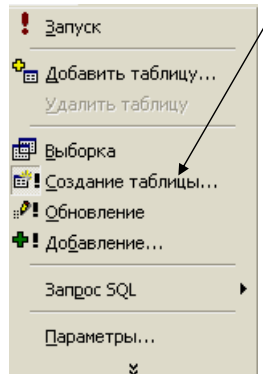


Рис. 8.28. Видгляд підменю пункту меню **Запрос**.

у діалогове вікно введіть назву нової таблиці і натисніть кнопку **Ок**;

3 (2.7) впливає, що елементи останнього рядка матриці (2.5) отримуються шляхом додавання відповідних елементів перших $m-1$ рядків, помножених попередньо на множники $\frac{\partial \varphi_m}{\partial \varphi_1}, \dots, \frac{\partial \varphi_m}{\partial \varphi_{m-1}}$. Але тоді визначник (2.5)

дорівнює нулю. Отримали суперечність, тому припущення (2.6) хибне.

Розглянемо загальний випадок. Рангом матриці Якобі (2.4) в області D назвемо найвищий з порядків визначників, що складаються з елементів цієї матриці і не дорівнюють тотожно нулю в D . Кажуть, що **ранг r досягається** в деякій точці $M^0(x_1^0, x_2^0, \dots, x_n^0)$ області, якщо визначник r -го порядку в цій точці відмінний від нуля.

Теорема 2.1. Нехай ранг матриці Якобі інформаційного процесу (2.4) в області D є r і досягається в точці $M^0(x_1^0, x_2^0, \dots, x_n^0) \in D$, при цьому $r = m$. Тоді в D знайдеться така область зміни вхідних параметрів інформаційного процесу, де вихідні будуть незалежними.

Доведення. В умовах теореми існує окіл D_0 точки M^0 , в якому r функцій з $\varphi_1, \varphi_2, \dots, \varphi_m$, похідні яких входять в визначник r -го порядку, який відрізняється від нуля в M^0 , будуть незалежними, а інші будуть залежати від них [12], звідки при $r = m$ витікає висновок теореми.

2.2. Варіаційна матриця інформаційного процесу

Нехай тепер в області D вихідні параметри інформаційного процесу (функції φ_i , $i = \overline{1, m}$) можуть бути як залежними, так і незалежними. Для однакової викладу всі параметри далі будемо позначати u_i , де $i = \overline{1, N}$, $N = n + m$ (для $i = \overline{1, n}$ u_i відповідають вхідним параметрам). Тоді реалізацію процесу можна формалізувати наступним чином:

$$u_k = F_k(u_{k_1}, \dots, u_{k_{s_k}}), \quad n < k \leq N, \quad k_1, \dots, k_{s_k} < k, \quad (2.8)$$

де всі F_k є досить гладкими функціями своїх аргументів. Як результат процесу розглядається сукупність величин u_k (вихідні параметри). Не накладаючи серйозних обмежень, можна припускати, що результат — це величина u_N .

Співвідношення (2.8) визначають процес обчислення функції (2.2), що є загальною формалізацією інформаційного процесу:

$$\Phi(x_1, \dots, x_n) = (\Phi_1, \Phi_2, \dots, \Phi_m)^T,$$

чи в нових позначеннях

$$\Phi(u_1, \dots, u_n) = (u_{n+1}, u_{n+2}, \dots, u_N)^T. \quad (2.9)$$

Якщо обчислення (2.9) спочатку задані за допомогою (2.8), то при великому N отримати явний вираз функції Φ (тобто функцій φ_i , $i=1, m$) через вхідні дані ($u_i, i=1, n$) важко й не завжди можливо. Для одержання достатньої умови такого представлення кожне рівняння системи (1.8) запишемо в еквівалентному виді:

$$F_k(u_{k_1}, \dots, u_{k_{s_k}}) - u_k = F_k(u_{k_1}, \dots, u_{k_{s_k}}) - u_k + 0 \cdot \sum_{i=k, k_1, \dots, k_{s_k}} u_i = G_k(u_1, u_2, \dots, u_N) = \\ = G_k(x_1, \dots, x_n, \Phi_1, \dots, \Phi_m) = 0$$

перетворивши (2.8) в еквівалентну систему рівнянь (у загальному випадку – нелінійну):

$$\begin{cases} G_1(x_1, \dots, x_n; \Phi_1, \dots, \Phi_m) = 0, \\ G_2(x_1, \dots, x_n; \Phi_1, \dots, \Phi_m) = 0, \\ \dots \dots \dots \\ G_m(x_1, \dots, x_n; \Phi_1, \dots, \Phi_m) = 0 \end{cases} \quad (2.10)$$

Якщо $G_i \in C^1(D)$, $i = \overline{1, m}$, де D — $n + m$ -вимірний прямокутний паралелепіпед

$$D = [x_1^0 - \Delta_1, x_1^0 + \Delta_1] \times \dots \times [x_n^0 - \Delta_n, x_n^0 + \Delta_n] \times [\Phi_1^0 - \bar{\Delta}_1, \Phi_1^0 + \bar{\Delta}_1] \times \dots \times [\Phi_m^0 - \bar{\Delta}_m, \Phi_m^0 + \bar{\Delta}_m]$$

з центром в точці $(x_1^0, \dots, x_n^0, \Phi_1^0, \dots, \Phi_m^0)$, координати якої задовольняють системі (2.10), і визначник матриці Якобі для функцій G_i , $i = \overline{1, m}$, по змінним Φ_1, \dots, Φ_m відмінний від нуля, тобто

$$\det \begin{pmatrix} \frac{\partial G_1}{\partial \Phi_1} & \frac{\partial G_1}{\partial \Phi_2} & \dots & \frac{\partial G_1}{\partial \Phi_m} \\ \frac{\partial G_2}{\partial \Phi_1} & \frac{\partial G_2}{\partial \Phi_2} & \dots & \frac{\partial G_2}{\partial \Phi_m} \\ \dots \dots \dots \\ \frac{\partial G_m}{\partial \Phi_1} & \frac{\partial G_m}{\partial \Phi_2} & \dots & \frac{\partial G_m}{\partial \Phi_m} \end{pmatrix} \neq 0,$$

тоді в деякому околі точки $(x_1^0, \dots, x_n^0, \Phi_1^0, \dots, \Phi_m^0)$ система (2.10) визначає Φ_1, \dots, Φ_m в виді (2.1), до того ж $\varphi_i \in C^1(D)$, $i = \overline{1, m}$.

Введіть у клітину **Поле: Підрозділ**, у клітину **Обновление: Склад**, а у клітину **Условие отбора: Плановий**.

Отримаємо наступне вікно з запитом на оновлення бази (рис. 8.26).

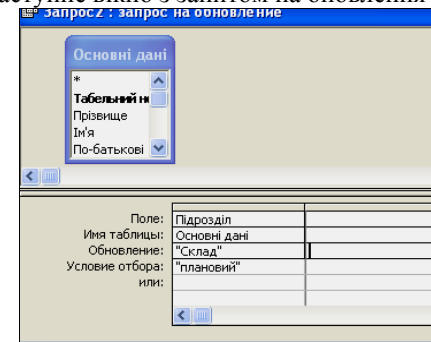


Рис. 8.26. Вікно з оформленим запитом на оновлення бази.

Використайте запит на оновлення для збільшення посадового окладу всіх співробітників на 15%.

Для цього у клітину **Поле** введіть: **Посадовий оклад**, клітину **Условие отбора** залишіть порожньою, а у клітину **Обновление** введіть вираз: **[Посадовий оклад]*0,15 + [Посадовий оклад]**. Для вводу цього виразу можна використати ПОБУДОВУВАЧ виразів, натиснувши на кнопку панелі інструментів **Побудувати**. Відкриється наступне вікно (рис. 8.27).

Далі у вікні побудовувача потрібно двічі клацнути на слові **Таблицы**, вибрати назву своєї таблиці **Основні дані**, у наступному стовпці – **Посадовий оклад**, натиснути кнопку **Вставити** і т.д.

Внесіть зміни у базу даних за допомогою запиту на оновлення, якщо всім жінкам збільшили оклад на 20%. (Дійте аналогічно до попереднього і додайте ще один стовпчик, у якому у клітину **Поле** введіть: **Стать**, а у клітину **Условие отбора** – “жін”).

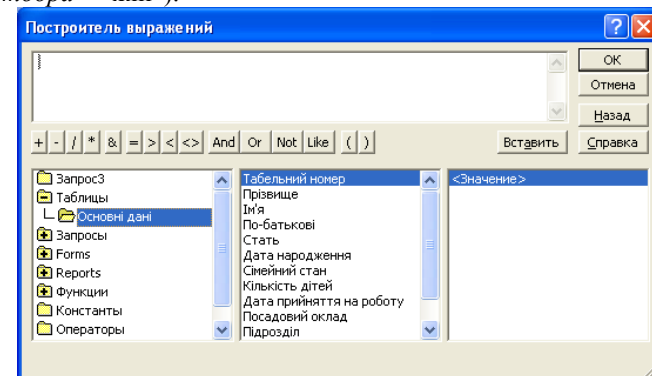




Рис. 8.27. Вигляд вікна побудовувача виразів для запитів оновлення бази.

виберіть: **Запрос, Создать**. У наступному вікні – **Конструктор, Ок**. Далі на вкладці **Таблицы** виберіть назву своєї таблиці, **Добавить, Закрыть**. У пункті меню **Запрос** (або кнопка на панелі інструментів **Тип запроса** ) виберіть команду **Удаление**  – у бланку *Конструктора* запитів з'явиться рядок **Удаление** (рис. 8.25).

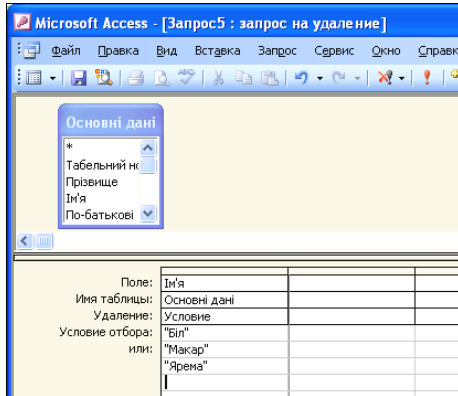



Рис. 8.25. Формування запити на видалення записів з іменами Біл, Макар, Ярема.


Введіть у клітину **Поле:** **Ім'я**, а у клітину **Условие отбора:** – “Біл”. У рядку **или** введіть ім'я Макар, а під ним у наступному рядку Ярема (див. рис. 8.1). Запустіть запит на виконання піктограмою **Запуск** . З'явиться вікно з повідомленням про наступне вилучення записів. Якщо вибрати “Да”, то видалення відбудеться.

Нехай було розформовано один з підрозділів, а всіх співробітників скорочено. Спробуйте видалити записи про цих співробітників. (**Не вилучайте записи остаточно**, вказавши у відповідь на запитання про вилучення “**Нет**”).

Спробуйте вилучити записи про всіх одружених. (**Не видаляйте записи остаточно**, вказавши у відповідь на запитання про вилучення “**Нет**”).

Запит на оновлення

Цей запит дозволяє одночасно змінити групу записів, вибрану на основі заданих критеріїв.

Щоб його створити, у вікні бази даних вибираємо: **Запрос, Создать**, а далі – **Конструктор, Ок**, вибираємо назву таблиці, для якої створюватиметься запит, натискаємо кнопки **Добавить** та **Закрыть**, і нарешті, у меню **Запрос** вибираємо команду **Обновление** . У бланку *Конструктора* запитів з'являється рядок **Обновление**. У нього вводимо нові значення полів.

Використайте запит на оновлення для зміни назви одного з підрозділів (наприклад, “Плановий” змінив назву на “Склад”).

Таким чином, інформаційний процес, який визначається за допомогою Φ , можна досліджувати через рекурентні співвідношення (2.8) [12,17]. Для цього система (2.8) для визначення величин u_k еквівалентно перетворюється:

$$F_k(u_{k_1}, \dots, u_{k_{s_k}}) - u_k = 0, \quad n < k \leq N; \quad k_1, \dots, k_{s_k} < k \quad (2.11)$$

Аналіз інформаційного процесу (2.8) з метою встановлення його чутливості до збурних дій може бути формалізований за допомогою дослідження чутливості задачі про рішення системи (2.11). Для цього розглядається збурена система:

$$F_k(u_k + \Delta u_{k_1}, \dots, u_{k_{s_k}} + \Delta u_{k_{s_k}}) - (u_k + \Delta u_k) = 0, \quad n < k \leq N; \quad k_1, \dots, k_{s_k} < k \quad (2.12)$$

де збурення Δu_k малі. Віднімаючи з (2.12) (2.11) і враховуючи представлення (2.4) для дійсної функції багатьох змінних, з точністю до нескінченно малих другого порядку, отримаємо систему лінійних алгебраїчних рівнянь відносно збурень Δu_k [12]:

$$\sum_{i=1}^{s_k} \frac{\partial F_k(u_{k_1}, \dots, u_{k_{s_k}})}{\partial u_{k_i}} \Delta u_{k_i} - \Delta u_k = 0, \quad n < k \leq N; \quad k_1, \dots, k_{s_k} < k \quad (2.13)$$

Матриця Ψ системи (2.13) — функціональна матриця Якобі функцій

$$y_k = F_k(u_{k_1}, \dots, u_{k_{s_k}}) - u_k = f_k(u_1, \dots, u_N)$$

по змінним u_1, \dots, u_N розміром $m \times N$. Вона зазвичай сильно розріджена. Матриця Ψ має повний ранг, оскільки є нижньою трикутною відносно діагоналі, що проходить через правий нижній кут матриці, на діагоналі знаходяться елементи, які дорівнюють -1. Будемо називати Ψ *варіаційною матрицею інформаційного процесу* (ВМІП) (2.8). Її елементи ψ_{ij} у загальному випадку визначаються як

Варіаційна матриця відіграє важливу роль при аналізі структури й властивостей інформаційного процесу, зокрема, при дослідженні його міри чутливості до збурних дій (похибок вхідних даних).

$$\psi_{ij} = \begin{cases} -1, & \text{якщо } j = i + n, \\ \frac{\partial F_{i+n}}{\partial u_j}, & \text{якщо } j \in \text{одним з чисел } (i+n)_1, \dots, (i+n)_{s_{i+n}}, \\ 0, & \text{в інших випадках} \end{cases}$$

Як уже було відзначено, при визначенні $v = F(u)$ в реальних умовах точне значення v , загалом кажучи, отримати не можна. Замість v буде отриманий елемент $\bar{v} \neq v$. При проведенні прямого аналізу похибки [15,17], результатом якого є оцінка $\|\bar{v} - v\|$, стає зрозумілим, що якщо така оцінка виявиться великою, це може бути пов'язане з нестійкістю оператора F в околі u . У такій ситуації добре зарекомендував себе зворотний аналіз похибок [15,17], основна ідея якого полягає в спробі представити реально отриманий елемент \bar{v} як точний результат перетворення F , але не елемента u , а елемента $\bar{u} \neq u$: $\bar{v} = F(\bar{u})$. Про це вже йшла мова вище. Якщо це вдається зробити, то вплив збурних дій оцінюється величиною $\|\bar{u} - u\|$, а $\bar{u} - u$ називається *еквівалентним збуренням*.

Розглянемо процес поширення похибок під час протікання інформаційного процесу (2.8), реалізація якого зводиться до обчислення функції (2.9) [12]. При реальних обчисленнях точних формул (2.8) мають місце співвідношення:

$$\bar{u}_k = \bar{F}_k(\bar{u}_{k_1}, \dots, \bar{u}_{k_{s_k}}), \quad n < k \leq N, \quad k_1, \dots, k_{s_k} < k, \quad (2.14)$$


$$\bar{u}_k = F_k(\bar{u}_{k_1}, \dots, \bar{u}_{k_{s_k}}) + \eta_k, \quad n < k \leq N, \quad k_1, \dots, k_{s_k} < k,$$

де \bar{F}_k — збурена «близька» до F_k функція, реальне обчислення якої здійснюється при реалізації інформаційного процесу (2.8), \bar{u}_k — реально задана або обчислена величина u_k , η_k — еквівалентна абсолютна похибка (підсумковий результат збурної дії), яка вноситься в результат обчислення F_k .

Представимо (2.14) в виді:

$$\bar{u}_k + \varepsilon_k = F_k(\bar{u}_{k_1} + \varepsilon_{k_1}, \dots, \bar{u}_{k_{s_k}} + \varepsilon_{k_{s_k}}), \quad n < k \leq N, \quad k_1, \dots, k_{s_k} < k, \quad (2.15)$$

де ε_k — збурення \bar{u}_k (ці збурення вносяться й у вхідні дані). Якщо взяти збурені вхідні дані $u_1 + \varepsilon_1, \dots, u_n + \varepsilon_n$ і провести з ними точний процес (2.8), то на кожному кроці цього процесу як точний результат виходить $\bar{u}_k + \varepsilon_k$.

вираз: **[Посадовий оклад]*0,15**. Запустіть запит на виконання **Запуск** . Отримаємо наступний результат (рис. 8.24).

Запрос4 : запрос на выборку			
Табельний номер	Прізвище	Посадовий оклад	Премія
1	Бачинський	123,00 грн.	18,45
2	Ваків	1 500,00 грн.	225
3	Двигун	1 800,00 грн.	270
4	Дідухів	420,00 грн.	63
4	Дідухів	420,00 грн.	63
6	Кожан	1 200,00 грн.	180
7	Мацігін	30 200,00 грн.	4530
8	Мозговий	3 500,00 грн.	525
9	Павлик	21 500,00 грн.	3225
10	Пронів	300,00 грн.	45
11	Кулешник	200,00 грн.	30
14	Дідухів	300,00 грн.	45
*		0,00 грн.	

Рис. 8.24. Фрагмент таблиці з результатами застосування обчислювальних полів у запитих.

Сформуйте довільний запит, що міститиме 2 обчислювальних поля.

Лабораторна робота 8. ЗАПИТИ НА ЗМІНУ ТА ПАРАМЕТРИЧНІ ЗАПИТИ

Запити на зміну використовують для зміни та перенесення даних у таблицях, для оновлення, додавання і видалення груп записів, а також для створення нової таблиці на основі однієї або декількох таблиць.

Є чотири типи запитів на зміну:

- запити на додавання;
- запити на видалення;
- запити на оновлення;
- запити на створення таблиці.

Запит на додавання використовується, якщо потрібно в одну таблицю додати записи з іншої.


Запит на видалення

Запит на видалення використовують для видалення з таблиці груп записів, що задовольняють певну умову.

Для перевірки того, як працює запит на видалення, поверніться до своєї таблиці (клавіша F11 поверне вас у вікно бази даних). Створіть 2 нових записи, у яких заповніть тільки три перші поля: **Табельний номер, Прізвище та Ім'я**. У поле **Ім'я** введіть імена співробітників, які необхідно видалити з бази, наприклад, Біл, Макар, Ярема. Поверніться у вікно бази даних. Далі

Групповые операции, а у всіх клітинах цього рядка, де вибрана назва поля над яким буде здійснюватися операція, – слово “**Групуировка**”. Це слово у даному стовпці потрібно, розкривши список, замінити на назву відповідної функції. У цьому ж стовпці повинен бути наявний прапорець **Вывод на экран**. Крім даного стовпця, на бланку необхідно задати стовпці, що містять умови запиту. Прапорці у рядку **Вывод на экран потрібно зняти**.

Виконайте наступні приклади.

Приклад 1. Знайдемо середній оклад всіх працівників у базі даних, що містить інформацію про кадровий склад установи. Для цього виберемо назву поля **Посадовый оклад**, а у правій частині клітини **Групуировка** клацнемо мишкою, у списку, що відкриється, виберемо **AVG** і запустимо запит на виконання кнопкою **Запуск** . Отримаємо наступний результат (рис. 8.23).

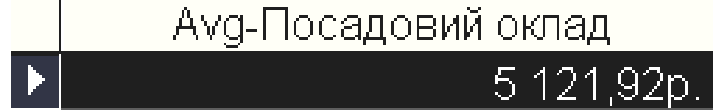


Рис. 8.23. Визначення значення середнього посадового окладу усіх працівників установи

Приклад 2. Визначити середній оклад жінок.

Діємо так, як і у прикладі 1, тільки заповнимо ще один стовпчик **Стать**. У ньому заповнимо лише клітину **Условие отбора**, ввівши значення "жіні". У цьому стовпці потрібно також зняти прапорець **Вывод на экран**.

Виконайте самостійно наступні завдання.

Визначити, яку максимальну кількість дітей має співробітник даної фірми.

Визначити кількість наявних фото (не порожніх полів) у таблиці бази даних.

Визначити найменший оклад у жінок.

Визначити найбільший оклад в одружених чоловіків.

Визначити кількість одружених співробітників.

Визначити кількість одружених чоловіків, що мають більше 2 дітей.

Визначити сумарний оклад всіх працівників одного з підрозділів, наприклад, бухгалтерії.

Визначити, скільки жінок має задане ім'я, наприклад, "Тетяна".

Визначити середній оклад співробітників, молодших ніж 25 років.

Визначити середній оклад співробітників заданого підрозділу, що працюють у даній установі не менше 6 років.

Обчислювальне поле.

Створіть обчислювальне поле для визначення розміру премії. (Нехай її виплатили всім співробітникам установи, і вона становить 15% від посадового окладу.) Для цього створіть запит на вибірку, розмістіть на бланку запиту усі поля і додайте ще один стовпчик з назвою **Премия**, у рядок **Поле** якого введіть

Сукупність значень ε_k , в якій еквівалентні збурення вхідних даних дорівнюють 0, описує прямий аналіз похибок. Якщо в множині ε_k еквівалентні збурення вихідних даних дорівнюють 0, то маємо зворотний аналіз похибок. Інші варіанти ε_k описують мішаний аналіз похибок [12].

З (2.14), (2.15) отримуємо

$$\varepsilon_k = F_k(\bar{u}_{k_1} + \varepsilon_{k_1}, \dots, \bar{u}_{k_{s_k}} + \varepsilon_{k_{s_k}}) - F_k(\bar{u}_{k_1}, \dots, \bar{u}_{k_{s_k}}) - \eta_k, \quad n < k \leq N, \quad k_1, \dots, k_{s_k} < k. \quad (2.16) \quad (0.10)$$

Якщо збурення вхідних даних $\varepsilon_1, \dots, \varepsilon_n$ відомі, то за допомогою (2.16) можна визначити інші збурення $\varepsilon_k, k > n$ (значення u_k , збурення η_k визначаються реалізацією інформаційного процесу (2.14)), що дасть можливість для встановлення міри чутливості процесу до похибок вхідних даних (до збурних дій). Розглянемо (2.16) докладно.

Система (2.16) в загальному випадку є нелінійною відносно ε_k . Враховуючи представлення (2.4) для дійсної функції багатьох змінних, (2.16) можна замінити лінійною системою:

$$\varepsilon_k = \sum_{i=1}^{s_k} \frac{\partial F_k(\bar{u}_{k_1}, \dots, \bar{u}_{k_{s_k}})}{\partial u_{k_i}} \varepsilon_{k_i} - \eta_k, \quad n < k \leq N, \quad k_1, \dots, k_{s_k} < k,$$

Замінімо реально обчислені величини $\bar{u}_i, i = k_1, \dots, k_{s_k}$, на точні:

$$\sum_{i=1}^{s_k} \frac{\partial F_k(u_{k_1}, \dots, u_{k_{s_k}})}{\partial u_{k_i}} \varepsilon_{k_i} - \varepsilon_k = \eta_k, \quad n < k \leq N, \quad k_1, \dots, k_{s_k} < k \quad (2.17)$$

Матриця системи (2.17) — це ВМП (2.8), тому система завжди сумісна. Таким чином, величини збурень ε_k визначаються як рішення системи лінійних алгебраїчних рівнянь (2.17), властивості якої визначаються властивостями ВМП.

Оскільки ε_k використовуються для аналізу інформаційного процесу з метою встановлення його чутливості до збурних дій, важливо, щоб ці значення були отримані якнайточніше, для чого матриця системи (2.17) — ВМП (2.8) — повинна бути добре обумовленою.

ВМП відіграє важливу роль не тільки при аналізі збурень параметрів, що визначають інформаційний процес.

2.3. Граф інформаційного процесу і його особливості

Довільному інформаційному процесу поставимо у відповідність орієнтований граф (орграф) наступним чином. Зіставимо k -й вершині графа отримання величини u_k . Перші n вершин будуть символізувати введення початкових даних u_1, \dots, u_n і називатися вхідними, а інші вершини — обчислення u_k як значень функцій F_k з (2.8). Будемо вважати, що дуга йде з i -ї вершини в j -у тоді й тільки тоді, коли при обчисленні величини u_j величина u_i використовується як аргумент. Відповідно до (2.8) дуги не будуть входити в k -у вершину, якщо $k \leq n$. Якщо $k > n$, то в k -у вершину будуть входити дуги з вершин з номерами k_1, \dots, k_{s_k} .

Будемо називати орграф *дводольним*, якщо множина його вершин V може бути розбита на дві підмножини V_1 і V_2 так, що кожне ребро графа є впорядкованою парою виду $\langle v_1, v_2 \rangle$, де $v_1 \in V_1$, а $v_2 \in V_2$.

Зі способу побудови графа стає очевидною істинність наступних тверджень.

Твердження 2.1. Граф інформаційного процесу є ациклічним.

Теорема 2.2. Вихідні параметри інформаційного процесу будуть незалежними тоді й тільки тоді, коли граф інформаційного процесу буде дводольним.

В графі, як моделі інформаційного процесу, наочно представлені відомості про те, як окремі перетворення при протіканні процесу пов'язані між собою інформаційно, які перетворення в ході їх моделювання можуть виконуватися одночасно, які потрібно виконувати пізніше або раніше, чим інші і т.д. Граф інформаційного процесу описує всю картину поширення інформації при його протіканні, а тому може бути використаний для аналізу інформаційного процесу в цілому, його структури.

Отриманому графу ставиться в співвідношення матриця Φ розміром $m \times N$ з елементами φ_{ij} :

$$\varphi_{ij} = \begin{cases} -1, & \text{якщо } j = i + n, \\ 1, & \text{якщо } j \in \text{одним з чисел } (i + n)_1, \dots, (i + n)_{s_{i+n}}, \\ 0, & \text{в інших випадках} \end{cases}$$

Очевидно, k -ий стовпець матриці Φ відповідає параметру u_k , а k -ий рядок — параметру u_{k+n} . В k -ому рядку елемент -1 знаходиться в тому стовпці, номер якого відповідає номеру параметра, що обчислюється — u_{k+n} . Елементи $+1$ знаходяться у тих стовпцях, номери яких відповідають номерам аргументів параметра, що обчислюється — u_{k+n} . Матриця Φ описує зв'язок параметрів u_k між собою і називається *матрицею інформаційної зв'язності*

Утворіть запит для пошуку записів про всіх співробітників, що мають 1 або 2 дітей. (Умова формується так, як і для розширеного фільтра.)

Утворіть запит для відшукування записів про всіх співробітників, що не мають дітей або мають їх більше ніж 5.

Утворіть запит для пошуку записів про всіх співробітників, що народилися між 1960 та 1970 роками. (Умову можна задати, як і для розширеного фільтра)

Утворіть запит для знаходження записів про всіх співробітників, що народилися між 1965 та 1970 або між 1956 та 1959 роками.

Утворіть запит для виведення на екран записів про всіх одружених співробітників (у поле *Условие* введіть: *Истина*).

Утворіть запит для відшукування записів про всіх одружених чоловіків, що мають посадовий оклад менший ніж 2000 грн.

Утворіть запит для пошуку записів про всіх неодружених жінок, молодших ніж 25 років, що мають посадовий оклад більший ніж 1800 грн.

Змініть останній запит так, щоб він стосувався заміжніх жінок, що мають більше ніж 2 дітей і посадовий оклад в межах від 2000 та 3500 грн. Для цього:

у вікні БД перейдіть на вкладку *Запросы* і відкрийте останній запит;

перейдіть у режим Конструктора запитів (кнопкою або меню);

змінити умови (видаліть непотрібні і введіть нові);

запустіть запит на виконання.

Відкрийте один із запитів і впорядкуйте його так, щоб прізвища розміщувалися за алфавітом. Для цього у стовпці *Прізвище*, у клітині *Сортировка* виберіть зі списку, що розкривається, команду *По возрастанию*.


Лабораторна робота 7.

ОБЧИСЛЕННЯ У ЗАПИТІ (Групувальні запити)

Запити можна використовувати для виконання розрахунків на основі даних таблиць. Такі запити називають підсумковими. Для цього призначено статистичні функції. Це **SUM** (сума значень полів тих записів, що відібрано запитом), **AVG** (середнє арифметичне значень полів тих записів, що відібрано запитом), **MIN** (мінімальне зі значень полів тих записів, що відібрано запитом), **MAX** (максимальне зі значень полів тих записів, що відібрано запитом), **COUNT** (кількість непорожніх значень поля у записах, що відібрано запитом), **FIRST** (визначає перше значення у даному полі), **LAST** (визначає останнє значення у даному полі), **STDEV** (стандартне відхилення значень даного поля для всіх записів, що відібрано даним запитом), **VAR** (варіація значень даного поля для всіх значень, відібраних запитом).

Щоб здійснити розрахункову операцію, потрібно у вікні *Конструктора* запиту вивести назву поля, над яким здійснюватиметься операція і натиснути на панелі інструментів кнопку *Групповые операции* Σ (або вибрати у меню *Вид* команду *Групповые операции*). У бланку з'явиться ще один рядок —

Зауваження.1 Між бланком запиту і таблицею переключаємось за

допомогою кнопки  або команди **Вид | Конструктор**.

Зауваження 2. Можна запит на вибірку робити інакше (у вікні БД вибрати **Запросы, Создать**; потім **Простой запрос, Ок**; у списку **Таблицы/Запросы** вибрати назву своєї таблиці й вибрати всі поля (>>). Потім перейти у режим Конструктора, скасувати появу непотрібних полів на екрані (забрати прапорець) і ввести потрібні умови. Далі діяти аналогічно до попереднього пункту.

Утворіть запит для відшукування всіх записів, у яких у полі **Прізвище** розміщено одне з двох заданих вами прізвищ. Нехай, наприклад, це прізвища: "Кожан" та "Павлик". Тоді умова формується так, як на рис. 8.22. А у інших стовпцях бланка запиту слід задати умови виводу на екран інших полів (тобто встановити прапорці).

Поле:	Прізвище
Имя таблицы:	Основні дані
Сортировка:	
Вывод на экран:	<input checked="" type="checkbox"/>
Условие отбора:	"Бойко"
или:	"Пучко"

Рис. 8.22. Завдання умови пошуку у бланку запиту на вибірку.

Утворіть запит для пошуку всіх записів про жінок, що працюють у заданому підрозділі. Наприклад, нехай цим підрозділом є бухгалтерія. Тоді умова формується так, як на рис. 8.22.

Поле:	Стать	Підрозділ
Имя таблицы:	Основні дані	Основні дані
Сортировка:		
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Условие отбора:	"жін"	"Бухгалтерія"
или:		

Рис. 8.22. Завдання умови пошуку у бланку запиту на вибірку.

Сформуйте запит для виведення на екран всіх записів про чоловіків, що працюють в одному з двох заданих підрозділів. (Заповнюється ще клітина у рядку **Или**).

Утворіть запит для пошуку всіх записів про співробітників, чие прізвище починається на задану букву. (Умова формується так, як для розширеного фільтра.)

Утворіть запит для знаходження всіх записів про жінок, чие прізвище починається на одну з двох заданих букв. (Умови будуть у стовпцях **Стать** та **Прізвище**. Під час цього у стовпці **Прізвище** потрібно заповнити клітини **Условие отбора** та **Или**).

інформаційного процесу (МІЗІП) (2.8). Очевидний зв'язок матриці зв'язності з ВМІП: структури ненульових елементів обох матриць повністю співпадають.

Для графа інформаційного процесу по МІЗІП заміною ненульових елементів якимись числами можна отримати нескінченну сукупність матриць, кожна з яких є *зв'язеною* МІЗІП. Будь-яка зв'язена МІЗІП, окремим випадком якої є й ВМІП, дозволяє однозначно відновити граф інформаційного процесу, а тому може використовуватися для його аналізу.

Для орієнтованого графа, що відповідає інформаційному процесу, стандартно визначається $N \times N$ -матриця суміжності B з елементами b_{ij}

$$b_{ij} = \begin{cases} 1, & \text{якщо з } i\text{-ої вершини в } j\text{-у йде ребро,} \\ 0, & \text{в інших випадках} \end{cases}$$

і матриця інцидентності A з елементами a_{ij}

$$a_{ij} = \begin{cases} 1, & \text{якщо } j\text{-е ребро виходить з } i\text{-ої вершини} \\ -1, & \text{якщо } j\text{-е ребро входить в } i\text{-у вершину,} \\ 0, & \text{в інших випадках} \end{cases}$$

Матриця B тісно пов'язана з МІЗІП Φ : Φ — ере нумера, яка складається з останніх m рядків матриці $B^T - I$, де I — одинична матриця відповідного розміру. У зв'язку з цим для приведення МІЗІП до більш «придатного» з погляду можливостей її обробки виду за рахунок ере нумерації рядків і стовпців можна здійснювати відповідну ере нумерації для матриці суміжності.

До значного зменшення часових витрат, необхідних для дослідження властивостей інформаційного процесу, приводить виявлення й наступне використання його внутрішнього паралелізму на основі відповідного графа [12,17], тобто можливостей паралельного (одночасного) виконання вхідних до інформаційного процесу операцій.

2.4. Паралельні форми інформаційного процесу

Поява паралельних обчислювальних систем і впровадження їх у практику рішення великих прикладних задач привела до необхідності аналізу інформаційних процесів з метою визначення можливості їх математичної формалізації й обробки в паралельних обчислювальних системах. Результатом аналізу повинне стати виявлення таких частин процесу, які інформаційно між

собою не пов'язані. Якщо такі частини знайдені, будемо говорити, що інформаційному процесу властивий *внутрішній паралелізм*.

Будемо використовувати граф $G = (V, E)$ інформаційного процесу, де V – множина вершин, а E – множина впорядкованих пар вершин (ребер), для аналізу його структури, не накладаючи ніяких обмежень на вид вхідних і вихідних параметрів. Граф інформаційного процесу не накладає, загалом кажучи, ніяких обмежень і на порядок виконання операцій, що входять до складу процесу, крім одного: до моменту початку реалізації будь-якої операції повинні закінчити своє виконання всі ті операції, які поставляють для неї параметри-аргументи. Таким чином граф інформаційного процесу визначає множину припустимих порядків виконання його операцій.

Будь-який інформаційний процес – це процес, що протікає в часі, будь-яка його реалізація породжує певне сортування вхідних до його складу операцій. Це сортування буде розбивку множини операцій (вершин відповідного графа) на такі групи, які виконуються послідовно, а операції всередині групи можуть виконуватися одночасно.

Ототожнюючи інформаційний процес з його графом, будемо припускати, що в графі відображені операції отримання всіх параметрів і зв'язки, вплив яких на реалізацію процесу підлягає вивченню.

Нехай $G = (V, E)$ — довільний орієнтований ациклічний граф з n вершинами. Тоді [12] існує таке натуральне число $s \leq n$, що всі вершини графа можна так помітити одним з індексів $1, 2, \dots, s$, що якщо ребро йде з вершини з індексом i в вершину з індексом j , то

$$i < j. \quad (2.18)$$

Така розмітка вершин називається *топологічним сортуванням* графа або *паралельною формою*. Очевидно, що ніякі дві вершини з однаковим індексом не є суміжними. Крім того, для будь-якого натурального $s \leq n$, більшого довжини критичного шляху, існує топологічне сортування, при якому використовуються всі s індексів, тобто граф має не єдине топологічне сортування.

Якщо співвідношення (2.18) замінити на $i \leq j$, то отримане сортування буде називатися *узгальненим топологічним сортуванням*.

Результатом топологічного сортування є виявлення можливостей паралельного (одночасного) виконання (аналізу) вхідних в інформаційний процес операцій. Ці операції будуть входити в одну групу сортування, яка називається *ярусом паралельної форми*. Сукупність всіх топологічних сортувань графа інформаційного процесу визначає його паралельні форми реалізації (обробки, аналізу). Операції, відповідні до вершин графа одного рівня топологічного сортування, є інформаційно незалежними, а тому можуть виконуватися паралельно. Групи операцій, що відповідають різним топологічним рівням, виконуються послідовно в порядку зростання номерів вершин графа, що входять у них.

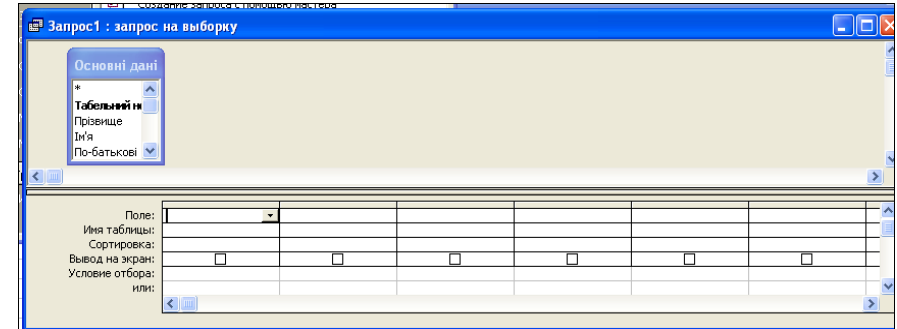


Рис. 8.20. Вигляд вікна *Конструктора* запитів.

у вікні *Конструктора* запитів побачимо список полів і нижче бланк запиту. Заповніть у бланку стільки стовпчиків, скільки полів хочемо бачити на екрані, наприклад, усі. Робимо це наступним чином:

кожен стовпчик у клітині *Поле* повинен містити назву поля. Клацніть кнопку для розкриття списку підстановок і виберіть зі списку назву першого поля “Табельний номер”, у наступному стовпці аналогічно розмістимо “Прізвище” і т.д.;

у рядку *Вивод на екран* розміщуються прапорці, якщо відповідне поле потрібно виводити на екран;

у стовпці, що містить поле *Стать*, у клітині в рядку *Умовие отбора* введемо: “жін” (лапки непотрібні);

натиснемо на кнопку *Запуск* панелі інструментів (на ній – знак оклику) або задамо команду *Запуск* з меню *Запрос*.

Запит на вибірку створено – на екрані з’являться відібрані записи (рис. 8.21).

Табельний нс	Прізвище	Ім'я	По-батькові	Стать	Дата народж
4	Дідухів	Галина	Іванівна	жін	01.01.199
5	Залужна	Христина	Семенівна	жін	03.03.199
7	Мацігін	Григорій	Петрович	жін	05.05.199
10	Пронів	Олена	Василівна	жін	08.08.199

Рис. 8.21. Результат застосування запиту на вибірку з бази даних *Кадрі*.

Перегляньте їх. Закрийте вікно і, якщо потрібно, збережіть запит, тобто натискаємо кнопку “Да” у діалоговому вікні, що з’явиться, і вводим назву (наприклад, Запит1). Завершуємо роботу натиском кнопки *Ок*. Після цього відбудеться повернення у вікно бази даних.

Таким чином, запит на вибірку створено. Зауважимо, що умови на вибірку формуються так, як і для розширеного фільтра.

Виведіть на екран записи про всіх чоловіків, чиє прізвище починається заданою буквою.

Виведіть на екран записи про всіх незаміжніх жінок, що мають дітей і мають оклад більший за 3000 грн.

Виведіть на екран записи про всіх, хто має одне з двох заданих прізвищ.

Виведіть на екран записи про всіх, хто працює у заданому підрозділі і має 3 або 4 дітей.

Лабораторна робота 6.

СТВОРЕННЯ ТА МОДИФІКАЦІЯ ЗАПИТІВ НА ВИБІРКУ

Запит на вибірку – найпростіший тип запиту. Він виводить на екран записи, вибрані з таблиці. Запит на вибірку не змінює саму базу даних.

Розглянемо приклад створення запиту у режимі *Конструктора* для таблиці з бази даних, що містить дані про кадровий склад підприємства (рис. 8.19).

Табельний нс	Прізвище	Ім'я	По-батькові	Стать	Дата народж	Сімейний ста	К
1	Мачинський	Назар	Романович	Чол	12.01.1960	одруж	
2	Маків	Дмитро	Васильович	чол	11.03.1960	неодруж	
3	Двигун	Сергій	Борисович	чол	12.04.1990	одруж	
4	Дідухів	Галина	Іванівна	жін	01.01.1991	неодруж	
5	Залужна	Христина	Семенівна	жін	03.03.1990	одруж	
6	Кожан	Лев	Маркович	чол	04.04.1990	одруж	
7	Мацігін	Григорій	Петрович	жін	05.05.1985	неодруж	
8	Мозговий	Іван	Іванович	чол	06.06.1960	одруж	
9	Павлик	Віктор	Климович	чол	07.07.1970	одруж	
10	Пронів	Олена	Василівна	жін	08.08.1960	одруж	
*							

Рис. 8.19. Фрагмент бази даних **Кадри**, що містить інформацію про кадровий склад.

Нехай потрібно створити запит на вибірку записів про всіх жінок, що працюють у даній установі.

Для виконання поставленого завдання виконаємо наступні дії:

у меню **Файл** виберемо команду **Открить** і введемо ім'я бази даних про кадровий склад фірми;

у вікні бази даних на вкладці **Запросы** натиснемо кнопку **Создать**.

у наступному вікні виберемо: **Конструктор** і натиснемо **Ок**;

у вікні **“Добавление таблицы”** вибираємо назву своєї таблиці (наприклад, **“Основні дані”**) і натискаємо кнопки **Добавить** та **Закрить**;

отримаємо вікно **Конструктора** запитів (рис. 8.20);

Як правило, чим складніше граф, чим більший його розмір, тим важче побудувати його топологічне сортування. Зменшення необхідного часу для цього процесу може бути досягнуте за рахунок розбивки графа на підграфи меншого розміру з наступною побудовою топологічних сортувань ере номер і відновленням сортування всього графа по сортуваннях ере номер [12] (відповідний алгоритм наведено в додатку 3). Крім того, для скорочення часу для аналізу інформаційного процесу зменшити розмір графа можна за допомогою гомоморфної згортки його ере номер.

За допомогою ере нумерації вершин графа інформаційного процесу можна спростити його опис. Отже, шляхом ере нумерації операцій (2.8) можна спробувати привести МІЗІП і відповідно ВМІП до більш зручного виду.

Розглянемо довільне топологічне сортування вершин графа інформаційного процесу. Позначимо вершини наступним чином: спочатку нумеруються вершини, що потрапили в перший ярус, потім у другий і т.д., визначаючи тим самим порядок рядків МІЗІП. При впорядкуванні стовпців спочатку нумеруються стовпці, які відповідають аргументам F_1 , потім стовпці, що відповідають лише тим аргументам F_2 , які ще не були занумеровані, і т.д. При цьому нова нумерація стовпців, відповідних до обчислюваних величин, береться такою, щоб з точністю до викидання стовпців, відповідних до входних параметрів, вона збігалася з новою нумерацією рядків. Після описаного переставлення рядків і стовпців МІЗІП буде мати вигляд, представлений на рис.2.1. В кожному рядку P_i існує хоча б один ненульовий елемент, і всі її ненульові елементи дорівнюють 1. В кожному рядку зафарбованої частини знаходиться лише один ненульовий елемент — 1.

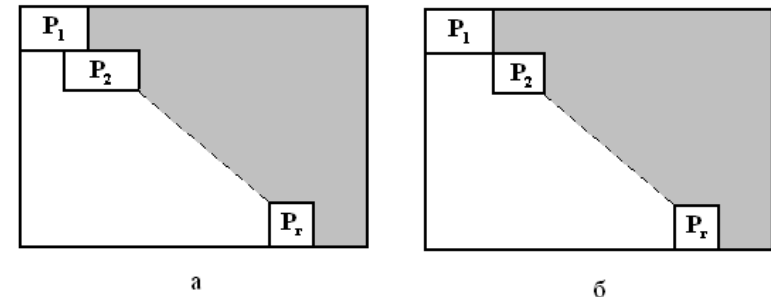


Рис.2.1. Структура МІЗІП, коли в інформаційному процесі розмноження інформації присутнє (а); відсутнє (б).

У загальному випадку в кожному стовпці МІЗІП може знаходитися більш однієї одиниці: в l -ому стовпці буде m_l одиниць, якщо m_l операцій, що входять в інформаційний процес, використовують параметр u_l як один з своїх аргументів, тобто якісь ребра, що виходять з деяких вершин графа інформаційного процесу, визначають перенесення однієї інформації. У цьому

випадку будемо говорити, що має місце *розмноження інформації*. Якщо розмноження інформації присутнє в інформаційному процесі, то деякі стовпці МІЗП можуть мати непусті перетинання більш ніж з одним рядком. При будь-якому переставленні рядків і стовпців ця властивість зберігається. Отже, описані вище переставлення приведуть МІЗП до виду, зображеного на рис.2.1(а), інакше її вид представлений на рис.2.1(б).

Для формалізації аналізу інформаційного процесу у відповідність його графу можна поставити не тільки одну з розглянутих вище двовимірних матриць, але й одновимірний вектор.

Однієї з характеристик протікання інформаційного процесу є час його виконання. При безпосередній реалізації визначені моменти виконання всіх операцій (2.8). Порушення певного часу виконання деякої операції в ході процесу є сигналом можливих збоїв, атак, спрямованих на засоби захисту, що забезпечують протікання процесу, і т.п.

Перенумеруємо вершини графа інформаційного процесу довільно й кожній i -й вершині поставимо в співвідношення час t_i закінчення виконання відповідної операції. Таким чином, з інформаційним процесом можна зв'язати вектор $t = (t_1, t_2, \dots, t_n)^T$, який будемо називати *вектором часового розгортання* процесу [12], який показує, як протікає процес у часі. Для елементів вектору t природно визначити деякі обмеження, наприклад, задати час h_j для реалізації j -ї операції. Якщо в графі інформаційного процесу існує ребро, що йде з i -ї вершини в j -у, то повинне виконуватися співвідношення: $t_j - t_i \geq h_j$.

Дійсно, час, що проходить від закінчення i -ї операції до закінчення j -ї, містить у собі не тільки час виконання j -ї операції, але й час, який затрачується на передачу інформації, необхідної для виконання j -ї операції. Невід'ємний вектор $h = (h_1, h_2, \dots, h_n)^T$ будемо називати *вектором реалізації інформаційного процесу*.

З кожним ребром графа можна зв'язати не тільки якусь інформацію, що передається від однієї вершини до іншої. Будь-яке часове розгортання однозначно визначає час t_i появи цієї інформації й час $t_j - t_i$ її існування, частково в незмінному, частково в перетвореному в j -й вершині виді. Можна вважати, що в момент t_j стара інформація повністю закінчує своє існування й народжується нова. Час $t_j - t_i$ — *час затримки інформації на дузі*, яка пов'язує i -у і j -у вершини. При реалізації інформаційних процесів на часи затримок накладаються обмеження знизу. Вони спричиняються часом передачі інформації по каналах і лініях зв'язку, часом зберігання інформації й іншими факторами. Ці обмеження можна задавати аксіоматично й вважати, що час затримки інформації на дузі, що зв'язує i -у і j -у вершини, не менше невід'ємного числа w_{ij} :

$$t_j - t_i \geq w_{ij}$$

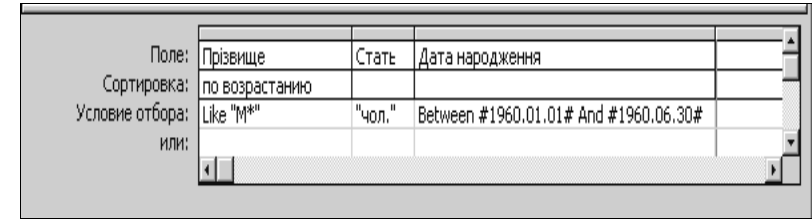


Рис. 8.16. Бланк розширеного фільтра.

у першому стовпці розмістити назву поля **Прізвище** та умову, яка складається з літери "М" і зірочки (програма сама додасть слово "Like" і лапки); у другому стовпці розмістити назву поля **Стать** та умову, що складатиметься зі слова "чол." (вводити можна без лапок, їх допише програма); у третьому стовпці умова вводиться, як на рис. 8.17 або: $>\#1.1.60\#$ and $<\#30.6.60\#$ (символ "#" можна не вводити – програма у даному випадку допише його сама).

таблица						
Прізвище	Ім'я	По-батькові	Стать	Дата народж	Сімейний ста	Кількість дітє
Мачинський	Назар	Романович	чол	12.01.1960	одруж	2
Маків	Дмитро	Васильович	чол	11.03.1960	неодруж	0
Мозговий	Іван	Іванович	чол	06.06.1960	одруж	2
						0

Рис. 8.17. Результат застосування розширеного фільтра, поданого на рис. 8.16.

Слід зауважити, що деякі версії програми вимагають вводити дати не з крапкою, а з символом "/"). Якщо використати вбудовані функції YEAR та MONTH, що визначають рік та місяць заданої дати, то бланк розширеного фільтра можна заповнити й так, як подано на рис. 8.18. Слова "Выражение 1" та "Выражение 2" дописує програма, а вводити потрібно функції Year ([Дата народження]) та Month ([Дата народження]).

Прізвище	Стать	Выражение1: Year([Дата народження])	Выражение2: Month([Дата народження])
по возрастанию			
Like "М*"	"чол."	1960	>=1 and <=6

Рис. 8.18. Застосування функцій Year та Month.

Завдання.

Виведіть на екран записи про всіх, хто народився між 1950 та 1955 роками.

Виведіть на екран записи про всіх, у кого ім'я починається однією з трьох заданих літер.

Скасуйте дію фільтра, натиснувши на ту ж кнопку (що змінить назву на **Удалити фільтр**) або командою **Записи, Удалити фільтр**.

Відфільтруйте записи так, щоб були видимими записи про співробітників, що мають двох або трьох дітей.


Виведіть на екран записи про всіх, хто не має дітей і працює у заданому підрозділі.

Виведіть на екран записи про всіх, у кого ім'я починається на задану літеру (використайте символ “*”).

Виведіть на екран записи про всіх, хто має одне з трьох заданих імен.

Розширені фільтри

Розширений фільтр задається командою **Записи, Фільтр, Раширенний фільтр**. Після цього потрібно заповнити бланк у вікні розширеного фільтра (рис. 8.15).

Починають створення фільтра натиском кнопки панелі інструментів **Очистити бланк** . Потім відкривають у кожному стовпці таблиці, у рядку **Поле**, список підстановок і вибирають назву того поля, яке бере участь в умові фільтрування. Самі умови записують у рядку **Условие отбора**. Для впорядкування записів за вибраним полем у цьому ж стовпці розкривають ще один список підстановок у рядку **Сортировка** і задають вид впорядкування – за зростанням (**по возрастанию**) чи спаданням (**по убыванию**).

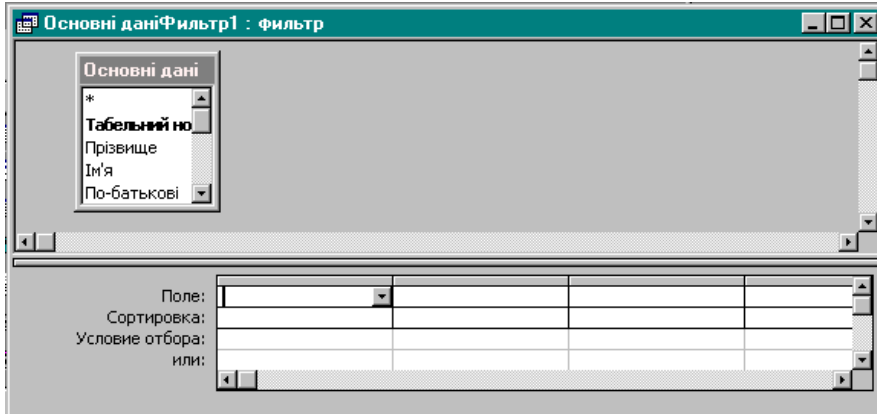


Рис. 8.15. Вікно розширеного фільтра.

Наприклад, для створення розширеного фільтра, за яким виводитимуться на екран записи про чоловіків, що народилися у першій половині 1960 року і прізвище яких починається з літер "М", слід заповнити бланк так (рис. 8.16):

Вектор w з координатами w_j називається **вектором затримок**.

Нехай інформаційний процес починає свою реалізацію в нульовий момент часу, і в кожний додатний цілочисловий момент виконується хоча б одна операція (2.8). Розглянемо відповідне часове розгортання. Згідно з розгортанням множина вершин графа розбивається на неперетинні підмножини, де в одну підмножину входять ті й тільки ті вершини, які відповідають операціям, що виконуються одночасно. Кожній з побудованих підмножин приписується індекс, що дорівнює моменту виконання відповідних операцій. Очевидно, що отримана розбивка вершин графа інформаційного процесу визначає деяку паралельну форму його протікання. Вірно й те, що будь-яка паралельна форма інформаційного процесу породжує деяке часове розгортання.

Таким чином, очевидно, що аналіз часових розгортань є перспективним напрямком в галузі досліджень інформаційних процесів.

2.5. Введення у функціональне дослідження структури інформаційного процесу

Представлення часових розгортань інформаційного процесу у вигляді векторів відповідає математичній формалізації процесу в виді (2.8), де виконувані операції ідентифікуються одним цілим параметром. Така ідентифікація при практичній використанні (2.8) приводить до значних незручностей і труднощів.

Будемо вважати вершини графа інформаційного процесу точками скінченновимірному простору (з можливою в деякому розумінні «зручною» цілочисловою (хоча це необов'язково) фіксацією координат). Множина вершин буде знаходитися в деякій області D , а часові розгортання можна розглядати як функції, визначені на множині точок-вершин. Таким чином, якщо задане часове розгортання, то кожній з точок-вершин x ставиться у відповідність число, яке дорівнює часу виконання відповідної операції, тобто часове розгортання — це деяка функція

$$t(x): D_d \rightarrow K, \quad (2.19)$$

де D_d — область визначення, що є дискретною множиною точок x з області D , $K \subseteq Z$ — область значень — дискретна множина, де Z — множина цілих чисел. Часові розгортання, представлені в такому виді, будемо називати **просторово-часовими**.

Будь-яка паралельна форма інформаційного процесу визначає деяку множину часових розгортань. При завданні розгортання у вигляді (2.19) яруси паралельної форми будуть мати явний геометричний зміст: вони визначаються поверхнями рівня (визначення 2.14) для $t(x)$ —

$$t(x) = const. \quad (2.20)$$

Дійсно, нехай множина точок (2.20) містить якісь вершини графа інформаційного процесу. Тоді відповідні вершинам операції виконуються в один момент, тобто утворюють ярус паралельної форми.

Для зручності подальшого аналізу, не обмежуючи значно спільності міркувань, продовжимо яким-небудь прийнятним способом просторово-часові розгортання (2.19) на всю область D . Позначимо відповідні функції $t_D(x)$. Будемо вважати, що функції $t_D(x)$ мають необхідну гладкість.

Нехай D — однозв'язна область. Поверхні рівня $t_D(x) = const$ можуть бути як одно- так і многозв'язними (рис.2.2 — цифрами відзначені поверхні рівнів, відповідні до послідовних значень t), однак, завдяки припущенням про

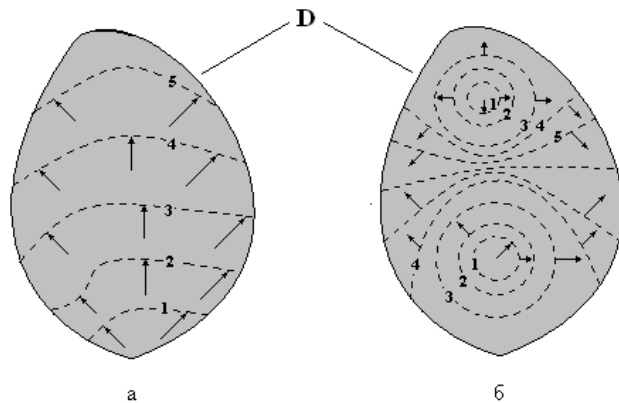



Рис.2.2. Системи поверхонь рівня: однозв'язні (а); многозв'язні (б).


гладкість функцій $t_D(x)$, з зміною t вони змінюються неперервно, тому по один бік від них завжди будуть знаходитися вершини, відповідні до операцій, що виконуються до часу t , а по інший — вершини, відповідні до операцій, що виконуються після часу t . Тоді перенесення інформації від вершин однієї групи до вершин іншої групи здійснюється лише тими ребрами графа інформаційного процесу, які для будь-якої поверхні рівня просторово-часового розгортання визначають орієнтований розріз графа процесу [12]. Поверхні рівнів показують розподіл потоків інформації в D .

Таким чином, опис паралельних форм інформаційного процесу зводиться до опису поверхонь рівня просторово-часових розгортань і дослідженню функцій, що їх визначають, і накреслює новий напрямок в області аналізу інформаційних процесів.

зробимо активною одну з клітин таблиці, що містить слово “жін”, і натиснемо кнопку панелі інструментів **Фільтр по виділенному** , на екрані з'явиться відфільтрована таблиця (рис. 8.14);

і дані : таблиця					
Ідентифікаційний нс	Прізвище	Ім'я	По-батькові	Стать	Дата народження
4	Дідухів	Галина	Іванівна	жін	01.01.1991
5	Залужна	Христина	Семенівна	жін	03.03.1990
10	Пронів	Олена	Василівна	жін	08.08.1960

Рис. 8.14. Фрагмент результату застосування фільтра за виділенням.

повернутись до попереднього стану можна натиском кнопки панелі інструментів **Удалить фільтр** .

Відфільтруйте записи так, щоб видимими були тільки записи про співробітників, які працюють у якомусь одному підрозділі (тобто вміст поля **Підрозділ** один і той же). Скасуйте дію фільтра.

Виведіть на екран усі записи, крім тих, що стосуються працівників бухгалтерії. Це означає, що потрібно вивести на екран усі записи, крім виділеного. Тоді:

зробимо активною клітину зі словом: “Бухгалтерія”;

задаємо команду **Записи, Фільтр, Ісключить выделенное**.

Виведіть на екран усі записи про чоловіків.

Виведіть на екран записи про всіх, хто працює у заданому підрозділі.



Виведіть на екран записи про всіх, хто має двох дітей.


Виведіть на екран записи про всіх, хто має ім'я, що починається на перші дві задані літери.


Виведіть на екран записи про всіх, крім тих, у кого прізвище починається на букву Б.

Звичайний фільтр

Відфільтруйте записи так, щоб видимими були тільки записи про співробітників, які працюють в одному з двох заданих підрозділів. Для цього потрібно у меню **Записи** вибрати **Фільтр, Изменить фильтр** або на панелі

інструментів піктограму . З'явиться вікно для створення звичайного фільтра. Для відміни дії попередніх фільтрів потрібно натиснути на кнопку **Очистить бланк** .

Далі зробіть активною будь-яку клітину у полі **Підрозділ**, клацніть кнопку для розкриття списку підстановок  і виберіть зі списку, що випадає, одну з назв підрозділу. Щоб вибрати назву ще одного підрозділу потрібно клацнути по кнопці **ИЛИ** (у нижній лівій частині вікна) і вибрати у списку підрозділів іншу назву місця праці. Залишилися застосувати

цей фільтр, клацнувши по кнопці **Применение фильтра**  (це та сама кнопку що використовувалася для видалення фільтра).

Створення таблиці та внесення змін у таблицю в режимі Конструктора

Створіть нову базу даних, а у ній – таблицю у режимі *Конструктора*. Під час цього :

таблиця повинна мати 5 полів;


для задання типу одного з полів потрібно використати *Майстер* підстановок;

одне з полів повинно бути ключовим;

одне з полів треба зробити індексованим (за ним, як правило, відбувається впорядкування та пошук);

на значення одного з полів накладіть умову.

Змініть ширину стовпця, що містить назви полів (як і ширину стовпця у режимі таблиці).

Доповніть таблицю ще одним записом. Записи додаються лише в кінці таблиці. Для цього потрібно знаходитись в будь-якому місці бази і виконати команди **Вставка, Новая запись** або скористатися піктограмою  на панелі інструментів.

Вилучіть щойно вставлений рядок командою *Правка, Удалить*. Видалений запис неможливо відновити.

Перейдіть у режим таблиці і введіть у новостворену таблицю 3 записи.

У режимі *Конструктора* створіть форму, в якій задайте всі поля таблиці, а також створіть один обчислювальний елемент і три кнопки; для завершення роботи, для переходу до попереднього запису, для переходу до наступного запису.

Введіть 20 записів, використовуючи форму.

Перейдіть у режим таблиці.

Закрийте базу даних.

Лабораторна робота 5. ФІЛЬТРИ

Відкрийте базу даних із записами про кадровий склад підприємства. Перейдіть у режим таблиці (у вікні бази даних виберіть на вкладці **Таблицы** назву своєї таблиці і натисніть кнопку **Открыть**).

Розглянемо можливості фільтрування таблиці на прикладах. Найпростішими фільтрами є фільтр за виділенням та звичайний. Розширений фільтр використовують для складніших умов вибірки.

Перед створенням кожного фільтра внесіть, якщо потрібно, такі зміни у таблицю даних, щоб фільтр мав зміст (тобто щоб відфільтрована база була не порожньою).

Фільтр за виділенням

Виведіть на екран усі записи таблиці, що стосуються жінок. Для цього використайте найпростіший фільтр – за виділенням.

Це робиться так:

РОЗДІЛ 3. ФОРМАЛІЗАЦІЯ ІНФОРМАЦІЙНОГО ПРОЦЕСУ

При рішенні реальної задачі в загальному випадку неможливо одержати точне значення шуканого чисельного результату [12,15,21,35]. Існування неусувної похибки в математичній моделі об'єкту або процесу, що фігурує в задачі, похибки вхідних даних, багато з яких у реальних умовах отримані експериментально або шляхом вимірів, обчислювальна похибка і похибка методу, використовуваного для рішення, похибки, що виникають при яких-небудь додаткових впливах на об'єкт, які часто трактуються як збурення вхідних даних, приводять до необхідності їх сукупного врахування при оцінці результату. Після побудови математичної моделі процесу, яка необхідно задовольняє вимозі адекватності (рішення математичної задачі, отримане за її допомогою, незначно відрізняється від рішення реальної задачі), подана задача і її математична формалізація в процесі рішення й аналізу отриманого результату, як правило, не розрізняються. Однак, навіть зневажаючи неусувною похибкою й похибкою методу, завдяки особливостям машинної арифметики в загальному випадку неможливо одержати точне рішення математичної задачі [12].

Отримане наближене рішення деякої обчислювальної задачі A може розглядатися як точне рішення збуреної задачі \bar{A} , де \bar{A} відрізняється від A зміною вхідних даних [12]. Для визначення якості отриманого наближення необхідно мати можливість оцінити *ступінь залежності рішення від збурень вхідних даних*.

Назвемо задачу *чутливою* до збурних дій (похибок вхідних даних), якщо навіть малі збурні дії (малі похибки вхідних даних) можуть привести до значної похибки результату, і *нечутливою* інакше [12]. Для чутливої задачі похибка отриманого рішення буде набагато більшою за похибки вхідних даних, і ця властивість не залежить від методу рішення.

Міра чутливості задачі може бути різною. Значна чутливість до збурень позбавляє навіть потенційної можливості отримання результату її рішення з малою похибкою, тому надзвичайно важливим і актуальним є встановлення параметрів, що визначають чутливість задачі, і їх чисельна оцінка, а також отримання достатніх умов нечутливості.

Нехай ξ — вхідні дані для деякої задачі, результатом рішення якої є $\phi(\xi)$; $\bar{\xi}$ — збурені вхідні дані, а $\phi(\bar{\xi})$ — відповідне рішення. *Числом обумовленості* задачі в самому загальному виді називається величина [12]:

$$\overline{\lim}_{\xi \rightarrow \bar{\xi}} \frac{\text{расстояние между } \phi(\xi) \text{ и } \phi(\bar{\xi})}{\text{расстояние между } \xi \text{ и } \bar{\xi}}. \quad (3.1)$$

Відстані, що фігурують у формулі (3.1), визначаються введенням відповідних метрик у просторах вхідних даних і результатів.

За змістом співвідношення (3.1) представляє з себе аналог абсолютного значення швидкості зміни дійсної функції результату в точці ξ .

Очевидно, чим менше число обумовленості, тим менше збурення результату залежить від збурення вхідних даних, тим менша чутливість задачі. При малім числі обумовленості задача виявиться нечутливою до похибок вхідних даних. Таким чином, *число обумовленості (3.1) задачі є її мірою чутливості до збурних дій.*

3.1. Неперервний інформаційний процес

Будь-який інформаційний процес (визначення 1.1) визначається зміною параметрів, що його задають, або приведенням одних параметрів (*вихідних*) у відповідність з іншими (*вхідними*) за деяким законом. Таким чином, інформаційний процес можна розглядати як рішення задачі, що полягає в одержанні значень вихідних параметрів за значеннями вхідних, а інформаційну систему як результат рішення задачі про її формування.

Визначення 3.1. *Чутливістю* інформаційної системи (інформаційного процесу) у загальному випадку назвемо чутливість задачі її формування (задачі одержання значень вихідних параметрів).

При формальному представленні процесу у вигляді математичної моделі виділяється лише скінченна множина параметрів (вхідних і вихідних), які несуть у собі всю важливу інформацію про його основні закономірності. Інформаційний процес у самому загальному виді можна представити як вектор-функцію скінченного числа змінних [12]:

$$\Phi(x_1, \dots, x_n) = \begin{pmatrix} \varphi_1(x_1, \dots, x_n) \\ \varphi_2(x_1, \dots, x_n) \\ \vdots \\ \varphi_m(x_1, \dots, x_n) \end{pmatrix} = \begin{pmatrix} \Phi_1 \\ \Phi_2 \\ \vdots \\ \Phi_m \end{pmatrix}, \quad (3.2)$$

що є законом відповідності вихідних параметрів (значень функції) $(\Phi_1, \Phi_2, \dots, \Phi_m) \in R^m$ вхідним (аргументам) $(x_1, \dots, x_n) \in D(\Phi) \subseteq R^n$, де $D(\Phi)$ — область визначення $\Phi(x_1, \dots, x_n)$.

Співвідношення (3.2) — це загальний вид будь-якої математичної моделі довільного інформаційного процесу. Кожна конкретна модель, необхідно адекватна реальному процесу, задається лише певним видом функції $\Phi(x_1, \dots, x_n)$.

Вектор-функція, яка діє в простір R^m , породжує m дійсних функцій (визначення 1.11)

$$\varphi_i(x_1, \dots, x_n), \quad (3.3)$$

$$\varphi_i : D(\Phi) \rightarrow R, \quad D(\Phi) \subseteq R^n, \quad i = \overline{1, m},$$

на своїй області визначення $D(\Phi)$ [14]. Таким чином, має місце наступне

застосуйте команду **Формат, Шрифт**, яка працює так, як і у програмі Microsoft Word.

Перевірте можливості пошуку — спробуйте відшукати запис за значенням певного поля, наприклад, за заданим прізвищем. Для цього задайте команду **Правка, Нйти** і у діалоговому вікні **Поиск**. Відкриється наступне вікно (рис. 8.13):

у текстове поле **Образец** введіть декілька початкових літер прізвища;

у полі **Поиск в** виберіть **Основные дані: Таблица**;

розкрийте список **Совпадение** і виберіть **С любой частью**;

у текстове поле **Просмотр** введіть **Все**;

заберіть усі прапорці;

натискайте кнопку **Нйти далее**, доки потрібний запис не буде знайдено.

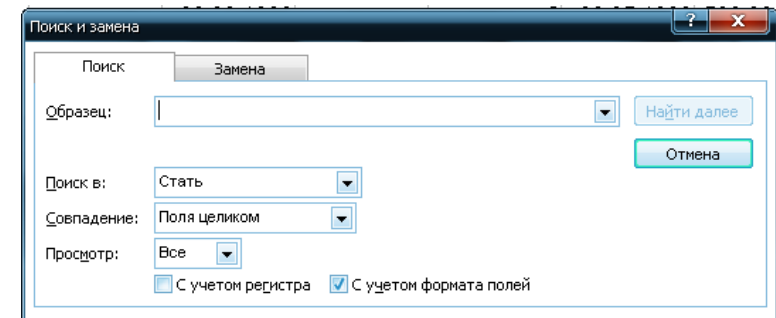


Рис. 8.13. Вигляд вікна для пошуку .

Перевірте можливості команди **Заменить** з меню **Правка**. Замініть, наприклад, назву якогось з підрозділів (згідно з припущення, що його перейменували). Для цього задайте цю команду і введіть відповідно стару (у поле **Образец**) і нову назви (в поле **Заменить на**), після чого натисніть кнопку **Заменить все**.

Перевірте можливість копіювання записів. Припустимо, потрібно створити ще один запис про особу, дані про яку дуже подібні до даних у наявному в таблиці записі про іншу особу. Тоді можна скопіювати цей запис, а потім лише внести у нього зміни. Необхідно виконати наступні дії:

вказанте такий запис, клацнувши лівою кнопкою миші на сірій смузі, у відповідному рядку, ліворуч від першого стовпця;

задайте команду **Правка, Копировать** — запис скопіюється у буфер обміну;

перейдіть у самий нижній із записів (порожній), тобто зробіть його активним;

задайте команду **Правка, Добавить из буфера**;

внесіть необхідні зміни у поля новоствореного запису.

Закрийте базу даних.

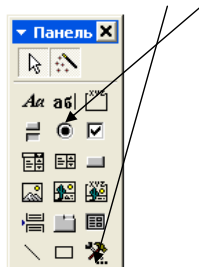


Рис. 8.12. Вікно Панелі елементів з натиснутою кнопкою *Мастер*.

20. Перейдіть у режим форми і введіть 20 записів.

21. Закінчіть роботу з формою натиском створеної кнопки. Збережіть форму.

Лабораторна робота 4. РОБОТА З ТАБЛИЦЯМИ ТА ФОРМАМИ

Відкриття бази даних та внесення змін у таблицю даних у режимі таблиці

Відкрийте базу даних, що містить дані про кадровий склад фірми. Для цього використайте або у початковому вікні команду *Открыть базу данных*, або меню *Файл*, команду *Открыть базу данных*, або кнопку панелі інструментів *Открыть базу данных*. Одночасно потрібно відкрити свій файл з розширенням *.mdb* і у головному вікні бази даних натиснути кнопку *Открыть* – на екрані з’явиться таблиця у режимі таблиці.

Збільшіть ширину тих стовпців, у яких не поміщаються назви полів. Це робиться так, як у програмі Microsoft Excel: розмістіть вказівник миші на межі між назвами стовпців, натисніть ліву кнопку миші і, не відпускаючи її, перемістіть мишу вправо.

Доповніть таблицю даних ще одним записом.

Впорядкуйте записи так, щоб прізвища розміщувалися за алфавітом. Для цього стовпчик *Прізвище* потрібно спочатку виділити (як у Microsoft Word). Щоб виділити стовпчик, розмістимо вказівник на рамці його назви так, щоб він набув вигляду вертикальної чорної стрілочки і клацнемо лівою кнопкою мишки. Після цього скористайтеся командою *Записи*, *Сортировка*, *Сортировка по возрастанию* або відповідною кнопкою на панелі інструментів.

Повторно впорядкуйте таблицю так, щоб табельні номери зростали.

Заховайте стовпчик з розмірами окладів. Для цього зазначте його і задайте команду *Формат*, *Скрыть*.

Поверніть стовпчик командою *Формат*, *Отобразить столбцы*.

Змініть шрифти у таблиці. Для цього спочатку зазначте дані, а потім

Твердження 3.1. Довільний неперервний інформаційний процес (інформаційна система, яка розглядається як результат процесу її формування) може бути формально представлений у вигляді скінченної множини дійсних функцій скінченного числа змінних (3.3), а аналіз цього процесу (системи) зведений до аналізу отриманих функцій.

Питання про зручність представлення інформаційного процесу (системи) у вигляді (3.3), безпосередніх методах реалізації, обчислювальних витратах і т.д. не обговорюються, важлива лише його принципова можливість.

Враховуючи важливість числа обумовленості будь-якої задачі для характеристики результату її рішення — чутливості до збурних дій, знайдемо безпосередні вирази для числа обумовленості задачі про обчислення довільної дійсної функції скінченного числа змінних, що відіграє, як впливає з твердження 3.1, визначну роль при формалізації інформаційного процесу (інформаційної системи).

Нехай $f(x_1, \dots, x_n)$ — дійсна функція дійсних змінних x_1, \dots, x_n , областю визначення якої є компактна множина $E \subseteq R^n$ (визначення 1.5, 1.6):

$$f: E \rightarrow R.$$

Необхідно обчислити $f(x_1, \dots, x_n)$. Для сукупності аргументів $(x_1, \dots, x_n) \in E$ є лише наближені значення $(x_1 + \delta x_1, \dots, x_n + \delta x_n)$ і межі для похибок $\delta x_1, \dots, \delta x_n$. Тоді [12]

$$f(x_1 + \delta x_1, \dots, x_n + \delta x_n) = f(x_1, \dots, x_n) + \frac{\partial f(x_1, \dots, x_n)}{\partial x_1} \delta x_1 + \dots + \frac{\partial f(x_1, \dots, x_n)}{\partial x_n} \delta x_n + o\left(\sqrt{\delta x_1^2 + \dots + \delta x_n^2}\right), \quad (3.4)$$

коли $\sqrt{\delta x_1^2 + \dots + \delta x_n^2} \rightarrow 0$,

де $o\left(\sqrt{\delta x_1^2 + \dots + \delta x_n^2}\right)$, коли $\sqrt{\delta x_1^2 + \dots + \delta x_n^2} \rightarrow 0$, — це така функція, що

$$\lim_{\sqrt{\delta x_1^2 + \dots + \delta x_n^2} \rightarrow 0} \frac{o\left(\sqrt{\delta x_1^2 + \dots + \delta x_n^2}\right)}{\sqrt{\delta x_1^2 + \dots + \delta x_n^2}} = 0.$$

Зі співвідношення (3.4) випливає, що

$$|f(x_1 + \delta x_1, \dots, x_n + \delta x_n) - f(x_1, \dots, x_n)| \approx \left| \frac{\partial f(x_1, \dots, x_n)}{\partial x_1} \delta x_1 + \dots + \frac{\partial f(x_1, \dots, x_n)}{\partial x_n} \delta x_n \right| = \|(grad f(x_1, \dots, x_n), (\delta x_1, \dots, \delta x_n))\|, \quad (3.5)$$

де $grad f(x_1, \dots, x_n)$ — вектор градієнта функції f в точці (x_1, \dots, x_n) (визначення 1.15);

(•,•) — скалярний добуток векторів-аргументів (визначення 1.16).

Використання в правій частині (3.5) нерівності Коші-Буняковського дає наступну оцінку [12]:

$$\|(grad f(x_1, \dots, x_n), (\delta x_1, \dots, \delta x_n))\| \leq \|grad f(x_1, \dots, x_n)\| \|(\delta x_1, \dots, \delta x_n)\|,$$

де $\|\bullet\|$ — векторна 2-норма (визначення 1.17), звідки

$$|f(x_1 + \delta x_1, \dots, x_n + \delta x_n) - f(x_1, \dots, x_n)| \approx \|grad f(x_1, \dots, x_n)\| \|(\delta x_1, \dots, \delta x_n)\|. \quad (3.6)$$

З (3.6) випливає, що $\|grad f(x_1, \dots, x_n)\|$ є абсолютним числом обумовленості, а тому мірою чутливості задачі обчислення $f(x_1, \dots, x_n)$ до збурень вхідних даних [12].

Будь-яке перетворення функції можна розглядати як її збурення в деяких точках області визначення E , чи в кожній точці цієї області. Реакція функції на збурення в конкретній точці $(x_1, \dots, x_n) \in E$ буде визначатися $\|grad f(x_1, \dots, x_n)\|$.

Теорема 3.1. Для того, щоб неперервний інформаційний процес, що формалізується у вигляді сукупності дійсних функцій (3.3), який розглядається як результат задачі визначення значень вихідних параметрів $(\Phi_1, \Phi_2, \dots, \Phi_m) \in R^m$ за значеннями вхідних $(x_1, \dots, x_n) \in D(\Phi)$, був нечутливим до збурних дій у точці (x_1, \dots, x_n) (на області визначення $D(\Phi)$) достатньо, щоб $\|grad \varphi_i(x_1, \dots, x_n)\|$, $i = \overline{1, m}$, мали малі значення в (x_1, \dots, x_n) (на всій області $D(\Phi)$).

Зуваження 3.1. На практиці вимогу малості значень $\|grad \varphi_i(x_1, \dots, x_n)\|$, $i = \overline{1, m}$, для нечутливості інформаційного процесу достатньо конкретизувати вимогою, щоб $\|grad \varphi_i(x_1, \dots, x_n)\|$, $i = \overline{1, m}$, були порядку одиниці.

Оскільки $\|grad \varphi_i(x_1, \dots, x_n)\|$ визначає значення максимальної швидкості зростання функції φ_i в точці (x_1, \dots, x_n) [14], то ідеальним з погляду забезпечення нечутливості неперервного інформаційного процесу (3.3) до збурних дій на $D(\Phi)$ очевидно буде тривіальний випадок, коли

$$\varphi_i(x_1, \dots, x_n) = c_i, \quad (x_1, \dots, x_n) \in D(\Phi),$$

зафіксуйте вказівник миші на кнопці **а6** на панелі елементів, а потім на вільному місці (там, де буде обчислювальний елемент) форми – на ній з’явиться нова пара з’єднаних прямокутників. Лівий призначено для назви, правий – для виразу, що обчислюватиметься; зазначте лівий прямокутник, видаліть розміщений там стандартний текст і введіть назву: “Відрахування на ремонт”; зазначте правий прямокутник, виберіть у меню **Вид** команду **Свойства**, на вкладці **Данные**, у рядку **Данные** клацніть по кнопці з трьома крапками – відкриється вікно побудовувача виразів (рис. 8.11);

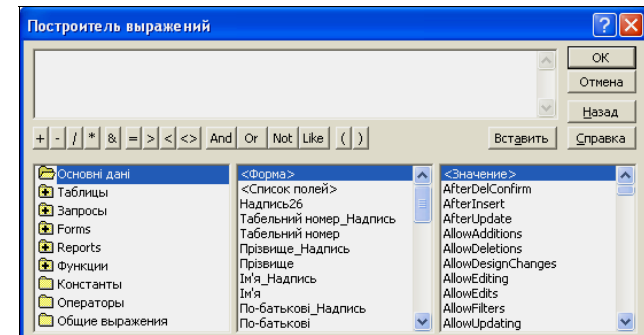


Рис. 8.11. Властивості вікна побудовувача виразів.

у вікні побудовувача зафіксуйте вказівник мишки у середній колонці на рядку **Список полей**, у сусідній справа колонці виберіть назву потрібного поля, що використовуватиметься в обчислювальному елементі керування (тобто **Посадовий оклад**). Далі слід натиснути кнопку **Вставить** і сформувати потрібний вираз (тобто = [Посадовий оклад]*0,02), використовуючи кнопки зі знаками операцій та клавіатуру. В кінці натисніть кнопку **Ок** і закрийте вікно команди **Свойства**.

Перевірте, як створено обчислювальний елемент керування. Для цього перейдіть у режим форми кнопкою **Вид**.

19. Поверніться до форми у режимі **Конструктора (Вид, Конструктор)**. Створіть кнопку для закриття форми і повернення у вікно бази даних. Це можна зробити так:

на панелі елементів виберіть кнопку з назвою **“Кнопка”** і розмістіть цей елемент на формі – відкриється вікно **Майстра** створення кнопок. (Якщо на панелі елементів не натиснуто кнопку **Мастер**, то це вікно не відкриється) (рис. 8.12);

у вікні **Майстра**, у списку **Категории** виберіть **Работа с формой**, а у списку **Действия – Закрытие формы** і натисніть кнопку **Далее**;

натисніть радіоклавішу **текст**, введіть слово **“Закінчення”** і натисніть кнопку **Готово**.

Введіть заголовок форми. Для цього:
 відтягніть мишкою вниз смужку **Область даних**, щоб вивільнити місце для заголовка;
 клацніть мишкою по кнопці **“Надпись”** панелі елементів керування і зафіксуйте вказівник миші на звільненому місці;
 введіть заголовок;
 клацніть мишкою ззовні тексту.

Змініть розміри шрифту для заголовка, а також стиль, кольори літер та фону. Для цього:
 зазначте заголовок, клацнувши по ньому (з’явиться рамка з квадратиками, як і завжди у зазначеного об’єкта);
 використайте необхідні кнопки та списки, що розкриваються на панелі інструментів **Формат** (якщо панель не встановлено, її можна вивести командою **Вид / Панелі інструментов / Настройка / Формат (форма / отчёт)**).

Збільшіть об’єкт, розтягнувши його, зачепившись за один із кутів (у той момент, коли курсор набуває вигляду двонапрямленої стрілочки).

Спробуйте переміщати об’єкт. Для цього:
 зазначте об’єкт, клацнувши на ньому;
 зафіксуйте вказівник миші над рамкою об’єкта, щоб вказівник набув вигляду розкритої долоні, натисніть ліву кнопку миші і, не відпускаючи її, перетягніть заголовок у потрібному напрямку.

В області даних розміщено назви полів та місця вводу даних у ці поля (елементи керування типу **“Поле”**), які розташовані у суміжних прямокутниках. Під час цього прямокутник з назвою поля розміщено ліворуч від прямокутника, призначеного для вводу даних у це поле. Деякі назви полів представлені на формі не повністю. Потрібно збільшити призначені їм ділянки на формі. Для цього потрібно знати про наступні можливості.

Якщо клацнути по одному з пов’язаних прямокутників, то на його рамці з’являться квадратики (маркери), тобто об’єкт буде зазначено. Якщо тепер розмістити вказівник миші над рамкою, то курсор миші набуває вигляду розкритої долоні. У цю мить можна перемістити об’єкт. Під час цього переміщуватися будуть обидва прямокутники.

Щоб від’єднати один об’єкт від другого, потрібно цей об’єкт виділити і розмістити вказівник миші на маркері у верхньому лівому куті так, щоб курсор набув вигляду руки, у якій випростано тільки вказівний палець, а решта зігнуто в кулак. Натиснувши у той момент кнопку миші, можна перемістити об’єкт, від’єднавши його від іншого.

Змініть висоту і ширину прямокутника, призначеного для поля **Прізвище**. Для цього зазначте його, виберіть команду **Свойства** з меню **Вид** і змініть ці параметри (на вкладці **Макет**).

Створіть на формі обчислювальний елемент керування для відрахування суми на ремонт приміщення, яка становить, наприклад, 2% від посадового окладу. Для цього виконайте наступні дії:

$$c_i = const, \quad i = \overline{1, m}.$$

Тоді

$$grad \varphi_i(x_1, \dots, x_n) = (0, 0, \dots, 0)^T,$$

$$(x_1, \dots, x_n) \in D(\Phi), \quad i = \overline{1, m}.$$

Припустимо, що $\varphi_i \in C^1(D(\Phi))$, $i = \overline{1, m}$, тобто мають всі частинні похідні, неперервні на $D(\Phi)$, і $D(\Phi)$ — компакт. Таке обмеження на область визначення функції Φ є абсолютно логічним: як обмеженість, так і замкненість, які притаманні $D(\Phi)$ завдяки компактності, очевидно відповідають реальному інформаційному процесу (інформаційній системі). Оскільки будь-яка неперервна на компактній множині дійсна функція, як впливає з теореми Вейерштрасса [12], буде обмеженою на цій множині (визначення 1.12), то

$$\forall i = \overline{1, m}, j = \overline{1, n} \quad \exists M_{ij} = const, M_{ij} \geq 0: \quad \left| \frac{\partial \varphi_i(x_1, \dots, x_n)}{\partial x_j} \right| \leq M_{ij}, \quad (x_1, \dots, x_n) \in D(\Phi).$$

Нехай

$$M = \max_{1 \leq i \leq m, 1 \leq j \leq n} M_{ij}. \quad (3.7)$$

Тоді

$$\left| \frac{\partial \varphi_i(x_1, \dots, x_n)}{\partial x_j} \right| \leq M, \quad (x_1, \dots, x_n) \in D(\Phi), \quad i = \overline{1, m}, j = \overline{1, n},$$

що необхідно приведе до обмеженості $\|grad \varphi_i(x_1, \dots, x_n)\|$, $i = \overline{1, m}$, на $D(\Phi)$ відповідно до співвідношення:

$$\|grad \varphi_i(x_1, \dots, x_n)\| \leq M \sqrt{n}, \quad i = \overline{1, m}.$$

Таким чином має місце наступне

Твердження 3.2. Нехай для неперервного інформаційного процесу (3.3) $\varphi_i \in C^1(D(\Phi))$, $i = \overline{1, m}$, $D(\Phi)$ — компакт. Тоді *достатньою умовою нечутливості* інформаційного процесу до збурних дій на всій області визначення $D(\Phi)$ буде мализна мажоруючої модулі частинних похідних першого порядку функцій φ_i сталої M , яка визначається відповідно до (3.7).

Зауваження 3.2. З врахуванням зауваження 3.1 на практиці вимогу малого значення сталої M для нечутливості інформаційного процесу достатньо конкретизувати вимогою:

$$M = O\left(n^{\frac{1}{2}}\right).$$

Визначення O -символів наведені в розділі 1.

3.2. Дискретний інформаційний процес

Повернемося до формального представлення інформаційного процесу у вигляді неперервної вектор-функції багатьох змінних (3.2).

При побудові функції (3.2) для інформаційного процесу (системи) отримання реальних значень вхідних параметрів, що є, як правило, результатами вимірів, експериментів і т.д., припускає дискретність цих значень. Процес обробки функції (3.2) з використанням обчислювальних засобів і чисельних методів так чи інакше приведе до її дискретизації, у результаті якої вийде n -вимірний простір з елементами з простору R^n . З врахуванням того, що кожна вектор-функція, яка діє в простір R^n , породжує m дійсних функцій (3.3) на своїй області визначення, результат дискретизації може бути представлений як множина, що складається з m n -вимірних матриць M_1, M_2, \dots, M_m з елементами з простору R , кожна з яких відповідає своїй певній функції (3.3).

Твердження 3.3. Довільний інформаційний процес (інформаційна система, що розглядається як результат процесу її синтезу) може бути формально представлений у вигляді скінченної множини матриць M_1, M_2, \dots, M_m скінченної вимірності з дійсними елементами, а аналіз інформаційного процесу принципово можна звести до аналізу відповідних матриць.

Питання про зручність матричного представлення, безпосередніх методах реалізації, обчислювальних витратах і т.д. не обговорюються, важлива лише його принципова можливість.

Зауваження 3.3. Вихідні параметри $\Phi_1, \Phi_2, \dots, \Phi_m$ можуть виявитися залежними, тоді одержувані m n -вимірних матриць M_1, M_2, \dots, M_m також будуть залежними. Однак на процесі аналізу стану інформаційної системи це не відіб'ється, оскільки цей процес не змінює наявних залежностей між вхідними й вихідними параметрами.

Ми повернемося до питання дослідження зв'язків між вихідними параметрами в розділі 6, присвяченому матричному аналізу структури інформаційного процесу.

Як показує практика, з урахуванням зручності обробки одержуваної моделі, найчастіше при моделюванні реальних процесів і систем використовуються двовимірні матриці, які завдяки наведеному нижче зауваженню й будуть розглядатися далі при описі інформаційних систем.

Зауваження 3.4. Якщо в отриманій при моделюванні інформаційного процесу (інформаційної системи) сукупності M_1, M_2, \dots, M_m вимірність матриць

Виберіть стиль форми (тобто фон, заставку) і натисніть кнопку *Далее*.
Надайте формі ім'я, натисніть кнопку *открытие формы для просмотра или ввода данных* і натисніть кнопку *Готово*, на екрані з'явиться форма у режимі форми (рис. 8.9). Перегляньте усі записи, використовуючи кнопки, розміщені у нижньому рядку форми. Натискаючи їх по черзі, визначте, для чого кожна з них призначено.

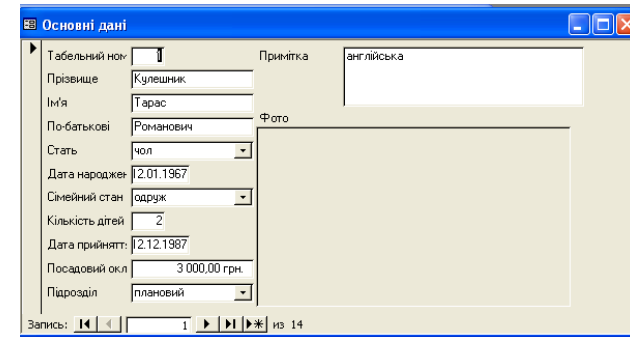


Рис. 8.9. Режим форми.

Вставте новий запис за допомогою команди *Новая запись* з меню *Вставка* або натисніть кнопку *Новая запись* на панелі інструментів.

Перейдіть у режим *Конструктора* форми (рис. 8.10). Для цього виберіть в меню *Вид* команду *Конструктор* або замість цього розкрийте список на кнопці *Вид* виберіть: *Конструктор*.

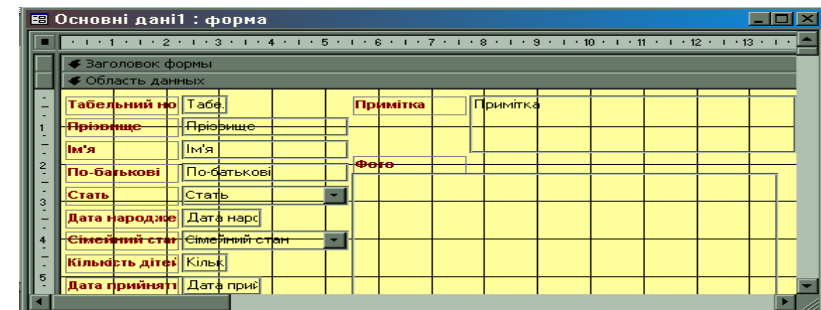


Рис. 8.10. Форма в режимі *Конструктора*.

Розгорніть вікно *Конструктора* на весь екран і перемістіть панель елементів керування так, щоб вона не закривала форму. За бажанням сітку можна зняти командою *Сетка* з меню *Вид*.

Встановіть вказівник миші на верхній край рядка *Примечание формы* і коли курсор набуде вигляду двонапрямленої стрілки, натисніть ліву кнопку миші і перетягніть цей рядок нижче.

Встановіть прапорець *вывести справку по работе с таблицей* і натисніть кнопку *Готово*.

Введіть у новостворену таблицю 25 записів.

Збережіть базу даних.

Лабораторна робота 3. РОБОТА З ФОРМАМИ

Відкрийте створену раніше базу даних, що містить дані про кадровий склад підприємства. Для цього використайте або у початковому вікні команду *Открыть базу данных*, або меню *Файл*, команду *Открыть базу данных*, або кнопку панелі інструментів *Открыть базу данных*. Під час цього потрібно відкрити свій файл з розширенням *.mdb*.

У головному вікні бази даних виберіть вкладку *Формы, Создание формы с помощью мастера*. Відкриється вікно *Создание формы* (рис. 8.8).

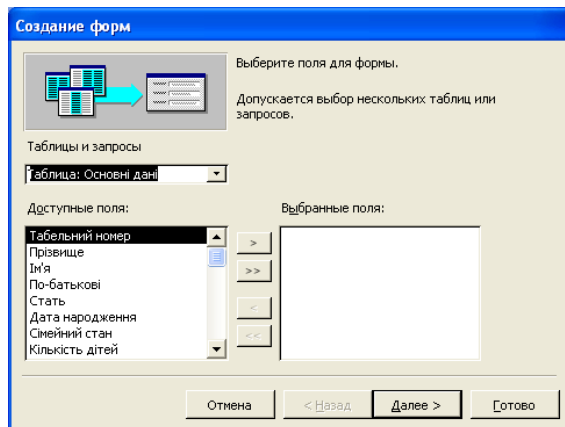


Рис. 8.8. Вікно для створення форми.

розкрийте список біля слів: *“Таблицы и запросы”* і виберіть назву вашої таблиці, що містить дані про кадровий склад підприємства *Основные дані*;

Далі необхідно вибрати поля, які розмістяться на формі. Виберіть всі поля, натиснувши кнопку *>>*, або лише деякі, використавши кнопку *>*, а потім кнопку *Далее*.

Виберіть зовнішній вигляд форми, натиснувши на першу з запропонованих кнопок:

- в один столбец;*
- ленточный;*
- табличный;*
- выровненный.*

Натисніть кнопку *Далее*.

$n > 2$, то кожній $M_j, j = \overline{1, m}$, можна поставити в співвідношення скінченну множину матриць вимірності 2, кожна з яких виходить з M_j шляхом фіксування в ній всіх індексів, крім двох.

Твердження 3.4. Будь-який інформаційний процес (інформаційна система) може бути формально представлений у вигляді скінченної множини двовимірних матриць з дійсними елементами, а *аналіз інформаційного процесу принципово може бути зведений до аналізу двовимірних матриць*.

Для простоти викладу, не обмежуючи при цьому спільності міркувань, відповідно до твердження 3.4, як математичну модель інформаційної системи далі будемо розглядати двовимірну (прямокутну або квадратну) матрицю F .

3.3. Матричне представлення перетворення інформаційної системи

Повні набори параметрів

Як впливає з твердження 3.4, результат будь-яких дій над інформаційною системою у загальному випадку формально можна представити як збурення ΔF матриці F , яка відповідає системі. При цьому самі дії представляються як збурні для F , а задача будь-якого перетворення системи, тобто генерації нової, для якої стара система є вхідними даними, формалізується у вигляді задачі отримання збуреної матриці \bar{F} для поданої матриці F , до того ж результуюча матриця очевидно задовольняє співвідношенню:

$$\bar{F} = F + \Delta F, \quad (3.8)$$

де $\Delta F = f(F)$, тобто $\Delta F \in$ деякою функцією матриці F [12].

Зі співвідношення (3.8) випливає істинність наступного твердження.

Твердження 3.5. Будь-які перетворення інформаційної системи формально можуть бути представленими у вигляді елементарних матричних операцій.

З твердження 3.4 випливає, що як набір параметрів, які однозначно визначають й всебічно характеризують інформаційну систему, можна використовувати будь-який з наборів, що однозначно визначає довільну двовимірну матрицю [12]. Назвемо такі набори параметрів *повними* й визначимо їх можливе наповнення.

Нехай F — $m \times n$ -матриця з елементами $f_{ij}, i = \overline{1, m}, j = \overline{1, n}, (m \geq n)$. Для неї існує розкладання, яке називається *сингулярним* [15] (SVD):

$$F = U \Sigma V^T, \quad (3.9)$$

де U, V — матриці розміру $m \times m$ і $n \times n$ відповідно;

$$\Sigma = \text{diag}(\sigma_1, \dots, \sigma_n), \quad \sigma_1 \geq \dots \geq \sigma_n \geq 0,$$

при цьому $U, V \in$ ортогональними, тобто задовольняють співвідношенням:

$$U^T U = I, V^T V = I,$$

де I — одинична матриця відповідного розміру.

Стовпці u_1, \dots, u_n матриці U і v_1, \dots, v_n матриці V називають відповідно лівими і правими сингулярними векторами (СНВ) матриці F , величини $\sigma_1, \dots, \sigma_n$ — сингулярними числами (СНЧ), а (σ_i, u_i, v_i) сингулярними трійками F . Якщо $m < n$, розглядається SVD матриці F^T .

Якщо F — симетрична $n \times n$ -матриця з власними значеннями (ВЗ) $\lambda_i \in R, i = \overline{1, n}$, і ортонормованими власними векторами (ВВ) $u_i, i = \overline{1, n}$ (визначення 1.7, 1.8), то для неї існує спектральне розкладання (СР) [36]:

$$F = U \Lambda U^T, \quad (3.10)$$

де $\Lambda = \text{diag}(\lambda_1, \dots, \lambda_n)$ — матриця ВЗ;

$U = [u_1, \dots, u_n]$ — ортогональна матриця ВВ:

$$U^T U = I.$$

Розкладання (3.10) може бути представленим у формі зовнішніх добутків:

$$F = \sum_{i=1}^n \lambda_i u_i u_i^T. \quad (3.11)$$

Для симетричної матриці F її спектр — множина всіх ВЗ, завжди дійсний. ВЗ, будучи розв'язками характеристичного рівняння $\det(F - \lambda E) = 0$, визначаються однозначно, на відміну від ВВ.

Зауваження 3.5. Якщо матриця F має загальну структуру, її ВЗ λ відповідають ліві й праві ВВ. Так ненульовий вектор u , який задовольняє умові

$$F u = \lambda u,$$

називається правим ВВ, а ненульовий вектор w , такий, що

$$w^T F = \lambda w^T,$$

називається лівим ВВ. У випадку $F = F^T$ ліві й праві ВВ співпадають.

У загальному випадку сингулярне (спектральне) розкладання матриці визначається неоднозначно. Відповідно до [12], назвемо вектор u

Після ознайомлення з новою базою даних вилучіть цей файл, оскільки він займатиме багато місця на твердому диску.

Створення власної таблиці у режимі Майстра

Задайте команду **Файл / Создать**.

У вікні **Создание**, на вкладці **Общие** виберіть **База данных** та натисніть кнопку **Ок**.

У вікні, що відкриється, введіть ім'я бази даних і натисніть **Создать**. Відкриється наступне вікно (рис. 8.6).

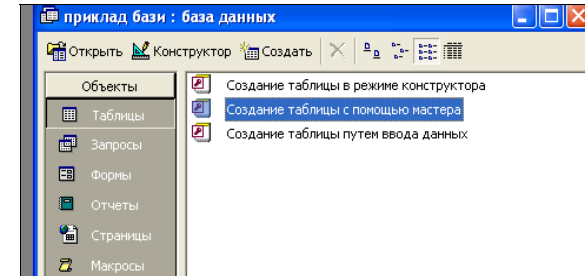


Рис. 8.6. Початок створення таблиці за допомогою Майстра.

Серед об'єктів виберіть **Таблицы, Создание таблицы с помощью мастера**.

Відкриється наступне вікно (рис. 8.7).

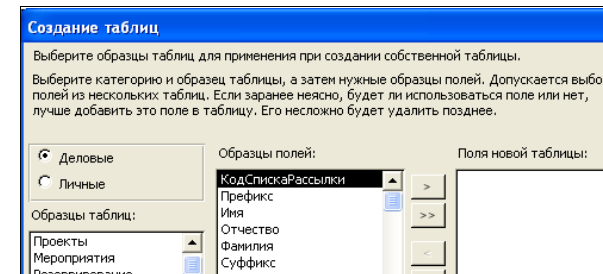


Рис. 8.7. Формування полів за допомогою Майстра.

У списку **Образцы таблиц** виберіть: **Список рассылки**.

У списку **Образцы полей** виберіть на власний розсуд 6 назв полів, використовуючи кнопки форми «>» та «<<».

Переіменуйте кожне поле — надайте полям назви українською мовою. Для цього по черзі зазначайте поля і натискайте кнопку **Переименовать**.

Натисніть кнопку **Далее**.

Введіть назву таблиці і натисніть **Далее**.

Включіть кнопку **Ввод данных в таблицу с помощью формы, создаваемой мастером**.

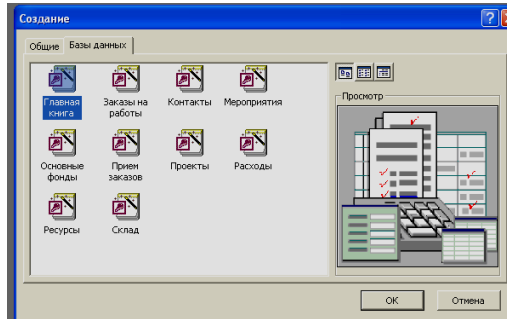


Рис. 8.4. Початок створення бази даних у режимі *Майстра*.

Серед запропонованих стандартних баз даних виберіть, наприклад, *Главная книга* і натисніть **Ок**.
 У вікні, що відкриється, введіть назву файлу, відмінну від вже існуючих. Спостерігайте за роботою *Майстра* по автоматичному створенню бази і натискайте у діалогових вікнах, що виникатимуть, кнопку *Далее*. У відповідь на запитання *Майстра* про заголовок бази даних введіть своє прізвище та, наприклад, слово: "Куляр". На завершення натисніть кнопку *Готово* і відкрийте створену базу даних, якщо вона не відкриється автоматично.
 Отримаємо вікно, що має наступний вигляд (рис. 8.5).

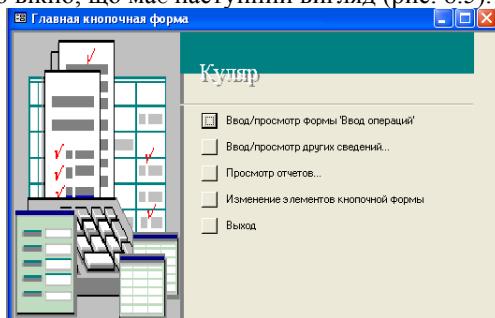


Рис. 8.5. Завершення створення бази даних за допомогою *Майстра*.

Розгляньте головну кнопкову форму.
 Натисніть кнопку **Ввод/просмотр** для вводу даних.
 Введіть 10 записів. Зауважте, що потрібно вводити дані на 2-х сторінках. Спочатку бачимо першу. Після вводу першого запису потрібно клацнути по кнопці внизу екрана, на якій зображено обернену вправо стрілочку і зірочку. З'явиться порожній запис.
 Після введення даних перегляньте можливі форми звітів.
 Натисніть кнопку **Выход** для завершення роботи з базою даних.
 Вилучіть файл щойно створеної бази даних і створіть аналогічним чином базу даних *Склад* (відповідний значок виберіть на вкладці *Базы данных* діалогового вікна команди **Файл, Создать**.

лексикографічно додатним, якщо його перша ненульова компонента є додатною, а сингулярне (3.9) (спектральне (3.10)) розкладання *нормальним*, якщо стовпці матриці U лексикографічно додатні.

Теорема 3.2. Невироджена матриця F має єдине нормальне сингулярне розкладання, якщо її СНЧ парами відмінні:

$$\sigma_1 > \dots > \sigma_n > 0.$$

Доведення. Розглянемо матрицю FF^T . Ця матриця має властивості:

а) симетричності:

$$(FF^T)^T = FF^T;$$

б) додатної визначеності (визначення 1.35):

для $\forall x \in R^n, x \neq 0$, завдяки невірності F виконується співвідношення:

$$x^T(FF^T)x = (x^T F)(x^T F)^T > 0,$$

а $x^T(FF^T)x = 0$ тоді й тільки тоді, коли $x = 0$.

Скориставшись SVD (3.9) для матриці F , отримаємо співвідношення

$$FF^T = (U\Sigma V^T)(U\Sigma V^T)^T = U\Sigma^2 U^T, \quad (3.12)$$

яке визначає СВ матриці FF^T . З (3.12) ВЗ FF^T дорівнюють $\sigma_i^2, i = \overline{1, n}$, і оскільки всі $\sigma_i \geq 0$ різні, то ВЗ матриці FF^T також парами відмінні, крім того, ліві СНВ $F \in \text{ВВ } FF^T$.

Таким чином, для симетричної матриці FF^T існує власний базис, кожний її власний підпростір є одновимірним [12]. Оскільки ВВ FF^T нормовані, для кожного з них існує два можливі варіанти, що відрізняються тільки знаком, причому лексикографічно додатним буде тільки один можливий вектор. Це означає, що ВВ матриці FF^T (стовпці U), а тому ліві СНВ матриці F визначаються однозначно. Порядок СНЧ в матриці Σ ($\sigma_1 > \dots > \sigma_n > 0$) визначить порядок стовпців в U . Таким чином, U в цілому визначається однозначно.

Оскільки всі СНЧ F ненульові, $\det \Sigma \neq 0$, тоді з (3.9) випливає, що $V = F^T U \Sigma^{-1}$, тобто V також визначається однозначно, що в підсумку приводить до одиничності нормального SVD.

Далі будемо вважати, що всі матриці, які розглядаються, задовольняють умові теореми 3.2. Таким чином, СНЧ і СНВ, що отримуються за допомогою нормального сингулярного розкладання, однозначно визначають матрицю, а тому можуть розглядатися як *повний набір параметрів* для відповідної інформаційної системи.

Будь-яке перетворення інформаційної системи збурить її матрицю F і, як наслідок, збурить відповідні СНЧ і СНВ.

Твердження 3.6. Будь-яке перетворення інформаційної системи формально представляється в вигляді сукупності збурень СНЧ і (чи) СНВ її матриці, що дозволяє звести задачу аналізу процесу перетворення й підсумкового стану системи до аналізу збурень СНЧ і СНВ, а задачу синтезу інформаційної системи із заданими властивостями формалізувати у вигляді задачі забезпечення певних характеристик збурень СНЧ і СНВ її матриці.

Таким чином, про результат перетворення системи, її властивості, у тому числі й про чутливість, можна судити по характерних рисах сукупності збурень параметрів, що однозначно визначають систему, — СНЧ і СНВ, отриманих нормальним сингулярним розкладанням відповідної матриці.

Для СНЧ $\sigma_j(F)$, $\sigma_j(F + \Delta F)$, $j = \overline{1, n}$, матриць F і $F + \Delta F$ відповідно має місце співвідношення [12]:

$$\max_{1 \leq j \leq n} |\sigma_j(F) - \sigma_j(F + \Delta F)| \leq \|\Delta F\|_2, \quad (3.13)$$

де $\|\bullet\|_2$ — спектральна матрична норма (визначення 1.18, 1.19).

З (3.13) витікає, що збурення СНЧ порівнянні зі збуренням даних — ΔF , тобто СНЧ матриці є нечутливими до збурних дій, незалежно від того, чутливою чи нечутливою виявиться розглянута задача по формуванню $F + \Delta F$.

Зауваження 3.6. Для оцінки чутливості задачі перетворення інформаційної системи з матрицею F має сенс аналізувати лише збурення СНВ F , які відбулися під час перетворення, а результат перетворення системи для встановлення міри її чутливості до збурних дій, відповідно до твердження 3.6, розглядати у вигляді сукупності збурень СНВ її матриці.

Твердження 3.7. Чутливість задачі, що полягає в одержанні результату довільного перетворення інформаційної системи, формальним представленням якої є двовимірна матриця, буде визначатися чутливістю збурених під час перетворення системи СНВ матриці.

Розглянемо другий з можливих повних наборів параметрів.

Теорема 3.3. Нехай F — невідроджена симетрична $n \times n$ -матриця, модулі ВЗ якої парами відмінні

$$|\lambda_1| > |\lambda_2| > \dots > |\lambda_n|.$$

Для неї існує єдине нормальне СР.

Доведення. Оскільки кожне ВЗ F має кратність, рівну одиниці, то вимірність кожного власного підпростору матриці F також буде одиничною. Тоді для будь-якого λ_i , $i = \overline{1, n}$, нормований базис такого підпростору може визначатися двома способами: це вектори одиничної довжини протилежних напрямків. Один з них є лексикографічно додатним. Таким чином, стовпець матриці U , який відповідає ВЗ λ_i , визначиться однозначно, крім того, всі

Введіть 20 записів у таблицю. Вони повинні бути записані українською мовою, правильно, без скорочень. Не забувайте прізвище, ім'я та по батькові починати великою літерою.

Зауваження 1. У поле МЕМО вводять запис довільної довжини.

Зауваження 2. Оскільки на диску немає файлів з фотографіями, введіть (але тільки у один із записів) у поле OLE довільний малюнок. Для цього виконайте дії: у меню **Вставка** виберіть команду **Об'єкт**, а далі — **Создать из файла**; натисніть кнопку **Обзор**, виберіть на якомусь диску довільний файл з розширенням .bmp. Це малюнок, що замінить фото, натисніть кнопку **Открыть**, **Ок**.

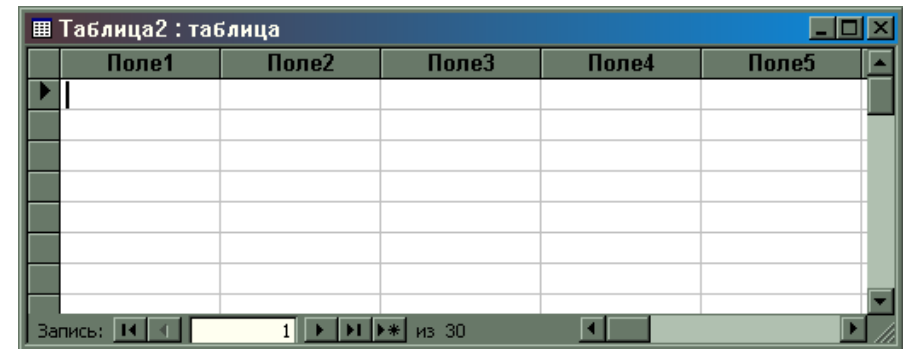


Рис. 8.3. Режим таблиці.

Закрийте вікно таблиці.

Закрийте вікно програми.

Зробіть копію вашої папки, що містить створену базу даних, скориставшись програмою ПРОВОДНИК. Для цього спочатку створіть нову папку на диску D. Бажано назвати її так, як першу, додавши слово: “КОПІЯ”. Після цього вміст першої папки скопіюйте у другу.

Лабораторна робота 2. ЗАСТОСУВАННЯ МАЙСТРА ДЛЯ СТВОРЕННЯ БАЗИ ДАНИХ

Запуск Майстра з початкового вікна програми

Запустіть на виконання програму Microsoft Access 2003.

Виберіть у початковому вікні програми піктограму **Создать, Базы данных**.

Отримаємо вікно, що має наступний вигляд (рис. 8.4).

Аналогічно використайте *Майстер* підстановок для створення списку фіксованих значень поля *Сімейний стан*. Введіть значення: “одруж.”, “неодруж.” (лапки не вводити).

Використайте *Майстер* підстановок для створення списку фіксованих значень поля *Підрозділ*.

Общие	Подстановка
Размер поля	Байт
Формат поля	
Число десятичных знаков	Авто
Маска ввода	
Подпись	
Значение по умолчанию	
Условие на значение	>0
Сообщение об ошибке	
Обязательное поле	Да
Индексированное поле	Да (Совпадения не допускаются)

Рис. 8.2. Нижня частина таблиці в режимі Конструктора під час задання характеристик поля *Табельний номер*.

Задайте властивості полів, використовуючи нижню частину таблиці опису полів (рис. 8.2):

змінить розміри наступних полів (властивість *Размер поля*):

Прізвище (встановити 30);

Ім'я (встановити 30);

По-батькові (встановити 30);

Табельний номер (встановити “байт”, вважаючи, що на підприємстві працює до 100 чоловік);

задайте властивість **Формат** для полів

Дата народження (встановити *Краткий формат даты*; після цього дати потрібно буде вводити у формі: дд.мм.рр)

Дата прийняття на роботу (встановити *Краткий формат даты*);

задайте наступні властивості для поля **Табельний номер**:

Значение по умолчанию – вилучити 0, залишити рядок порожнім

Обязательное поле – Да;

задайте властивість **Условие на значение** для поля **Дата народження**, а саме: >#1.1.1930#.

Зробіть поле **Табельний номер** полем первинного ключа, клацнувши по відповідній кнопці панелі інструментів.

Після вводу всіх полів та їхніх характеристик структуру таблиці створено. Для збереження таблиці використайте команду *Сохранить* з меню *Файл*, вилучіть стандартну назву “Таблица 1” і введіть назву: “Основні дані”.

Перейдіть у режим таблиці для вводу даних (рис. 8.3). Для цього або у меню *Вид* виберіть *Режим таблицы*, або клацніть по стрілочці на кнопці *Вид* і виберіть команду *Режим таблицы* зі списку, що розкриється.

стовпці U парами ортогональні. Порядок стовпців однозначно відповідає порядку елементів діагоналі Λ .

Далі будемо вважати, що всі симетричні матриці, які розглядаються, задовольняють умові теореми 3.3. Таким чином, $V3$ і $V6$, що отримуються за допомогою нормального СР, однозначно визначають матрицю, а тому також можуть розглядатися як *повний набір параметрів* для відповідної інформаційної системи.

Для інформаційної системи, моделлю якої є симетрична матриця, має місце твердження, яке аналогічне твердженню 3.6.

Твердження 3.8. Будь-яке перетворення інформаційної системи у випадку симетричності відповідної їй матриці формально представляється у вигляді збурень спектра і (або) $V6$ матриці, які однозначно визначаються нормальним спектральним розкладанням, що дозволяє звести задачу аналізу процесу перетворення й підсумкового стану інформаційної системи до аналізу збурень $V3$ і $V6$, а задачу синтезу системи із заданими властивостями — до забезпечення певних характеристик збурень $V3$ і $V6$ відповідної матриці.

Для аналізу стану інформаційної системи з симетричною матрицею завдяки твердженню 3.8 визначальну роль грає оцінка чутливості $V3$ і $V6$ матриці до збурних дій. Поведінка $V3$ аналізується в наступних двох теоремах, основою яких є наведена нижче лема [12].

Лема 3.1. Нехай λ — просте $V3$ $n \times n$ -матриці F загального виду, а u і w — відповідні лівий і правий нормовані $V6$, $\lambda + \Delta\lambda$ — відповідне $V3$ матриці $F + \Delta F$. Тоді

$$\Delta\lambda = \frac{w^T \Delta F u}{w^T u} + \mathcal{O}(\|\Delta F\|^2), \quad (3.14)$$

$$|\Delta\lambda| \leq \frac{\|\Delta F\|}{|w^T u|} + \mathcal{O}(\|\Delta F\|^2).$$

Теорема 3.4. Нехай λ — просте $V3$ симетричної матриці F , а $\lambda + \Delta\lambda$ — відповідне $V3$ матриці $F + \Delta F$. Тоді

$$|\Delta\lambda| \leq \|\Delta F\| + \mathcal{O}(\|\Delta F\|^2). \quad (3.15)$$

Доведення. Враховуючи зауваження 3.5, отримуємо, що $w^T u = u^T u = 1$. Тоді з (3.14) випливає (3.15).

Результат (3.15) можна поліпшити [12]:

Теорема 3.5. Для $V3$ симетричної матриці F має місце оцінка:

$$\max_{1 \leq j \leq n} |\lambda_j(F) - \lambda_j(F + \Delta F)| \leq \|\Delta F\|_2. \quad (3.16)$$

Завдяки (3.16) збурення $V3$, як і СНЧ відповідно до (3.13), порівнянні зі збуренням даних — ΔF , $V3$ симетричної матриці є нечутливими до збурних

дій чи добре обумовленими (чого не можна стверджувати в загальному випадку для несиметричних матриць [12]), незалежно від того, чутливою чи нечутливою виявиться розглянута задача по формуванню $F + \Delta F$.

Твердження 3.9. Чутливість задачі, що полягає в одержанні результату довільного перетворення інформаційної системи, математичною формалізацією якої є симетрична матриця, буде визначатися чутливістю збурених під час перетворення системи ВВ відповідної матриці.

Зауваження 3.7. Для оцінки чутливості задачі перетворення інформаційної системи з симетричною матрицею F має сенс аналізувати лише збурення ВВ F , які відбулися під час перетворення. Відповідно до твердження 3.8, результат перетворення інформаційної системи для встановлення міри її чутливості до збурених дій будемо розглядати в виді сукупності збурень її ВВ.

З врахуванням зауважень 3.6, 3.7 для $n=3$ геометрична ілюстрація довільного перетворення інформаційної системи представлена на рис.3.1, де u_1, u_2, u_3 — сингулярні (власні) вектори матриці поданої системи, а $\bar{u}_1, \bar{u}_2, \bar{u}_3$ — сингулярні (власні) вектори матриці збуреної системи.

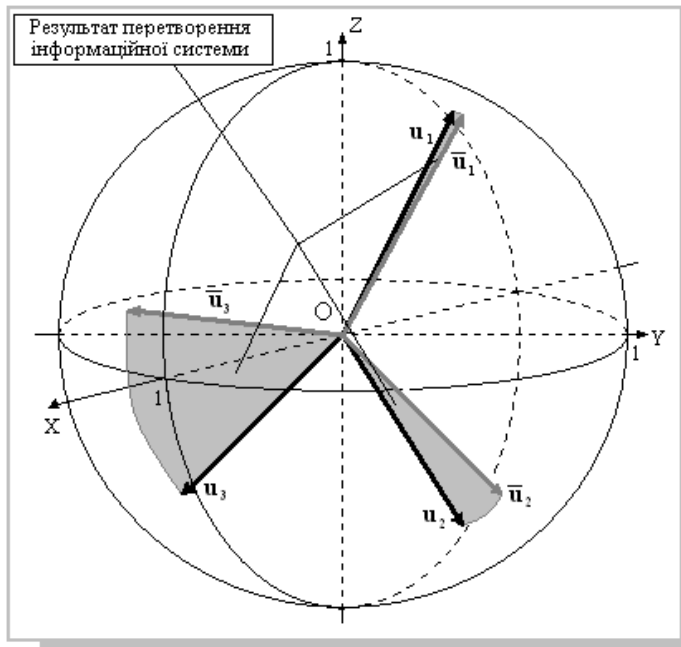


Рис.3.1. Геометричне представлення довільного перетворення інформаційної системи.

8.2. Лабораторний практикум з СУБД MICROSOFT ACCESS

Лабораторна робота 1. СТВОРЕННЯ ТАБЛИЦІ У РЕЖИМІ КОНСТРУКТОРА

Почніть створювати нову базу даних, скориставшись або початковим вікном, яке виникає на екрані після завантаження програми, (**Новая база данных, Ок**); або меню **Файл**, командою **Создать базу данных** (далі вкладка **Общие, Новая база данных, Ок**); або кнопкою на панелі інструментів **Создать базу данных** (далі вкладка **Общие, Новая база данных, Ок**).

У вікні **Файл новой базы данных** потрібно:

створити папку, що міститиме базу даних про кадровий склад підприємства, використавши відповідний значок панелі інструментів діалогового вікна;

відкрити створену папку;

ввести ім'я файла, що матиме розширення **.mdb**;

натиснути кнопку **Создать**.

З'явиться головне вікно бази даних. У ньому на вкладці **Таблицы** слід натиснути кнопку **Создать**.

У діалоговому вікні **Новая таблица** потрібно вибрати **Конструктор** (тобто створення таблиці в режимі Конструктора) і натиснути кнопку **Ок** – з'явиться вікно таблиці в режимі Конструктора.

У вікні таблиці в режимі Конструктора необхідно зробити опис структури першої таблиці щойно створеної бази даних. Для цього виконайте наступні дії.

Введіть такі ж поля та типи, як на рис. 8.1.

Розмістіть курсор у рядку поля **Стать**, у стовпці **Тип данных** і виберіть зі списку, що розкривається, **Мастер подстановок**. Після цього у діалогових вікнах, що виникатимуть, виберіть: **Будет введен фиксированный набор значений, Далее**; **Число столбцов – 1**, введіть у першу клітинку "жін", у наступну "чол" (лапки не вводити), **Готово**;

Основні дані : таблиця	
Имя поля	Тип данных
Табельний номер	Числовой
Прізвище	Текстовый
Ім'я	Текстовый
По-батькові	Текстовый
Стать	Текстовый
Дата народження	Дата/время
Сімейний стан	Текстовый
Кількість дітей	Числовой
Дата прийняття на роботу	Дата/время
Посадовий оклад	Денежный
Підрозділ	Текстовый
Примітка	Поле MEMO
Фото	Поле объекта OLE

Рис. 8.1. Верхня частина вікна таблиці у режимі **Конструктора**.

10% від собівартості, якщо собівартість <1100 грн;
15% від собівартості, якщо собівартість >=1100 грн та <=2200 грн;
19% від собівартості, якщо собівартість >2200 грн.

15. Обчислити прибутки фірми-посередника від продажу одиниці товару, якщо продається довільна кількість товарів, і оплата за посередництво становить:

7% від ціни, якщо ціна <800 грн;
11% від ціни, якщо ціна >=800 грн та <=2000 грн;
13% від ціни, якщо ціна >2000 грн.

РОЗДІЛ 4. VISUAL BASIC FOR APPLICATION

Незважаючи на досить широкі можливості системи MS Excel з створення нескладних баз даних у ряді випадків виникає ситуація, коли користувачеві не вистачає якоїсь функції для повного задоволення його потреб, а використати більш потужні СУБД немає необхідності. У таких випадках може допомогти розширення можливостей системи MS Excel шляхом написання власних функцій користувача, які дозволяють виконувати нестандартні, але необхідні користувачеві дії з оброблення даних.

Такі засоби надає вмонтований у систему MS Excel інтерпретатор мови високого рівня Visual Basic for Application (скорочено VBA). Є ряд причин, які обумовили таку популярність VBA, зокрема:

- VBA є єдиним середовищем розробки додатків у всіх програмах Microsoft Office. Подібний вибір обумовлено винятковими властивостями VBA, що бере свій початок ще в 1960-х роках і безупинно розвивається з тих пір. Однією з таких властивостей є унікальне поєднання простоти базових конструкцій і широких можливостей з реалізування алгоритмів оброблення даних довільного ступеня складності. У цьому сенсі VBA поступається, хіба що, Фортрану;
- за допомогою VBA можна записувати послідовності повторюваних команд, характерних для довільного додатка, і призначати правила (умови) відповідно до яких ці послідовності (процедури VBA) будуть викликатися в додатку. Наприклад, після натискання деякої кнопки робочий аркуш MS Excel може бути відформатований з урахуванням вимог користувача, записаних на VBA і збережених у робочій книзі або шаблоні;
- використавши засоби VBA можна забезпечити нестандартний діалог з користувачем, створюючи діалогові форми і обробляти реакцію користувача на подію в додатку;
- використання єдиної мови програмування сприяє більш тісній взаємодії Excel із продуктами Microsoft Office. За допомогою VBA можна розробляти додатки, що одночасно застосовують компоненти декількох програм.

Зрозуміло, що в межах даного навчального видання неможливо описати всі засоби VBA. Тому, для більш глибокого ознайомлення з можливостями VBA варто звернутися до спеціальної літератури.

4.1. Загальний опис VBA

Visual Basic for Applications – це об'єктно-орієнтована мова програмування високого рівня, вбудована в усі програми MS Office. Включення в зазначений пакет сучасної повнофункціональної мови програмування, інструментів для написання програмних кодів, їх редагування і відлагодження, а також великих бібліотек стандартних процедур дозволяє створювати прог-

рамні продукти, здатні з достатньою повнотою автоматизувати вирішення конкретних завдань користувача.

VBA є об'єктно-орієнтованим середовищем, що містить великий набір об'єктів, кожен з яких має безліч властивостей і методів. Сам об'єкт і всі інструменти, необхідні для його створення, відтворення або знищення, відносяться до певного класу. Ім'я класу вказує на тип об'єктів, які він містить.

Класи можна розглядати як основу для створення інших об'єктів цього ж типу. Тому всі об'єкти одного класу будуть діяти однаково.

Властивості й методи є членами класу. За допомогою властивостей описується як виглядає об'єкт, зокрема, дається інформація про прийоми форматування тексту, колір і розмір шрифту. Методи є процедурами, або, інакше кажучи, множиною операторів, що здійснюють певні завдання, які можуть бути виконані для об'єкта.

При об'єктно-орієнтованому програмуванні практично всі компоненти середовища розроблення є **об'єктами**. У MS Excel VBA-об'єктом вважається довільний елемент додатку – комірка, аркуш, робоча книга, діаграма. Фактично об'єктом є, властиво, і сам додаток MS Excel.

Об'єкти можуть включати області комірок, рамки комірок, вікна, сценарії, стилі і цей перелік можна продовжити. Кожен клас об'єктів має свою множину властивостей, функцій і подій.

Властивості об'єктів. Властивість є атрибутом об'єкта, що описує, як об'єкт виглядає (його колір, розмір і місце розташування) і як він діє (чи є видимим, чи посилається на інший об'єкт). Коли створюється об'єкт, MS Excel виконує процедуру створення цього об'єкта. Дана процедура, збережена в класі об'єктів, привласнює значення всім властивостям, що дозволяє відразу ж працювати з об'єктом.

Якщо користувач захоче змінити кілька властивостей, тоді він, як правило, виявить, що здебільшого вони задані саме так, як йому потрібно, тому вносити зміни звичайно не доводиться. Для того, щоб у макросі VBA все-таки надати властивості нове значення необхідно лише розробити оператор присвоєння, у якому ліворуч від символу присвоєння будуть зазначені ім'я й властивість об'єкта (розділені крапкою), а праворуч – нове значення.

Властивість аркуша Name відображається на його ярличку. Причому дана властивість входить також у безліч інших класів.

Метод – це дія, що може бути виконана над об'єктом. Методи реалізуються за допомогою виконання процедури, що є членом класу об'єктів. Синтаксис виклику методу такий: ім'я об'єкта й ім'я методу, розділені крапкою.

Події. Щоразу, коли користувач взаємодіє з певним об'єктом на робочому аркуші, відбувається **подія**. Кожен клас об'єктів має власну групу подій на які об'єкти даного класу реагують. Так, робоча книга має події:

- подія NewSheet, яка виконується щоразу, коли користувач додає до неї новий аркуш;

14% від собівартості, якщо собівартість ≥ 3000 грн та ≤ 4000 грн;

16% від собівартості, якщо собівартість > 4000 грн.

7. Обчислити прибутки фірми-посередника від продажу одиниці товару, якщо продається довільна кількість товарів, і оплата за посередництво становить:

6% від ціни, якщо ціна < 650 грн;

8% від ціни, якщо ціна ≥ 650 грн та ≤ 800 грн;

10% від ціни, якщо ціна > 800 грн.

8. Обчислити суму "До виплати" для довільної кількості працівників, якщо задано заробіток, а податок становить:

14% від заробітку, якщо заробіток < 7000 грн;

16% від заробітку, якщо заробіток ≥ 7000 грн та ≤ 9000 грн;

19% від заробітку, якщо заробіток > 9000 грн.

9. Для довільної кількості працівників для заданого заробітку обчислити допомогу, яка нараховується:

13% від заробітку, якщо заробіток < 4000 грн;

11% від заробітку, якщо заробіток ≥ 4000 грн та ≤ 11000 грн;

12% від заробітку, якщо заробіток > 11000 грн.

Обчислити сумарний заробіток.

10. Для довільної кількості назв товарів при заданій собівартості обчислити ціну, яка становить: собівартість + торговий додаток. Додаток обчислюється:

11% від собівартості, якщо собівартість < 1700 грн;

14% від собівартості, якщо собівартість ≥ 1700 грн та ≤ 2500 грн;

17% від собівартості, якщо собівартість > 2500 грн.

11. Обчислити прибутки фірми-посередника від продажу одиниці товару, якщо продається довільна кількість товарів, і оплата за посередництво становить:

9% від ціни, якщо ціна < 1500 грн;

11% від ціни, якщо ціна ≥ 1500 грн та ≤ 3000 грн;

14% від ціни, якщо ціна > 3000 грн.

12. Обчислити суму "До виплати" для довільної кількості працівників, якщо задано заробіток, а податок становить:

14% від заробітку, якщо заробіток < 950 грн;

16% від заробітку, якщо заробіток ≥ 950 грн та ≤ 1500 грн;

18% від заробітку, якщо заробіток > 1500 грн.

13. Для довільної кількості працівників для заданого заробітку обчислити допомогу, яка нараховується:

6% від заробітку, якщо заробіток < 4000 грн;

8% від заробітку, якщо заробіток ≥ 4000 грн та ≤ 9000 грн;

10% від заробітку, якщо заробіток > 9000 грн.

Обчислити сумарний заробіток

14. Для довільної кількості назв товарів при заданій собівартості обчислити ціну, яка становить: собівартість + торговий додаток. Додаток обчислюється:

Програмний код процедури Upcomis

```
Private Sub Upcomis()  
Dim C As Variant  
Dim i As Integer  
i = 8  
For Each C In Range("Продажі")  
Cells(i, 3) = comis(C)  
i = i + 1  
Next C  
End Sub
```

3. Індивідуальні завдання

1. Для довільної кількості працівників при заданому заробітку обчислити допомогу, яка нараховується:

15% від заробітку, якщо заробіток <6000 грн;
10% від заробітку, якщо заробіток >=6000 грн та <=10000 грн;
5% від заробітку, якщо заробіток >10000 грн.

Обчислити сумарний заробіток.

2. Для довільної кількості назв товарів при заданій собівартості обчислити ціну, яка становить: собівартість + торговий додаток. Додаток обчислюється:

6% від собівартості, якщо собівартість <1000 грн;
8% від собівартості, якщо собівартість >=1000 грн та <=2000 грн;
18% від собівартості, якщо собівартість >2000 грн.

3. Обчислити прибутки фірми-посередника від продажу одиниці товару, якщо продається довільна кількість товарів і оплата за посередництво становить:

7% від ціни, якщо ціна <500 грн;
9% від ціни, якщо ціна >=500 грн та <=1000 грн;
10% від ціни, якщо ціна >1000 грн.

4. Обчислити суму "До виплати" для довільної кількості працівників, якщо задано заробіток, а податок становить:

10% від заробітку, якщо заробіток <8000 грн.
15% від заробітку, якщо заробіток >=8000 грн. та <=11000 грн.
20% від заробітку, якщо заробіток >11000 грн.

5. Для довільної кількості працівників при заданому заробітку обчислити допомогу, яка нараховується:

11% від заробітку, якщо заробіток <4500 грн;
7% від заробітку, якщо заробіток >=6000 грн та <=10000 грн;
4% від заробітку, якщо заробіток >10000 грн.

Обчислити сумарний заробіток.

6. Для довільної кількості назв товарів при заданій собівартості обчислити ціну, яка становить: собівартість + торговий додаток. Додаток обчислюється:

12% від собівартості, якщо собівартість <3000 грн;

- подія SelectionChange відбувається при виборі користувачем іншої комірки або діапазону комірок.

При події створення процедури MS Excel автоматично додає в її код перший і останній рядки.

При бажанні можна самостійно написати програму для оброблення події, на яку повинні реагувати об'єкти. Для цього необхідно знати основні правила взаємодії об'єктів, властивостей і методів. Наприклад, коли Ви створюєте аркуш у робочій книзі, функцією-членом у класі Worksheet створюється об'єкт Worksheet. Значення властивостей нового об'єкта присвоюються за замовчуванням.

4.2. Процедури і функції VBA

У VBA підтримується така програмна структура коли на вищому рівні ієрархії розташовані додаток, а далі йдуть проекти, пов'язані з фактичними документами цього додатка. На третьому рівні перебувають модулі (модулі додатка, модулі користувача, модулі класу, модулі форм і модулі посилань), а на нижчому – їхні процедури та функції.

Модуль – це частина програми оформлена у вигляді, який допускає її незалежну трансляцію. Модуль складається із двох розділів: розділу оголошень (Declaration) і розділу процедур і функцій. У першому розділі описуються глобальні змінні, типи, визначені користувачем, у другому – процедури та функції.

Процедурою називається фрагмент програмного коду (мінімальна, семантично закінчена програмна конструкція), розміщений між операторами Sub і End Sub.

Спрощений формат процедури є таким:

```
[Private|Public] Sub <ім'я_процедури> (<аргумент_1>, ... , <аргумент N>)  
<оператор 1>
```

...

```
<оператор n>
```

```
End Sub
```

Формат функцій трохи відрізняється від синтаксису процедур, тому що для функції необхідно вказати тип результату, який повертається нею:

```
[Private|Public] Function <ім'я_функції> (<аргумент_1>, ... , <аргумент N>)  
As <тип_результату>  
<оператор 1>
```

...

```
<оператор n>
```

```
<ім'я_функції> = <результат>
```

```
End Function
```

Ключове слово Private обмежує область видимості процедури або функції, для якої воно задане модулем, у якому вона описана. У результаті її

можуть викликати тільки процедури або функції цього ж модуля. Ключове слово Public, навпаки, оголошує процедуру або функцію доступною для всіх модулів проекту. За замовчуванням (попередньою домовленістю, згодою) довільна процедура або функція є загальнодоступною, тобто має статус Public.

Виконання програмного коду процедури або функції відбувається при передаванні їй керування. Передавання керування (виклик) може здійснюватися різними способами. У загальному випадку підпрограма викликається із програмного коду за допомогою спеціального оператора Call, у якому, крім цього ключового слова, вказуються ім'я процедури і фактичні параметри виклику, список яких записується у круглих дужках.

Дуже важливу роль у програмуванні на VBA відіграє такий різновид процедури, як процедура без параметрів. Процедури цього типу можуть виступати у ролі процедур оброблення подій. Процедури оброблення подій є основою механізму зв'язку подій з програмним кодом для їхнього оброблення.

Функції в VBA, як правило, містять списки параметрів. Вони не можуть використовуватися для оброблення подій. Викликати ж їх можна як за допомогою оператора Call, так і безпосередньо у процесі обчислення значення виразу VBA (з коду інших процедур або функцій) або формули робочого аркуша MS Excel.

Для створення функції користувача необхідно:

- завантажити Редактор VBA;
- у пункті Insert (Вставка) головного меню Редактора вибрати підпункт Module (Модуль) та натиснути клавішу Enter;
- у пункті Insert головного меню Редактора вибрати підпункт Procedure та натиснути клавішу Enter.

У результаті у вікні Редактора VBA візуалізується вікно у якому необхідно вибрати тип процедури "Function", а у полі "Name" вказати ім'я функції, наприклад, st та клацнути "мишкою" на клавіші "OK" діалогового вікна. Наслідком виконання таких дій є візуалізування у вікні коду рядка заголовку процедури-функції та рядка, який замикає розділ описової та виконуваної частин програмного коду

Кутові дужки <...> – це засіб формалізованого опису конструкцій VBA. У конкретних програмах їх не використовують.

Аналіз поданої структури коду процедури свідчить, що кожна функція користувача починається з службового (ключового) слова Function, за яким записується ім'я функції. Ім'я функції повинно обов'язково починатися з латинської літери. В імені не бажано використовувати арифметичні символи, символи логічних операцій або відношень.

У круглих дужках записуються послідовності аргументів функції та назв їх типів. Після дужок записують тип результату виконання функції.

Структуру програмного коду функцій розглянемо на прикладі задачі обчислення площі S трикутника зі сторонами a, b і c за формулою Герона

$$S = \sqrt{P(P-a)(P-b)(P-c)}, \quad (4.1)$$

вибравши пункт головного меню Вставка, Ім'я, Присвоить та вказується ім'я діапазону комірок.

1. Завдання

Заповнити вказану Excel-таблицю, не заповнюючи діапазон комірок G8:H12.

Обчислити суму комісійних, які нараховуються за такою схемою.

Комісійні нараховуються за правилом 5% від суми продажу + премія.

Премія дорівнює:

1% від продажу, якщо сума продажі більша або дорівнює 5000 грн та менша або дорівнює 10000 грн;

2% від продажу, якщо сума продажу більша або дорівнює 10000 грн та менша або рівна 15000 грн;

3% від продажу, якщо сума продажу більша або дорівнює 15000 грн.

2. Порядок виконання роботи

Дана програма має модульну структуру: головна програма (процедура Urcomis) та підпрограма-функція Comis. Головна програма для всіх комірок діапазону з іменем Продажі зопераційсноє виклик процедури-функції Comis, яка зопераційсноє, залежно від значень суми продаж, обчислення премій.

	A	B	C	D	E	F	G	H	I
5			Суми комісійних						
6		Постановка				Результат			
7	Компанія	Продажі	Комісійні			Компанія	Продажі	Комісійні	
8	IBM	15200				IBM	15200	212	
9	Motorola	12800				Motorola	12800	134	
10	Intel	6000				Intel	6000	10	
11	Cyrix	7500				Cyrix	7500	25	
12	AMD	16000				AMD	16000	260	

Програмний код процедури Comis:

```
Private Function comis(s As Variant) As Variant
Dim Runtot As Variant, Bonus As Variant
Runtot = s * 0.05
Bonus = 0
If (s > 5000) And (s < 10000) Then
Bonus = 0.01 * (s - 5000)
Runtot = Runtot + Bonus
End If
If (s > 10000) And (s < 15000) Then
Bonus = 0.02 * (s - 10000)
Runtot = Runtot + Bonus
End If
If s > 15000 Then
Bonus = 0.03 * (s - 15000)
Runtot = Runtot + Bonus
End If
comis = Runtot
End Function
```


Лабораторна робота 9.

Розрахунок заробітної плати з використанням оператора циклу FOR

1. Завдання

	A	B	C	D	E	F	G	H	I
1			Розрахунок заробітної плати			0			
2	№	Прізвище	Оклад	Стаж	Пільги	Премія	нараховано	Податок	До виплати
3	1	Грак	23	10	0	2,3	25,3	0	25,3
4	2	Доцко	230	3	0	0	230	46	184
5	3	Мельник	240	15	1	24	264	0	264

Розрахувати заробітну плату, якщо премія нараховується у розмірі 10%, якщо стаж роботи більший або дорівнює 10 років. Податок становить 20%, якщо нараховано більше, ніж 7000 грн. У іншому випадку податок не обчислюється. Пільговики податок не сплачують.

Початкові дані для обчислень задано діапазоном комірок C3:E5. Результати обчислень необхідно помістити у діапазон комірок F3:I5.

Програмний код процедури:

```
Public Sub zarp()
Dim i As Integer
For i = 3 To 5
If Cells(i, 4).Value >= 10 Then Cells(i, 6).Value = Cells(i, 3).Value * 0.1
Else Cells(i, 6).Value = 0
Cells(i, 7).Value = Cells(i, 3).Value + Cells(i, 6).Value
If Cells(i, 5).Value = 0 Then
If Cells(i, 7) <= 70 Then Cells(i, 8).Value = 0
Else Cells(i, 8).Value = Cells(i, 7) * 0.2
End If
If Cells(i, 5).Value <> 0 Then Cells(i, 8) = 0
Cells(i, 9).Value = Cells(i, 7).Value - Cells(i, 8).Value
Next i
End Sub
```

Лабораторна робота 10.

Розрахунок суми комісійних, використовуючи оператор For Each

Для організації циклів використовується оператор For Each, який має формат:

```
For Each Cell in Range(ім'я діапазону)
сукупність операторів, які виконуються в циклі
Next Cell
```

Для оператора. Для усіх комірок, які входять у діапазон комірок, виконати сукупність заданих операторів. Cell – змінна, що описується як variant і вказує на кожну комірку з діапазону. Ім'я діапазону задається в Excel,

де: P – периметр трикутника.

Public Function SG(a as Single, b as Single, c as Single) as Single

Dim P as Single

‘Периметр трикутника

P=a+b+c

‘Площа трикутника

SG=(P*(P-a)*(P-b)*(P-c))^0.5

End Function

Аналіз поданого програмного коду свідчить, що важливим розділом програмного коду функції користувача є рядки, які містяться між першим та останнім рядками. Програмний код складається з двох частин (підрозділів): частини оголошення констант, змінних та коментарів; виконуваної частини.

Коментар – це фрагмент процедури, який починається з апострофу. Він слугує для пояснення (тлумачення) функціонування програмного коду і не впливає на виконання операторів. Він може бути розташований у довільному місці програмного коду.

Для обчислення площі трикутника зі сторонами, значення довжин яких дорівнюють, наприклад, 5, 8 і 12, відповідно, необхідно клацнути "мишкою" на значку панелі інструментів вікна Редактора VBA. У результаті на моніторі візуалізується робочий аркуш книги Excel. Вибрати у аркуші комірку, до якої потрібно занести результати обчислення площі трикутника. Після цього викликати вмонтовану в Excel функцію "Майстер функцій" та зі списку "Категорії" вікна "Майстра функцій" вибрати категорію "Користувацькі функції", а зі списку "Функції" – функцію з іменем SG. У результаті візуалізується діалогове вікно створеної функції користувача в яке заносяться значення аргументів a, b і c. Після цього клацнути "мишкою" на клавіші ОК діалогового вікна "Майстра функцій". Наслідком виконання такої дії буде результат – значення площі трикутника, яке візуалізується у обраній комірці Excel.

4.3. Константи

Константи – це контейнер для зберігання даних, але, на відміну від змінних, вони не змінюють числового значення у процесі виконання VBA-програми. Константи використовують у наступних випадках:

- програмний код стає більш читабельним, усуваються потенційні помилки;
- щоб змінити якість значення у програмному коді, це потрібно зробити всього один раз – в оголошенні константи.

У VBA константи визначаються за допомогою ключового слова Const:

```
Const COMP_NAME As String = "Student_1"
```

Область видимості константи залежить від того, в якому місці модуля вона оголошена. Щоб константа була доступна тільки всередині однієї

процедури, вона повинна бути оголошена після оператора Sub або Function. Така константа називається локальною.

Константа, оголошена перед першою процедурою в модулі, доступна у всіх процедурах модуля.

Щоб константа була доступу у всіх модулях робочої книги, вона має бути оголошена з ключовим словом Public перед першою процедурою модуля.

Використання у програмному кодї іменованих констант замість явного використання певних значень і рядків надає ряд переваг. Наприклад, якщо у процедурі потрібно звернутися до деякого значення декілька разів, тоді краще оголосити це значення як константу і застосовувати у виразах ім'я константи, а не явне числове значення. У цьому випадку програмний код буде більш наочним і зрозумілим. Крім того, змінити значення константи можна в одному операторі, а не у декількох.

Головна відмінність констант від змінних полягає у тому, що при спробі змінити значення константи у тілі процедури буде видане повідомлення про помилку.

У VBA вмонтовано безліч службових констант: календарних, для роботи з файлами; формами; типами дисків тощо. Переглянути їх можна через довідкову систему VBA: Microsoft Visual Basic Documentation; Visual Basic Language Reference; Constants. Про одну з констант (вона знаходиться у розділі Miscellaneous Constants) слід сказати особливо. Константа vbCrLf дозволяє провести перехід на новий рядок, наприклад:

"Перший рядок" + vbCrLf "Другий рядок"

4.4. Змінні

Описані у процедурах VBA дії виконуються над змінними або об'єктами. **Змінна** – це поіменована область пам'яті, яка використовується для зберігання даних протягом виконання процедури. Використанню змінної, звичайно, передує її оголошення за допомогою оператора Dim, у якому вказуються ім'я (ідентифікатор) змінної і тип значення. Хоча такий оператор не є обов'язковим (VBA здатний ідентифікувати тип значення змінної за останнім символом її імені: !; #; \$; %; &; @), проте, хороший стиль програмування вимагає попереднього опису всіх ідентифікаторів використовуваних змінних.

Імена змінних можуть мати довжину до 255 допустимих символів і не повинні містити пробілів, розділових знаків і перерахованих вище спеціальних символів (за винятком символу @). Вони не можуть збігатися з ключовими (зарезервованими) словами VBA або іменами стандартних об'єктів. Приклади опису змінних:

Dim x, y As Integer

Dim Holiday As Date

Тип повідомлення	Константа
Критичне повідомлення. Відбулась поважна помилка.	VbCritical
Попереджувальне повідомлення. Можлива помилка.	VbExclamation
Запитальне повідомлення. Необхідна додаткова інформація	VbQuestion
Інформаційне повідомлення	VbInformation

Набір клавіш	Константа
OK	VbOKonly
OK,CANCEL	VbOKCANCEL
ABORT,RETRY,IGNORE	VbABORTRETRYCANCEL
YES,NO,CANCEL	VbYESNOCANCEL
YES,NO	VbYESNO
RETRY,CANCEL	VbRETRYCANCEL

Клавіша	Значення, яке повертає функція
OK	1
CANCEL	2
ABORT	3
RETRY	4
IGNORE	5
YES	6
NO	7

1. Завдання

Вивести на екран вікно повідомлення критичного типу, у вікні помістити текст "Поважні помилки. Продовжити?", та клавіші YES, NO, CANCEL. У заголовку помістити – "Увага".

Програмний код процедури:

Public Sub povidom()

MsgBox "Поважні помилки.Продовжити?", vbCritical + vbYesNoCancel, Увага!!"

End Sub

2. Завдання для самостійної роботи у лабораторії

1. Вивести на екран повідомлення попереджувального типу з текстом "Перевірте вхідну інформацію", із заголовком "Будьте уважні!!! " та клавішами YES та NO.

2. Вивести на екран повідомлення запитального типу з текстом "Продовжити роботу? ", із заголовком "Обчислення статистичних величин!!! " та клавішами OK та CANCEL.

3. Вивести на екран повідомлення інформаційного типу з текстом "Проводиться розрахунок", із заголовком "Економічні задачі" та кнопкою OK.

2. Порядок виконання роботи

Розробити вказану форму і для опрацювання події Click() ввести подану нижче процедуру.

Програмний код процедури для опрацювання події Click():

```
Private Sub CommandButton1_Click()
```

```
Dim s1, s2, s3, s4 As String
```

```
Dim k As Integer
```

```
s1 = txtb1.Text
```

```
s2 = txtb2.Text
```

```
s3 = txtb3.Text
```

```
'об'єднання
```

```
txtb4.Text = s2 & s3
```

```
s4 = txtb5.Text
```

```
'позиція входження
```

```
k = InStr(s1, s4)
```

```
txtb6.Text = k
```

```
'усічення символів
```

```
txtb11.Text = Left(s1, 2)
```

```
txtb12.Text = Right(s1, 2)
```

```
txtb13 = Mid(s1, 3, 2)
```

```
End Sub
```

3. Завдання для самоконтролю

Розробити форму, в якій поміщається початковий рядок, наприклад, Облік та аудит і результати роботи таких функцій:

перетворити всі літери на великі;

відзначити із заданого рядка з 3 позиції підрядок довжиною 6 символів;

визначити, з якої позиції символи "ау" входять в заданий рядок.

Лабораторна робота 8.

Використання вікон вводу, відлагодження та повідомлень

Вікно вводу активується процедурою ([...] необов'язкові параметри)

InputBox(повідомлення,[заголовок вікна],[значення початкове])

Вікно відлагодження активується процедурою

Debug.Print повідомлення, назва змінної

Вікно повідомлення активується процедурою

MsgBox повідомлення,[опції],[заголовок вікна]

Опції визначають, які клавіші або піктограми (тип повідомлення) будуть у вікні, якщо опцій декілька, тому вони з'єднуються знаком "+". Аналогічне вікно повідомлень можна викликати функцією:

змінна=MsgBox(повідомлення,[опції],[заголовок вікна]),

де змінна визначає натиснену клавішу.

4.4.1. Типи змінних

У процедурах і функціях VBA можуть використовуватися змінні таких типів:

- Boolean (логічні змінні, які приймають одне з двох значень: True або False);

- Byte (ціле число від 0 до 255);

- Currency (змінні для грошових обчислень з фіксованою кількістю розрядів після десяткової коми, які дозволяють уникнути нагромадження похибок при заокругленні);

- Date (змінні для зберігання дати й часу);

- Double (змінні, які використовуються для зберігання значень із плаваючою комою подвійної точності);

- Integer (цілі числа від -32 768 до +32 767);

- Long (цілі числа від -2 147 483 648 до +2 147 483 647);

- Object (змінні для зберігання посилань на об'єкти);

- Single (змінні для зберігання значень одинарної точності з плаваючою комою);

- String (змінні для зберігання рядків символів довжиною від 0 до 64 Кбайт);

- Variant (змінні, тип яких визначається типом останнього присвоєного їм значення).

VBA надає можливість користувачеві поряд зі стандартними типами створювати власні типи даних (структури) і зберігати дані в масивах (групах змінних із загальним ім'ям, які розрізняють за комбінацією індексів). Типи даних описуються не тільки для змінних, але й для формальних параметрів процедур і функцій, а також для самих функцій (тип значення, яке повертається функцією).

4.4.2. Область видимості змінних

Область видимості змінної – це область програми, де ім'я змінної вважається доступним (видимим) і, таким чином, можливий доступ до її числового значення.

Існують три рівні видимості змінної:

- процедура (областю видимості є тільки та процедура, у якій змінна оголошена);

- модуль (область видимості – усі процедури модуля у якому змінна оголошена);

- додаток (область видимості – усі процедури всіх модулів активного додатку).

Для досягнення видимості у межах процедури оператор Dim оголошує змінну в довільному місці процедури, але завжди передує операторам, які

використовують її. Час життя такої змінної дорівнює часу виконання процедури, тобто при вході в цю процедуру під змінну відзначається пам'ять і відбувається її ініціалізування. У процесі виконання цієї процедури значення змінної може змінюватися. Після виходу із процедури виділена пам'ять звільняється, отже, втрачається значення змінної.

Якщо необхідно зберегти значення змінної між входженнями до процедури, використовується оператор Static. Дія його аналогічна до дії оператора Dim, але він оголошує статичну змінну. Різниця полягає у тім, що при виході з процедури пам'ять статичної змінної не звільняється, а стає тимчасово недоступною, тому значення змінної зберігається.

Для досягнення видимості за межами процедур необхідно, щоб оператор Dim оголошував змінну в розділі описів Declaration модуля (тобто поза його процедурами).

Таким чином, область видимості змінних VBA визначається місцем їхнього опису. Якщо змінна описана всередині процедури або функції, тоді її область видимості обмежується цією процедурою або функцією. Довільна спроба використати ім'я даної змінної поза процедурою або функцією, які містять її опис, приведе або до помилки, або до створення нової змінної – з тим же ім'ям, але з іншою областю видимості та, природно, значенням.

Якщо змінну описано на рівні модуля (тобто відповідний їй оператор Dim розміщено безпосередньо в модулі, перед описом процедур і функцій), її можна використовувати в довільній процедурі або функції, описаній в цьому модулі. Але якщо передбачається, що змінна буде застосовуватися в процедурах і функціях, описаних у різних модулях проекту, тоді її варто описати в одному із цих модулів, скориставшись замість оператора Dim оператором Public.

4.5. Масиви

Масив – це логічно впорядкована послідовність даних однакового типу, які зберігаються у пам'яті комп'ютера послідовно. Розрізняють статичні і динамічні масиви. Розмірність (часто використовують термін межі) статичного масиву встановлюються на етапі розроблення програмного коду, вони можуть змінюватися тільки у новій версії коду. Динамічні масиви можуть змінювати розмірність у процесі виконання програмного коду.

Звернення до числового значення елемента масиву здійснюється відповідно до його індексу.

Оголошення одновимірного статичного масиву, зазвичай, виконується наступним чином:

```
Dim MyArray(20) As Integer
```

Кількість елементів, які можуть зберігатися у масиві – від нижньої межі масиву (lower bound) до верхньої межі (upper bound) включно. Верхня межа позначає максимальний індекс і може бути тільки константою. Індексвання елементів масиву, за замовчуванням, розпочинається з нуля. Отже, індекс 0

Лабораторна робота 7. Робота з рядками

Для роботи з рядками використовуються такі функції:

& – об'єднання декількох рядків в один рядок;

Len (рядок) – визначення довжини рядка;

left (рядок, кількість символів) – вибрати задану кількість символів із заданого рядка, починаючи з крайнього лівого символу;

right (рядок, кількість символів) – вибрати задану кількість символів із заданого рядка, починаючи з крайнього правого символу;

mid (рядок, позиція відзначення, довжина відзначеного рядка) – вибрати із рядка задану кількість символів, починаючи з вказаної позиції ;

instr (позиція пошуку, заданий рядок, шуканий рядок, чутливість до регістру) – повернути номер позиції, з якої шуканий рядок входить в заданий

ucase (рядок) – повернути рядок, де всі символи великі літери;

lcase (рядок) – повернути рядок, де всі символи малі літери;

str (число) – перетворити числову інформацію в символну.

Завдання. Розробити форму, яку подано нижче.

Вказані у формі поля містять:

txtb1 – початковий рядок;

txtb2, txtb3 – рядки, які об'єднуються;

txtb4 – результуючий рядок після об'єднання;

txtb5 – шуканий рядок, для визначення позиції входження;

txtb6 – визначена позиція входження;

txtb11 – відзначення 2-х символів зліва;

txtb12 – відзначення 2-х символів справа;

txtb13 – відзначення 2-х символів, починаючи з 3-ої позиції.

5. Помістити у форму командну клавішу з назвою cmdAdd та заголовком – "+".

6. Програмування події: клацання мишкою на клавіші "+". Два рази клацнути на командній клавіші і у вікні опрацювання процедури події CLICK(), ввести програмний код процедури.

Програмний код процедури:

```
Private Sub cmdAdd_Click()  
' Опис першого числа  
Dim x as integer  
' Опис другого числа  
Dim y as integer  
' Опис результату  
Dim z as integer  
' Присвоєння значення числам  
x=txtNumOne.text  
y=txtNumTwo.text  
' Отримання результату  
z=x+y  
' Присвоєння значення результату інформаційному полю  
txtNumTotal.text=z  
End Sub
```

7. Виконати програмний код.

3. Індивідуальні завдання

Розробити форму для вводу значень та виводу результату для таких завдань:

Переведення значення кута, заданого у градусах в радіани.

Обчислення гіпотенузи прямокутного трикутника за заданими його двома катетами.

Обчислення площі прямокутника.

Обчислення площі трапеції.

Обчислення площі трикутника.

Обчислення степеневі функції за заданою основою та показником степеня.

Обчислення експоненти довільного числа.

Обчислення кореня числа довільного степеня.

Обчислення довжини кола довільного радіуса.

Обчислення площі кола довільного радіуса.

Перевід величини суми, поданої у гривнях на долари.

Перевід величини суми, поданої у доларах на гривні.

Обчислення довільного відсотку від заданого числа.

Обчислення синуса та косинуса для довільного кута.

Обчислення тангенса та котангенса для довільного кута.

позначає перший елемент масиву MyArray. Індекс 1 – другий і за чергою далі У одновимірному статичному масиві MyArray можуть зберігатися 21 цілочисельне значення елементів.

Тип даних для елементів масиву можна не оголошувати:

```
Dim MyArray (20)
```

У цьому випадку для елементів масиву буде використаний тип даних Variant. Такий масив зможе зберігати у собі значення елементів різних типів даних, але вимоги до обсягу пам'яті, виділеної для зберігання значень елементів масиву будуть вищі і програмний код працюватиме трохи повільніше у порівнянні з використанням масиву, для якого тип даних оголошений явно (наприклад, Integer або String).

Присвоїти значення якомусь окремому елементу масиву (у нашому випадку – першому) можна так:

```
MyArray (0) = 100
```

Пізніше до числового значення першого елементу можна буде звернутися:

```
Cells(2, 5).Value = MyArray (0)
```

VBA дозволяє формувати багатовимірні масиви. При оголошенні багатовимірного масиву описи кожної розмірності (індексів) розділюються комами. Оголошення багатовимірного статичного масиву:

```
Dim OurArray(20, 25) As Integer ' 20*25 цілочисельних значень
```

Кількість індексів у описі масивів визначає розмірність масиву. Одновимірні масиви, за аналогією з термінами лінійної алгебри, називають векторами, двовимірні – матрицями, а тривимірні – тензорами.

Оголошення двовимірного статичного масиву типу Variant:

```
Dim MatrixArray (4, 9)
```

Масиви можна створювати й заповнювати одночасно:

```
Dim MyArray
```

```
MyArray = Array (100, 200, 300, 400, 500)
```

Вказувати розмір масиву необов'язково – він буде автоматично налаштований у відповідності з кількістю переданих елементів.

Очистити масив можна командою Erase:

```
Erase MyArray
```

4.6. Оператор присвоєння

Прості лінійні програми складаються з операторів присвоєння, введення/виведення даних та викликів процедур. Оператор присвоєння використовують для присвоєння початкового значення змінній або для зміни її поточного значення. Оператор присвоєння має формат:

A=B,

де: A – ім'я змінної; B – числова константа, ім'я змінної або вираз.

Символ "=" є символом присвоєння. Вираз призначений для описування формул, за якими виконуються обчислення. Він може складатися з чисел, змінних, імен функцій, які з'єднані символами операцій.

Дія *оператора*: обчислюється значення виразу В і результат присвоюється змінній А. Попереднє значення змінної А втрачається.

4.6.1. Арифметичні вирази

Арифметичні вирази призначені для описування послідовності дій (операцій) з числовими даними. Результатом обчислення арифметичного виразу є число.

Арифметичні вирази формує користувач відповідно до умов задачі з констант, змінних, функцій, поєднуючи їх за певними правилами символами арифметичних та логічних операцій і відношень.

Над числовими даними виконують такі операції:

- піднесення до степеня (^);
- множення (*) та ділення (/);
- додавання (+) та віднімання (-).

Над даними цілого типу виконують ще дві операції:

- цілочисельного ділення (символ \);
- обчислення остачі від ділення двох чисел (MOD).

Перша операція відкидає дробову частину результату, наприклад, $5 \setminus 2 = 2$, $359 \setminus 100 = 3$, а $5 \text{ MOD } 2 = 1$, $17 \text{ MOD } 3 = 2$.

Приклад. Розробити функції користувача для обчислення цілої частини та остачі від ділення двох цілих чисел X та Y.

```
Public Function CCH(X As Integer, Y As Integer) As Integer
CCH = X \ Y
End Function
```

```
Public Function OST(X As Integer, Y As Integer) As Integer
OST = X Mod Y
End Function
```

Послідовність виконання операцій при обчисленні виразів така ж, як у математиці і визначається правилом пріоритетів (порядком виконання операцій). Є чотири рівні пріоритетів:

- обчислюються значення функцій, якщо вони є у виразі;
- виконуються всі наявні операції піднесення до степеня;
- виконуються операції множення та ділення;
- виконуються операції додавання та віднімання.

Операції однакового рівня виконуються послідовно зліва направо. Для зміни природної послідовності виконання операцій використовують круглі дужки. *Кількість відкритих та закритих дужок у виразі повинна бути*

```
End Sub
```

вверх праворуч

```
Private Sub cmdTopRight_MouseDown(ByVal Button As Integer, ByVal Shift As Integer, ByVal X As Single, ByVal Y As Single)
```

```
lblNotify = "вверх праворуч"
```

```
End Sub
```

вниз ліворуч

```
Private Sub cmdBottomLeft_MouseDown(ByVal Button As Integer, ByVal Shift As Integer, ByVal X As Single, ByVal Y As Single)
```

```
lblNotify = "вниз ліворуч"
```

```
End Sub
```

Для запуску на виконання вибирається пункт головного меню Запуск, Запуск програми, User/Form.

Лабораторна робота 6. Програмування лінійних алгоритмів

Завдання

Розробити форму суматор (вигляд форми подано нижче), який є формою з трьома полями: поле вводу першого числа; поле вводу другого числа; поле виводу результату. Отримання результату здійснюється натискуванням лівої клавіші "мишки" на командну клавішу "+".

2. Порядок виконання роботи

1. Вставити у вікно проекту форму. Задати заголовок форми – Суматор.
2. У дану форму помістити поля для вводу чисел та виводу результату. Дані поля утворюються за допомогою елемента керування `abl`.
3. Надати назву першому полю вводу – `txtNumOne`, другому полю – `txtNumTwo`, полю результату – `txtNumTotal`.
4. Задати, за допомогою позначки, підписи для відповідних полів: перше число; друге число; результат.

Властивість Top вказує позицію форми відносно верхнього краю екрану.

Програмні коди процедур:

Для клавіші вверх ліворуч

```
Private Sub cmdTopLeft_Click()
    frmmove.Top = 0
    frmmove.Left = 0
End Sub
```

Повернутись в режим форми за допомогою пункту головного меню Вид, підпункт Об'єкт. Аналогічні операції, проробляємо для клавіш:

вниз ліворуч

```
Private Sub cmdBottomLeft_Click()
    frmmove.Top = 200
    frmmove.Left = 0
End Sub
```

вниз праворуч

```
Private Sub cmdBottomRigh_Click()
    frmmove.Top = 200
    frmmove.Left = 300
End Sub
```

вверх праворуч

```
Private Sub cmdTopRigh_Click()
    frmmove.Top = 0
    frmmove.Left = 300
End Sub
```

4. Додавання повідомлень про події

При натискуванні на командній клавіші у центрі форми повинно появлятися повідомлення про позицію переміщення форми.

Коли користувач клацає на клавіші, відбуваються події: мишку вниз – MouseDown, мишку вверх – MouseUp, передавання управління клавіші, тобто клавіша у фокусі, подія – GotFocus.

Клацаємо на командній клавіші вверх ліворуч, появляється вікно, у верхній частині якого міститься список пооперацій, вибираємо пооперацію MouseDown. У вікні, яке появилось, задаємо процедуру опрацювання даної події, причому 1, 2 та 4 рядки з'являються автоматично.

Програмні коди процедур для клавіш:

вверх ліворуч

```
Private Sub cmdTopleft_MouseDown(ByVal Button As Integer, ByVal Shift As Integer, ByVal X As Single, ByVal Y As Single)
    lblNotify = "вверх ліворуч"
End Sub
```

вниз праворуч

```
Private Sub cmdBottomRigh_MouseDown(ByVal Button As Integer, ByVal Shift As Integer, ByVal X As Single, ByVal Y As Single)
    lblNotify = "вниз праворуч"
```

однаковою. Спочатку обчислюють вирази у дужках. Якщо є вкладені дужки, тоді спочатку обчислюють вирази у внутрішніх дужках, а потім у зовнішніх.

4.6.2. Стандартні математичні функції

Функції поділяють на стандартні та нестандартні. Аргумент функції завжди записують у круглих дужках. Аргументом може бути константа, змінна, арифметичний вираз, інша стандартна функція. Основні стандартні функції подані у табл. 4.1.

Таблиця 4.1

Стандартні функції

Функція	Математичний запис	Коментар
SIN(X)	$\sin x$	x задають у радіанах
COS(X)	$\cos x$	x задають у радіанах
TAN(X)	$\operatorname{tg} x$	x задають у радіанах
ATAN(X)	$\operatorname{arctg} x$	арктангенс x
ABS(X)	$ x $	модуль (абсолютна величина x)
SQR(X)	\sqrt{x}	квадратний корінь x
LOG(X)	$\ln x$	логарифм натуральний x
EXP(X)	e^x	експонента
INT(X)	$[x]$	найбільше ціле число яке не перевищує x
FIX(X)		ціле число, що утворюється відкиданням дробової частини числа x
RND(X)		випадкове число з інтервалу [1;0)
SGN(X)	$\operatorname{sign} x$	знак числа $\operatorname{sign} x = \begin{cases} 1, & \text{якщо } x > 0 \\ 0, & \text{якщо } x = 0 \\ -1, & \text{якщо } x < 0 \end{cases}$

4.6.3. Нестандартні математичні функції

До нестандартних математичних функцій відносяться: логарифми з основою відмінною від числа $e \approx 2,718282$; тригонометричні функції, які визначаються за допомогою стандартних тригонометричних функцій, поданих у табл. 4.1, тощо.

Наприклад, обчислення логарифма з основою $a \neq e$ ґрунтується на співвідношенні

$$\log_a x = \frac{\ln x}{\ln a} \quad (4.2)$$

Public Function Logax(X As Single, a As Single) As Single

Logax = Log(X) / Log(a)

End Function

В основі обчислення нестандартних тригонометричних функцій (котангенс, арксинус, арккосинус, арккотангенс) покладені відомі з курсу вищої математики формули:

$$ctgx = \frac{1}{tgx}; \quad (4.3)$$

$$arcsinx = arctg\left(\frac{x}{\text{Sqr}(-x^2 + 1)}\right); \quad (4.4)$$

$$arccosx = arctg\left(\frac{x}{\text{Sqr}(-x^2 + 1)}\right) + 2arctg(1); \quad (4.5)$$

$$arccotgx = arctgx + 2arctg(1). \quad (4.6)$$

Програмні коди функцій користувача для обчислення залежностей $ctgx$, $arcsinx$, $arccosx$, $arctgx$, згідно з (4.3) – (4.6), матимуть, відповідно, вигляд:

- обчислення $ctgx$

Public Function Cotan(X As Single) As Single

Cotan = 1 / Tan(X)

End Function

- обчислення $arcsinx$

Public Function Arcsin(X As Single) As Single

Arcsin = Atan(X / Sqr(-X * X + 1))

End Function

- обчислення $arccosx$

Public Function Arccos(X As Single) As Single

Arccos = Atan(-X / Sqr(-X * X + 1)) + 2 * Atan(1)

End Function

- обчислення $arctgx$

Public Function Arccotan(X As Single) As Single

Arccotan = Atan(X) + 2 * Atan(1)

End Function

4.7. Введення і вивід інформації

У VBA є засоби для обміну інформацією між фізичними носіями і оперативною пам'яттю – **оператори вводу/виводу**. Докладний опис їх істотно збільшив би обсяг даного навчального видання, тому детально розглянуто лише дві функції, найбільш корисні і широко використовувані при обміні інформацією між програмою й типовими пристроями вводу/виводу (дисплеєм і клавіатурою). Це функція виводу на екран MsgBox і функція введення з клавіатури InputBox. Функція MsgBox має такий формат:

MsgBox (*prompt* [, *buttons*] [, *title*] [, *helpfile*, *context*])

управління, зробивши підписи: командна клавіша, позначка, прапорець.



1. Завдання

Створити форму, яка містить 4 командні клавіші, за допомогою яких відбувається переміщення даної форми у положення вгору ліворуч, вниз ліворуч, вгору праворуч, вниз праворуч. У центрі форми появляється повідомлення про положення форми.

Виконання завдання складається з таких кроків:

1. Розроблення форми, яка містить відповідні командні клавіші, з відповідними підписами.
2. Програмування опрацювання події "клацання мишкою", тобто події CLICK.

3. Додавання повідомлення про події, які відбуваються з формою.

2. Порядок виконання роботи

1. Активувати Редактор VBA.
2. За допомогою пункту головного меню Вставка, UserForm, задаємо форму.
3. Назвемо форму frmmove, задамо заголовок Переміщення.
4. Із вікна Панель елементів перетягуємо 4 командні клавіші і симетрично розміщуємо їх у кутках форми. У центрі форми, за допомогою значка A із Панелі елементів створюємо позначку і задаємо ім'я lblNotify.

5. Почергово активуємо кожну клавішу і, у залежності від їх розміщення, надаємо у вікні властивостей назви – cmdTopLeft і підпис клавіші – "вгору ліворуч", аналогічно для інших клавіш: назва – cmdTopRight підпис – "вгору праворуч"; назва – cmdBottomLeft, підпис – "вниз ліворуч"; назва – cmdBottomRight, підпис – "вниз праворуч".

3. Опрацювання події CLICK()

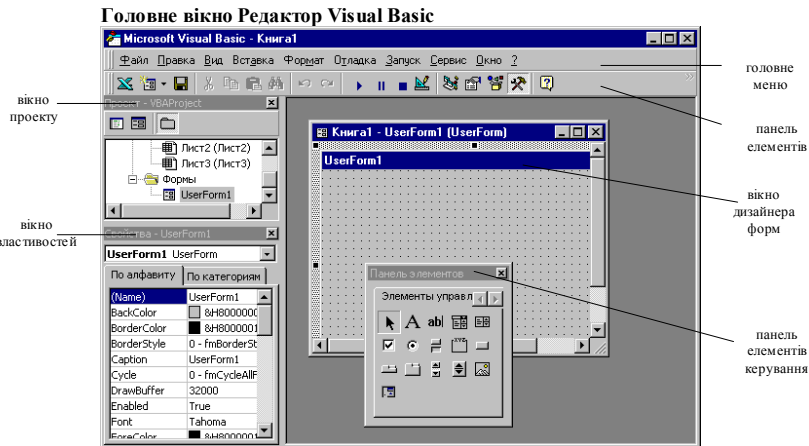
Двічі клацнути кнопкою на клавіші вгору ліворуч. Появляється вікно процедури для опрацювання події: клацання на верхній лівій командній клавіші. Набираємо текст процедури, крім першого і останнього оператора, оскільки вони з'являються автоматично.

Властивість Left вказує позицію форми відносно лівого краю екрану.

вітановлюємо вказівник "мишки" на маркери, які розташовані на межі форми, і розтягуємо її у потрібному напрямку.

4. Для того, щоб змінити назву форми, у вікні властивостей поля Name задаємо своє ім'я – frmmain.

5. Для того, щоб задати заголовок форми, необхідно у вікні Свойства поля Caption ввести назву – Форма.



2. Робота з елементами управління

1. Для того, щоб вставити Панель елементів, яка містить елементи управління, вибирається пункт головного меню Вид, Панель Елементов (звичайно панель є на екрані) або піктограма

2. Для того, щоб вставити елемент керування у форму, вибираємо потрібний, клацнувши лівою клавішею "мишки", і не відпускаючи її перетягуємо у форму. Вставте у свою форму командну клавішу

(клавіша), прапорець , позначку A (напис), інформаційне поле abl (поле).

3. Імена елементів управління змінюються аналогічно, як у формах. Задайте елементам управління свої власні імена: командній клавіші – cmdMain; позначці – lblMain; полям виводу – txtMain; прапорцеві – chkmain.

4. Для перенесення елемента управління екраном, установлюємо курсор "мишки" на нього, він прийме зображення сніжинки, і не відпускаючи лівої клавіші "мишки", переносимо у потрібне місце.

5. Зміна розмірів елементів управління відбувається аналогічно, як у формах.

6. Для того, щоб поміняти напис, який є на елементі управління, у вікні Свойства поля Caption задаємо потрібну назву. Проте не всі елементи управління мають дану властивість. Поміняйте назви на елементах

Функція виводить повідомлення у діалоговому вікні, очікує натискання клавіші користувачем і повертає ціле число, що відповідає натиснутій клавіші. Всі аргументи, крім prompt є необов'язковими. Функція MsgBox повертає значення, що може бути оброблене в програмі.

Зміст операндів такий:

- prompt – підказка або інформація для виводу (обов'язковий параметр);
- buttons – чисельний вираз, що є сумою значень, що визначають такі характеристики вікна: число й кількість кнопок, стиль значка, кнопку за замовчуванням, модальність вікна. За замовчуванням значення *buttons* дорівнює нулю;
- title – заголовок вікна MsgBox;
- helpfile – посилання на файл допомоги, організований програмістом;
- context – вказівник на розділ у файлі допомоги.

Функція повертає такі значення:

- 1 – було натиснуто клавішу OK;
- 2 – було натиснуто клавішу Cancel;
- 3 – було натиснуто клавішу Abort;
- 4 – було натиснуто клавішу Retry;
- 5 – було натиснуто клавішу Ignore;
- 6 – було натиснуто клавішу Yes;
- 7 – було натиснуто клавішу No.

Функція InputBox має такий формат:

InputBox(prompt [, title] [, default] [, xpos] [, ypos] [, helpfile, context])

Функція відтворює підказку в діалоговому вікні, очікує введення тексту або натискання кнопки й повертає рядкове значення, яке містить введений текст. Всі аргументи, крім prompt є необов'язковими.

Зміст операндів такий:

- prompt – підказка (обов'язковий параметр);
- title – заголовок вікна InputBox;
- default – значення, яке повертає функція у випадку, якщо інформацію користувачем не введено;
- xpos – відстань (в одиницях 1/20 пікселя) лівого краю діалогового вікна від лівої границі екрана (якщо аргумент відсутній – вікно центрується по горизонталі);
- ypos – відстань (в одиницях 1/20 пікселя) верхнього краю діалогового вікна від верхньої границі екрана (якщо аргумент відсутній – вікно центрується по вертикалі);
- helpfile – посилання на файл допомоги, організований програмістом;
- context – вказівник на розділ у файлі допомоги.

РОЗДІЛ 5. СУБД MICROSOFT ACCESS

1) Будь-яка інформаційна система створюється на основі бази даних (БД). База даних – це організована структура даних певної предметної області, призначена для збереження інформації, до якої можуть мати доступ багато програм. Такі програми називаються системами управління базами даних (СУБД). На сьогодні розроблено багато СУБД, серед яких найбільш відомими є FoxPro+, dBase, DataEase, Access. Розрізняють три типи баз даних: реляційні, ієрархічні, мережеві.

2) Реляційні бази є найпростішим типом (щодо фізичної організації) і характеризуються наведенням даних у вигляді декількох таблиць і створенням зв'язків між ними. Концепція реляційної бази даних розроблена Е.Ф. Коддом у 1970 р. В основі цієї концепції лежить математичне поняття відношення, яке представляють у вигляді двовимірних таблиць.

3) У порівнянні з іншими моделями реляційна модель вимагає від комплексу програм СУБД набагато більш високого рівню складності. У ній робиться спроба звільнити користувача від необхідності виконання рутинних операцій щодо керування даними, характерних для ієрархічної й мережної моделей.

4) У рядках таблиць частина полів містить дані, що відносяться безпосередньо до запису, а частина - посилання на записи інших таблиць. Таким чином, зв'язки між записами є невід'ємною властивістю реляційної моделі. Кожен запис таблиці має однакову структуру. Наприклад, у таблиці, що містить опис автомобілів, у всіх записів буде той самий набір полів: виробник, модель, рік випуску, пробіг тощо. Такі таблиці легко зображувати в графічному виді.

5) У реляційній моделі досягається інформаційна й структурна незалежність. Записи не пов'язані між собою настільки, щоб зміна одного з них торкнулося інших, а зміна структури бази даних не обов'язково призводить до перекомпіляції працюючих з нею додатків.

6) У реляційних СУБД застосовується мова SQL (Structured Query Language), що дозволяє формулювати довільні, нерегламентовані запити. Це мова четвертого покоління, тому будь-який користувач може швидко навчитися створювати запити. До того ж, існує безліч додатків, що дозволяють будувати логічні схеми запитів у графічному виді. QBE (Query-by-Example) – це графічна версія реляційної мови SQL. Ця мова створювалася на фірмі IBM під керівництвом (Zluff) Злуфа в 1975 р. Вона дістала широке розповсюдження і в даний час реалізована в більшості СУБД.

Реляційним базам даних властиві розходження у реалізації мови SQL, хоча це й не проблема реляційної моделі. Кожна реляційна СУБД реалізує якусь підмножину стандарту SQL плюс набір унікальних команд, що ускладнює завдання програмістам, які намагаються перейти від однієї СУБД до іншої. Доводиться робити нелегкий вибір між максимальною переносимістю й максимальною продуктивністю. У першому випадку потрібно дотримуватися

$$8) a_{ij} = \frac{x + (i - j^{2.3})}{\cos(x + j) - 2.5i};$$

$$b_{ij} = \frac{i \cdot \cos j + x \cdot \sin i}{x + \lg(i + j^2)}.$$

$$9) a_{ij} = x^3 \frac{\cos(i + j^2)}{\sin^2(i^2 + j)} - j^2;$$

$$b_{ij} = \frac{e^{\cos i/x}}{\sin x/j} \cdot \lg(i + j).$$

$$10) a_{ij} = \frac{2i + j}{x5.5 - 2j} \cos^2 \frac{ij}{x};$$

$$b_{ij} = \frac{i^2 - x}{j^2 + x} \cdot \lg(j + 1).$$

$$11) a_{ij} = \sqrt[3]{\frac{i + 3j}{x + j}} \cdot \sin \frac{i}{x};$$

$$b_{ij} = \frac{i + j}{j + x} \cdot \cos(2j + i)$$

$$12) a_{ij} = \frac{x^{1.3} + \lg(i + j)}{x + e^{-0.1j}};$$

$$b_{ij} = \frac{i^2 + j}{x + i/j} \cos(j + i).$$

$$13) a_{ij} = i^{1.3} \cdot \frac{\cos(2j - i)}{\sin^2(x + j)};$$

$$b_{ij} = \frac{e^{i/j}}{\sin(x + i)} \cdot \cos(i + j).$$

$$14) a_{ij} = \frac{x^2 - (i + 3j)^{1.2}}{i + \log_2(j^2/2)};$$

$$b_{ij} = x^2 + \frac{i^2 - \cos j}{x + ij + 1.5}.$$

$$15) a_{ij} = \frac{i + j}{x + i/j} \lg|\cos(i + j)^2|;$$

$$b_{ij} = \frac{\arctg(i + j)}{x^2 + \lg(i + j)}.$$

Обчислити:

кількості додатних елементів		кількості від'ємних елементів	
1)	у кожному із стовпців матриці A	2)	
3)	у кожному із стовпців матриці B , які мають парні номери	4)	
5)	у кожному із стовпців матриці A , які мають непарні номери	6)	
7)	у стовпцях матриці B , за винятком першого і останнього	8)	
суми додатних елементів		суми від'ємних елементів	
9)	у кожному із рядків матриці B	10)	
11)	у кожному із рядків матриці A , які мають парні номери	12)	
13)	у кожному із рядків матриці B , які мають непарні номери	14)	
15)	у рядках матриці A , за винятком першого і останнього	16)	

Лабораторна робота 4.

Робота з формами та елементами управління

Форми – це основні "будівельні блоки програми". Форми – це місце, куди поміщаються всі елементи, з якими працює користувач. Ці елементи називають елементами управління. Ними можуть бути клавіші, прапорці, написи, текстові поля. Форми та елементи управління називають об'єктами.

1. Робота з формами

1. Активувати середовище Excel. Вибрати пункт головного меню Сервіс, підпункт Макроси, з вікна, яке появилось – пункт Редактор Visual Basic.

2. Для того, щоб вставити форму, необхідно вибрати пункт головного меню Вставка, підпункт UserForm. Вставте форму у вікно проекту.

3. Для того, щоб змінити розмір форми, активуємо її як об'єкт,

1) $z = \frac{ C ^{0.25}}{M \cdot m} \sum_{i=1}^{10} a_i^2 - \log_2^3 M + m^2 $	9) $z = \frac{M \cdot m}{C + m} \sum_{i=1}^{10} \frac{i^2}{a_i} - \log_2^{0.3} \left \frac{M}{m} + C ^{0.32} \right $
2) $z = \frac{C + m ^{0.55}}{M \cdot m} \sum_{i=1}^{10} a_i ^{1.2} - \log_3^{0.4} M + m $	10) $z = \frac{M \cdot m}{ C ^{3.26} + m} \sum_{i=1}^{10} \frac{i^3}{a_i} - \log_3^{1.3} \left \frac{M}{m^2} + C \right $
3) $z = \frac{C \cdot m}{M + m} \sum_{i=1}^{10} a_i ^{1/i} - \log_4^{0.5} M + m^2 $	11) $z = \frac{M \cdot m}{C + m} \sum_{i=1}^{10} \frac{i^{1.2}}{a_i} - \log_4^{2.3} \left \frac{M}{m} + C ^{1.32} \right $
4) $z = \frac{M + C}{M \cdot m} \sum_{i=1}^{10} a_i ^2 - \log_5^{0.8} M + m $	12) $z = \frac{M \cdot m}{ C ^{2.26} + m} \sum_{i=1}^{10} \frac{i^{1.6}}{a_i} - \log_5^{3.3} \left \frac{M}{m^2} + C \right $
5) $z = \frac{C + m}{M \cdot m} \sum_{i=1}^{10} \frac{a_i}{ a_i ^{0.56}} - \log_6^{1.3} M + m^2 $	13) $z = \frac{M \cdot m}{C + M} \sum_{i=1}^{10} \frac{a_i}{i^{1.3}} - \log_6^{0.2} \left \frac{m}{M} - C ^{2.32} \right $
6) $z = \frac{C \cdot M ^{0.21}}{M \cdot m} \sum_{i=1}^{10} \frac{ a_i ^{0.56}}{a_i} - \log_7^{2.3} M + m $	14) $z = \frac{M \cdot m}{ C ^{1.26} + M} \sum_{i=1}^{10} \frac{a_i}{i^{2.3}} - \log_7^{1.2} \left \frac{m^2}{M} - C \right $
7) $z = \frac{ C ^{0.2} + m}{M \cdot m} \sum_{i=1}^{10} \frac{a_i}{i^2} - \log_8^{3.3} M + m^2 $	15) $z = \frac{M \cdot m}{C + M} \sum_{i=1}^{10} \frac{a_i}{i^3} - \log_8^{2.2} \left \frac{m}{M} - C ^{3.32} \right $
8) $z = \frac{ M ^{0.3} - C}{M \cdot m} \sum_{i=1}^{10} \frac{i^2}{a_i} - \log_9^{2.3} M + m $	16) $z = \frac{M \cdot m}{ C ^{0.26} + M} \sum_{i=1}^{10} \frac{a_i}{i^2} - \log_9^{3.2} \left \frac{m^2}{M} - C \right $

7. Індивідуальні завдання 5

Задано дійсне число $x=2.7\mathcal{N}_6$ (де \mathcal{N}_6 – номер варіанту) і цілі числа $m=10$ і $n=5$. Відповідно до номера варіанту розробити програмний код для обчислення значень елементів двох матриць

$$\overline{A} = [\overline{A}_i = [a_{ij} = f[x, i, j], j = \overline{1, n}], i = \overline{1, m}] \quad \text{і} \quad \overline{B} = [\overline{B}_i = [b_{ij} = f[x, i, j], j = \overline{1, n}], i = \overline{1, m}]$$

за заданими формулами:

$$1) a_{ij} = \frac{i + j^2}{4.5x - j} \sin^2 ij;$$

$$b_{ij} = \frac{i + j}{x + 3} \cdot \operatorname{tg}^3 j / i.$$

$$2) a_{ij} = \sqrt[3]{\frac{i + j}{0.2x}} \cdot \cos i;$$

$$b_{ij} = \frac{i^3 + j}{x + j} \cdot \sin(i / j)^2$$

$$3) a_{ij} = \sin^2 \frac{i^3 + j^{1.3}}{x + j};$$

$$b_{ij} = \frac{i - x}{i^2 + j} \cos(j / i)^2$$

$$4) a_{ij} = i^3 \cdot \frac{x + \sin j}{xe^{0.2i-j}};$$

$$b_{ij} = \frac{e^{i+x}}{i + j} \cdot \cos^2(x \cdot j).$$

$$5) a_{ij} = \frac{x^{1.3} + \sqrt{i + j}}{i + j^2 / 2};$$

$$b_{ij} = \frac{i^2 - \cos j}{x + 1.5 + ij}.$$

$$6) a_{ij} = i^2 + j^3 \frac{\cos(i + j)}{e^{-(i+2)}};$$

$$b_{ij} = \frac{\operatorname{arctg}(i \cdot j)}{\lg(i/j + x)}.$$

$$7) a_{ij} = \frac{e^i}{x} \cdot \frac{i + \sin j}{2.5 + i \cdot j};$$

$$b_{ij} = \frac{i^3 + j - x}{2i + \cos(x - j)}.$$

мінімального загального набору команд, підтримуваних у кожній СУБД. У другому випадку програміст просто зосереджується на роботі в даній конкретній СУБД, використовуючи переваги її унікальних команд і функцій.

5.1. Об'єкти бази даних

СУБД *Microsoft Access* дозволяє створювати та використовувати об'єкти семи різних типів.

Таблиці – це основні об'єкти будь-якої бази даних, що зберігають усі дані, які є в базі, та структуру бази даних (поля, їхні типи та властивості).

Запити – спеціальні засоби для відбору і аналізу даних, котрі служать для отримання даних з таблиці і представлення їх користувачеві у зручному вигляді. За допомогою запитів виконуються такі операції як відбір даних, сортування та фільтрація. Також можна виконувати перетворення даних за заданим алгоритмом, створювати нові таблиці, виконувати автоматичне наповнення таблиць даними, що імпортуються з інших джерел тощо.

Форми – це спеціальні засоби для введення та відображення записів бази даних. Одночасно у формі можна розміщувати спеціальні елементи управління для автоматизації введення, оформлення та перегляду даних.

Звіти – це спеціальні об'єкти, призначені лише для зображення даних у друкованій формі. Користувач має повний контроль над розміром і виглядом всіх елементів звіту, тому ці дані можна зобразити в зручному для читання вигляді.

Сторінки – це особливий об'єкт, виконаний в коді HTML, який розміщується на WEB-сторінці і передається клієнту разом з нею. Сторінки доступу до даних виконують роль інтерфейсу між клієнтом, сервером і базою даних, яка розміщена на сервері.

Макроси – це команда або набір команд, кожна з яких виконує певну дію: відкриття форм, друкування звітів, запуск запитів на виконання, сортування даних тощо. Макроси дають змогу автоматизувати виконання часто повторюваних дій. Головна властивість макросів – виконуватись у відповідь на ту чи іншу подію. Макрос може бути одиночним (складається з послідовності макрокоманд) або складовим, який входить в групу макросів.

Модулі – це набір описів змінних і процедур мови *Visual Basic*, що зберігаються як єдине ціле. Це один із засобів, за допомогою якого розробник бази може закласти в неї стандартні функціональні можливості, підвищити швидкодію системи, а також рівень її захищеності.

5.2. Робота з таблицями

Створювати таблиці в *Access* можна трьома способами:

- скористатися *Майстром* для побудови таблиць на основі зразків *Access 2007*;

- за допомогою режиму *Конструктора*. Цей метод дає можливість контролювати властивості полів таблиці;
- вводити інформацію в таблицю і надати можливість *СУБД Access* самостійно визначити тип інформації на основі введених даних.

5.2.1. Створення таблиць за допомогою *Майстра*

Для створення таблиці за допомогою *Майстра* потрібно у вікні *База даних* два рази клацнути на кнопці *Створення таблиці за допомогою Майстра*. На екрані відкриється діалогове вікно, що містить описи різних видів таблиць баз даних із зразками полів. Таблиці поділяються на дві групи *Ділові* та *Особисті* (рис. 5.1).

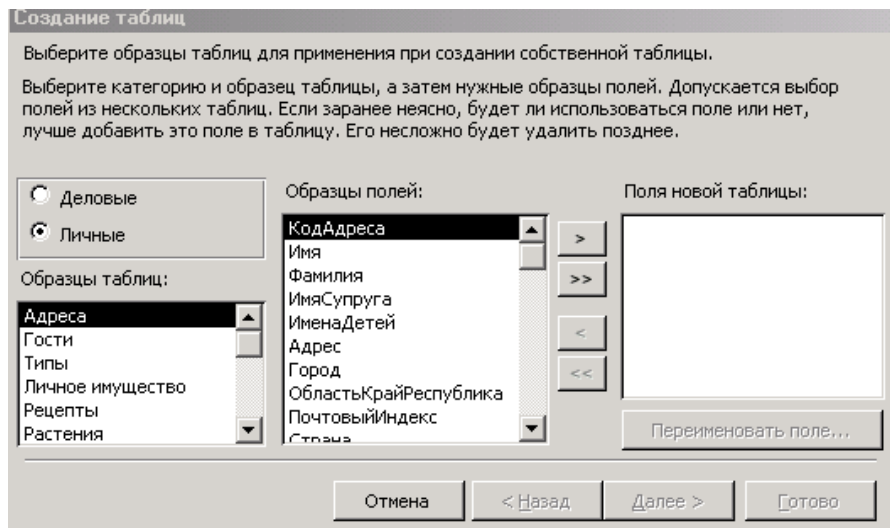


Рис. 5.1. Зразки запропонованих видів таблиць та полів.

Після цього потрібно переглянути список таблиць і знайти варіант, який найбільш підходить для створюваної БД. Клацаючи на кнопках ">" і ">>", вибираємо поля, які потрібні у створюваній таблиці. Додавати необхідні поля можна також і з інших таблиць. Після визначення всіх полів (рис. 5.2), потрібно вибирати кнопку *Далше*. На екрані відкриється діалогове вікно, в яке можна ввести назву створеної таблиці.

```
Cells(nr, ns).Value ="j": Cells(nr + 1, ns - 1).Value ="i"
For j=1 To n
  Cells(nr, ns + j).Value =j
  A(j)=FnAi(x, j)
  Cells(nr + 1, ns + j).Value =A(j)
Next j
For i=1 To n
  Cells(nr + 1 + i, ns - 1).Value =i
  Cells(nr + 1 + i, ns).Value =A(i)
  For j=1 To n
    B(i, j)=FnBij(A(i), A(j))
    Cells(nr + 1 + i, ns + j).Value =B(i, j)
  Next j
Next i
Cells(nr + n + 2, 3).Value ="Кінець розрахунку"
End Sub
```

6. Індивідуальні завдання 4

1. Задано дійсне число $x=0.68 \cdot N_2$ (де N_2 – номер варіанту) і ціле число $n=10$. Відповідно до номера варіанту розробити програмний код для обчислення значень елементів одновимірного масиву $\tilde{A}=\{a_i, i=\overline{1, n}\}$ за заданими формулами:

- 1) $a_i = \frac{i^3 + \cos^2(i+3)}{\log_2(i+x)^{1.24}} - e^{-\frac{x+i}{i}}$;
- 2) $a_i = \frac{3 \cos^3(i+2)}{\log_2(i^2+x)} + \sqrt[4]{\sin^2(i-x)}$;
- 3) $a_i = \frac{i^{2.3} + 3.5}{\sqrt[3]{i^2 - 2x + 5}} + \sin^2 x^{1/i}$;
- 4) $a_i = \frac{i^2 + x \cos i}{\sqrt[4]{|i^3 - 4x|} - 5} - \cos \frac{x^{1/i}}{2 + i^2}$;
- 5) $a_i = \frac{\cos^2(i+3)}{\sqrt{i^{2/i}} - 1.8x} + \sqrt[4]{\log_3^2(i+2)}$;
- 6) $a_i = \frac{\sin(i+3)^2}{i^2 + 1.23x} + \log_2^i(i^2 + 2x)$;
- 7) $a_i = \frac{\cos^3(i+3)}{i^{2/i} + 3.5x} - \sqrt[4]{\log_2(i+x)}$;
- 8) $a_i = \frac{\ln^{1.2}(i+2)}{\sin^2(i+x^{2/i})} + \frac{x + e^{i-5.1}}{\sqrt[4]{i+x}}$;
- 9) $a_i = \frac{i^{3/i} + 3x}{\cos(i+2)} - \frac{\log_3^3(i+1)}{\sin^2(i+x)}$;
- 10) $a_i = \frac{|2-x/i|^{2/i}}{\cos(i+2x)} - \frac{\log_4^{0.3}(i+x)}{\sin^2|i-3|^{0.43}}$;
- 11) $a_i = \frac{\sin^2(i+5x)}{e^{i+3} + x^2} - \frac{\log_5^{0.3}(i+1.5)}{\cos^2(i-3x)}$;
- 12) $a_i = \frac{\cos^2(i-5)}{e^{i+3} - x^2} + \frac{\log_4^{0.2}(i+1.2x)}{\cos^2(i-2.3)}$;
- 13) $a_i = \frac{i^2 + 2.5x}{\cos^2(i^{0.1} + 3)} - \sqrt[4]{\log_i x^{1.3}}$;
- 14) $a_i = \frac{e^{i+1} - x^2}{\cos^2(i-5)} + \frac{\log_3^{0.2}(i+2x)}{\sin(i^2 - 0.3)}$;
- 15) $a_i = \frac{\cos^2(i+5x)}{i + x e^{i+3}} + \lg(i^{1.4} + 0.8)$;
- 16) $a_i = \frac{\sin(i-5)^2}{e^{i+0.6} + x} - \log_3^2(i+x)$.

Обчислити:

- середнє значення елементів масиву (C);
- мінімальне (m) та максимальне (M) значення елементів масиву;
- значення арифметичного виразу:

```
Cells(nr - 2, 6).Value ="Z=": Cells(nr - 2, 7).Value =z
Cells(nr, 6).Value ="Кінець розрахунку"
End Sub
```

5. Зразок виконання завдання 3

Задано дійсне число $x=1.3$ і ціле число $n=10$. Розробити програмний код для обчислення значень елементів одновимірного масиву $\tilde{A}=\{a_i=f[x,i], i=\overline{1,n}\}$,

$$a_i = \frac{i - x^2 + 1}{i^3 + \cos(x + i^2)}.$$

З елементів одновимірного масиву потрібно обчислити значення елементів двовимірного масиву $\tilde{B} = \{B_i = \{b_{ij} = f[a_i, a_j], j = \overline{1,n}\}, i = \overline{1,n}\}$,

$$b_{ij} = a_j \cdot \frac{|a_i - 3.2 \cdot a_j|^{0.3}}{1 + \sin(a_i - 1.2)}.$$

Програмні коди:

' Обчислення значень елементів масиву {A}

```
Function FnAi(x, i) As Single
```

```
Dim a1, a2 As Single
```

```
a1 = i - x ^ 2 + 1
```

```
a2 = i ^ 3 + Cos(x + i ^ 2)
```

```
FnAi = a1 / a2
```

```
End Function
```

' Обчислення елементів матриці

```
Function FnBij(ai, aj) As Single
```

```
Dim b1, b2 As Single
```

```
b1 = Abs(ai - 3.2 * aj) ^ 0.3
```

```
b2 = 1 + Sin(ai - 1.2)
```

```
FnBij = aj * b1 / b2
```

```
End Function
```

```
Sub Zavd_4 ()
```

```
Dim n, nr, ns, i, j As Integer
```

```
Dim A(10), B(10, 10) As Single
```

'Введення вхідних даних

```
nr = 68 ' номер рядка розрахунку
```

```
ns = 3 ' номер стовпця розрахунку
```

```
x = 1.3 * Cells(nr, ns).Value
```

```
Cells(nr, ns + 1).Value ="x=": Cells(nr, ns + 2).Value =x
```

```
n = Cells(nr, ns + 5).Value
```

' Виведення позначення елементів масиву

```
nr = nr + 2: Cells(nr - 1, ns + 1).Value ="Результат розрахунку"
```

' Обчислення та виведення елементів матриці

```
Cells(nr - 1, ns).Value ="B": Cells(nr + 1, ns).Value ="ai\aj"
```

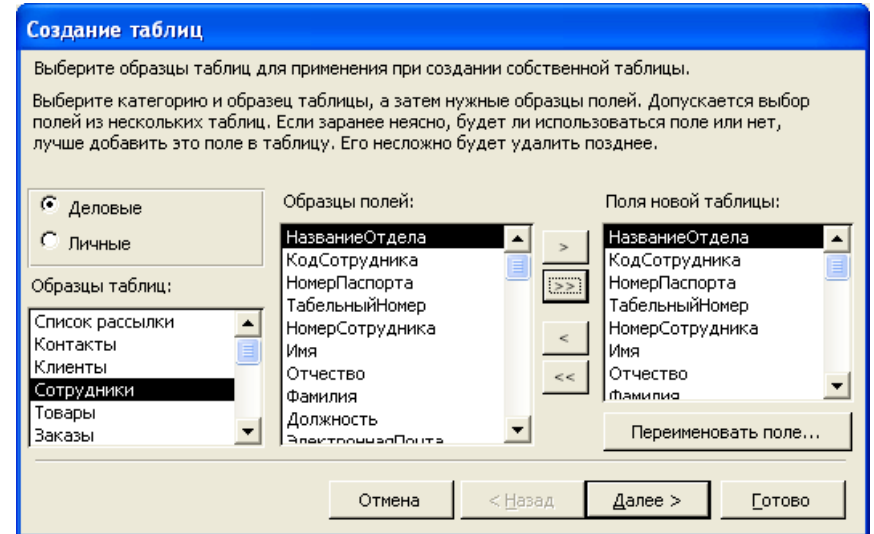


Рис. 5.2. Вибрана у якості зразка таблиця *Співробітники* з усіма полями.

Після введення назви таблиці необхідно відмітити як буде створюватися первинний ключ. Якщо ключ будемо створювати самостійно, то на екран буде виведено вікно, що буде мати наступний вигляд (рис. 5.3).

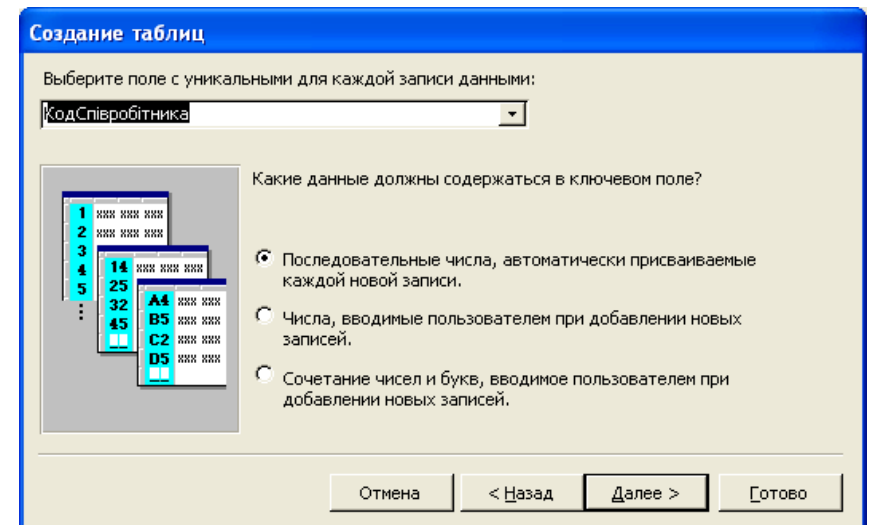


Рис. 5.3. Вибраный первинный ключ *КодСпівробітника* у вигляді послідовності чисел, що автоматично присвоюються кожному запису

Коли Майстер автоматично визначає первинний ключ для таблиці, він переглядає поля відшукуючи поле типу *Autonumber*. Перше знайдене поле такого типу використовується в ролі первинного ключа. Якщо Майстер не знаходить жодного подібного поля він створює нове поле з назвою, що складається з назви таблиці і букв ID.

Якщо ключ буде створюватися самостійно *Майстер* відображає на екрані список полів таблиці. Далі користувач повинен виділити поле, яке буде мати унікальне значення і вибрати його тип. *Майстер* може формувати один з трьох типів первинного ключа. Номер, що створюється Access автоматично, номер, що створюється користувачем і текстовий рядок, що створюється вручну. Для продовження роботи потрібно натиснути кнопку *Даліше*.

Якщо БД складається з декількох таблиць, *Майстер* запитає, чи зв'язана дана таблиця з іншими таблицями. Якщо зв'язків немає, то у вікні *Майстра* можна змінити вигляд таблиці за допомогою *Режиму Конструктора* і ввести дані за допомогою таблиці.

5.2.2. Ключові поля

У базах даних використовують різні типи ключів відношень.

У Access найпростіший спосіб створення ключа – це додавання додаткового поля типу лічильник. Після натискування кнопки із зображенням ключа на панелі інструментів, в лівій частині назви поля відкривається зображення ключика. Це буде означати, що це поле є ключовим. У Access таке поле заповнюється автоматично.

Якщо жодне з полів не є унікальним (неповторним), то ключове поле можна задати через контекстне меню на базі двох (чи більше) полів, попередньо відзначивши їх (рис. 5.4).

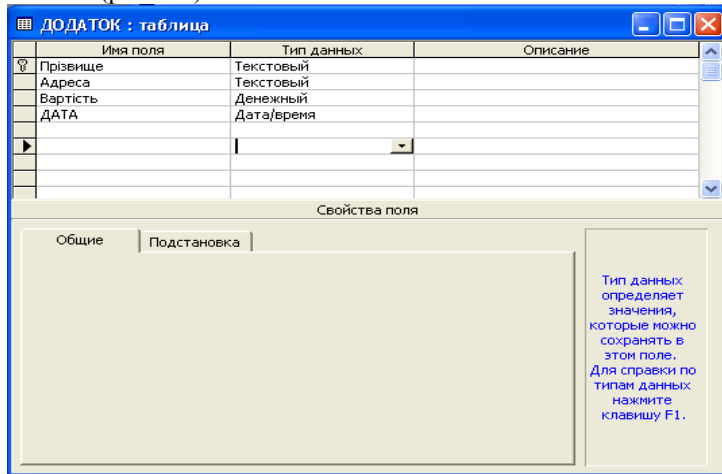


Рис. 5.4. Поле Прізвище визначене як ключове.

```
Function SumE(A, n) As Single
```

```
Dim c As Single
```

```
Dim i As Byte
```

```
c = 0
```

```
For i = 1 To n
```

```
c = c + A(i)
```

```
Next i
```

```
SumE = c
```

```
End Function
```

' Обчислення середнього значення елементів масиву

```
Function CrZn(A, n) As Single
```

```
CrZn = SumE(A, n) / n
```

```
End Function
```

```
Function LogN(x, n) As Single
```

```
Dim i As Byte
```

```
LogN = Log(x) / Log(n)
```

```
End Function
```

```
Sub Zavd_3()
```

```
Dim n, nr, i As Integer
```

```
Dim Sum, Czn, Max, Min, z As Single
```

```
Dim A(20) As Single
```

' Введення вхідних даних

```
nr = 11 ' номер рядка розрахунку
```

```
x = 0.68 * Cells(nr, 4).Value: Cells(nr, 5).Value = "x=":
```

```
Cells(nr, 6).Value = x: n = Cells(nr, 8).Value
```

' Виведення позначення елементів масиву

```
nr = nr + 1: Cells(nr, 3).Value = "Результат розрахунку"
```

```
nr = nr + 1: Cells(nr, 3).Value = "i": Cells(nr, 4).Value = "Ai":
```

' Обчислення та виведення елементів масиву

```
For i = 1 To n
```

```
A(i) = Fn_Ai(x, i)
```

```
Cells(nr + i, 3).Value = i: Cells(nr + i, 4).Value = A(i)
```

```
Next i
```

' Обчислення показників масиву

```
Sum = SumE(A, n): Czn = CrZn(A, n):
```

```
Max = MaxE(A, n): Min = MinE(A, n)
```

' Виведення показників масиву

```
nr = nr + n + 1
```

```
Cells(nr, 3).Value = "S=": Cells(nr, 4).Value = Sum
```

```
Cells(nr - 6, 6).Value = "C=": Cells(nr - 6, 7).Value = Czn
```

```
Cells(nr - 5, 6).Value = "M=": Cells(nr - 5, 7).Value = Max
```

```
Cells(nr - 4, 6).Value = "m=": Cells(nr - 4, 7).Value = Min
```

' Обчислення та виведення значення математичного виразу

```
z = Czn / (Max * Min) * Sum - LogN(Abs(Max + Min), 2) ^ 3
```

CTUN = Sum
End Function

4. Зразок виконання завдання 2

Задано дійсне число $x=0.68$ і ціле число $n=10$. Розробити програмний код для обчислення значень елементів одновимірного масиву

$$\tilde{A} = \left\{ a_i = \frac{i^2 + 5 \cdot \cos i}{\sqrt[3]{x^2 + 2.3i}}, i = \overline{1, n} \right\}$$

і обчислити:

- середнє значення елементів масиву;
- мінімальне та максимальне значення елементів масиву;
- значення арифметичного виразу

$$z = \frac{C}{M \cdot m} \sum_{i=1}^n a_i - \log_2^3 |M + m|,$$

де C, M, m – середнє, максимальне і мінімальне значення елементів масиву.

Програмні коди:

' Обчислення значень елементів масиву {A}

Function Fn_Ai(x, i) As Single

Dim a1, a2 As Single

a1 = i ^ 2 + 5 * Cos(i)

a2 = x ^ (2 / 3) + 2.3 * i

Fn_Ai = a1 / a2

End Function

' Обчислення max значення елементів масиву

Function MaxE(A, n) As Single

Dim m As Single

Dim i As Byte

m = A(1)

For i = 2 To n

If m < A(i) Then m = A(i)

Next i

MaxE = m

End Function

' Обчислення min значення елементів масиву

Function MinE(A, n) As Single

Dim m As Single

Dim i As Byte

m = A(1)

For i = 2 To n

If m > A(i) Then m = A(i)

Next i

MinE = m

End Function

' Обчислення суми елементів масиву

Для пришвидшення сортування і пошуку записів використовуються індекси. Індекси, так само як і ключі, можуть бути простими і складними. В одній таблиці може бути багато індексів, але їх велика кількість сповільнює роботу. Для ключового поля автоматично будується індекс.

Для створення простого індексу потрібно вибрати поле і встановити властивість *Індексне поле* в одне з двох значень: *Ні* (співпадіння не допускаються) або *Так* (співпадіння допускаються). У результаті отримаємо індекс з назвою, що співпадає з назвою поля.

Для створення і редагування індексів потрібно вибрати пункт меню *Вид* далі *Індекси* або натиснути на кнопку *Індекси* панелі інструментів. На екрані відкриється діалогове вікно (рис.5.5), в якому із списку полів треба вибрати поле, для якого будемо будувати індекс і якщо потрібно, то задати порядок сортування і властивості індексу.

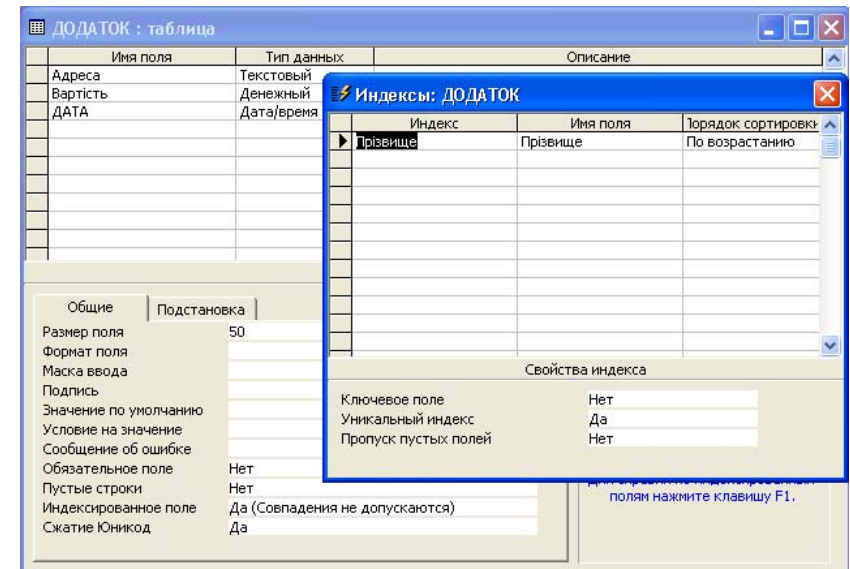


Рис. 5.5. Вікно для створення індексного поля.

Назва складового індексу задається тільки для першого індексу і тільки в цьому полі можливі зміни властивостей індексу. Порядок сортування можна задавати для кожного поля окремо.

5.2.3. Режим таблиць

Режим таблиць – це один із способів створення таблиць в Access. Для того, щоб створити таблицю, потрібно у вікні “Створення таблиці” вибрати

пункт *Режим таблиці*. На екрані відкриється вікно, що буде мати наступний вигляд (рис. 5.6).

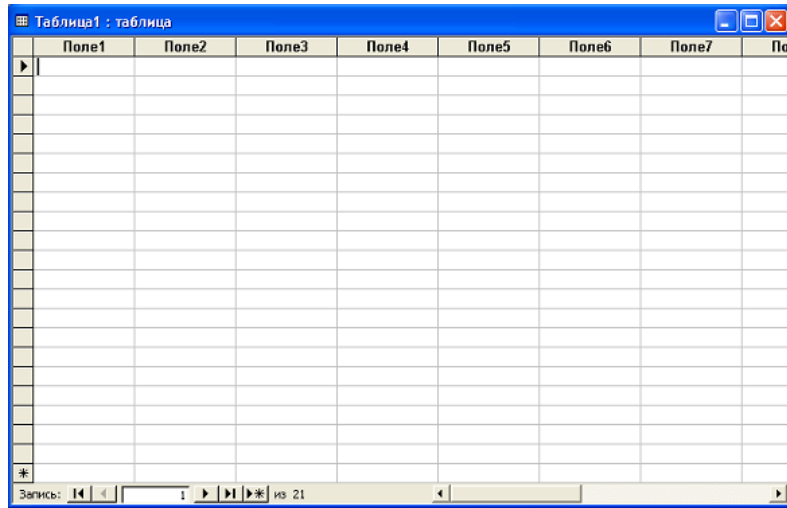


Рис. 5.6. Початковий вигляд вікна в режимі таблиці.

Після відкриття у цьому вікні буде 10 стовпців і 21 запис. Назви стовпців мають назви *поле1*, *поле2* і т.д. Кожне з цих полів можна перейменувати. Для перейменування стовпця потрібно натиснути праву кнопку мишки на його заголовку. На екрані відкриється контекстне меню з якого потрібно вибрати пункт перейменувати стовпець. У цю ж таблицю можна вносити дані, нові записи будуть додаватися автоматично. Нове поле можна додати через контекстне меню. Під час запису в таблицю Access проаналізує інформацію і створить відповідний тип даних. Стовпці, в які не введено інформацію, до таблиці не ввійдуть.

Для подальшої роботи створимо базу даних **kadry**, в основу якої покладемо таблицю *Основні дані*, що містить наступні поля:

- Табельний номер-тип даних *Числової*;
- Прізвище – тип даних *Текстовий*;
- Ім'я- тип даних *Текстовий*;
- По-батькові - тип даних *Текстовий*;
- Стать - тип даних *Текстовий*;
- Дата народження - тип даних *Дата/время*;
- Сімейний стан - тип даних *Текстовий*;
- Кількість дітей - тип даних *Числової*;
- Дата прийняття на роботу - тип даних *Дата/время*;
- Посадовий оклад - тип даних *Денежний*;

DO

<послідовність операторів>
LOOP WHILE <логічний вираз>

DO

<послідовність операторів>
LOOP UNTIL <логічний вираз>

Дія операторів. У випадку використання оператора DO...LOOP WHILE спочатку виконуються оператори, які розташовані між ключовими словами DO та LOOP WHILE. Після цього аналізується значення логічного виразу <логічний вираз>. Якщо цей вираз приймає значення TRUE, тоді знову повторюється виконання операторів тіла циклу (операторів, які містить <послідовність операторів>). У іншому випадку виконуються наступні оператори програмного коду.

У випадку використання оператора DO...LOOP UNTIL <послідовність операторів> виконується у циклі поки, у результаті аналізу, <логічний вираз> приймає значення FALSE.

Приклад 1. Використовуючи команди DO...LOOP WHILE та DO...LOOP UNTIL розробити функції користувача для обчислення суми N перших членів арифметичної прогресії перший член якої дорівнює a_1 , а різниця d.

Програмні коди функцій користувача:

```
Public Function CTPW(N As Integer, d As Single, a1 As Single) As Single
Dim I As Integer
Dim Sum As Single
Dim aI As Single
Sum = 0: I = 1
Do
aI = a1 + (I - 1) * d
Sum = Sum + aI
I = I + 1
Loop While I <= N
CT = Sum
End Function
```

```
Public Function CTPU(N As Integer, d As Single, a1 As Single) As Single
Dim I As Integer
Dim Sum As Single
Dim aI As Single
Sum = 0: I = 1
Do
aI = a1 + (I - 1) * d
Sum = Sum + aI
I = I + 1
Loop Until I > N
```


LOOP

Ключове (зарезервоване) слово LOOP вказує на закінчення виконання команд тіла (області дії) циклу в обох випадках і позначає місце, із якого VBA повертається на початок циклу для аналізу значення логічного виразу.

Дія оператора. У випадку використання оператора DO WHILE...LOOP послідовність команд виконується у циклі, поки значення логічного виразу є TRUE. Це значення логічного виразу є умовою продовження циклу.

У випадку використання оператора DO UNTIL...LOOP послідовність команд виконується у циклі, поки значення логічного виразу є FALSE. Це значення логічного виразу є умовою продовження циклу.

Приклад 1. Використовуючи оператори DO WHILE...LOOP та DO UNTIL...LOOP розробити функції користувача для обчислення суми N перших членів арифметичної прогресії перший член якої дорівнює a_1 , а різниця d.

Програмні коди функцій користувача:

```
Public Function CT(N As Integer,
d As Single, a1 As Single) As Single
Dim I As Integer
Dim Sum As Single
Dim aI As Single
Sum = 0: I = 1
Do While I <= N
aI = a1 + (I - 1) * d
Sum = Sum + aI
I = I + 1
Loop
CT = Sum
End Function
```

```
Public Function CTUN(N As Integer, d As Single, a1 As Single) As Single
Dim I As Integer
Dim Sum As Single
Dim aI As Single
Sum = 0: I = 1
Do Until I > N
aI = a1 + (I - 1) * d
Sum = Sum + aI
I = I + 1
Loop
CTUN = Sum
End Function
```

3. Цикли за післяумовою

У VBA цикли за післяумовою мають формати:

- Підрозділ - тип даних *Текстовий*;
- Примітка - тип даних *Мемо*;
- Фото - тип даних *Поле объекта OLE*.

5.2.4. Введення зовнішніх таблиць

Ще одним способом створення таблиць є імпорт і приєднання таблиць. За допомогою цих команд можна імпортувати або приєднати таблицю з іншого файлу бази даних, лист Excel та інші файли.

Під час виконання імпорту таблиць таблиця фізично поміщається у файл активної бази даних (рис. 5.7). Для здійснення імпорту таблиць потрібно в меню *Файл* вибрати пункт *Зовнішні дані, Імпорт*. Початкова таблиця залишається незмінною. У вікні бази даних з'явиться ім'я імпортованої таблиці.

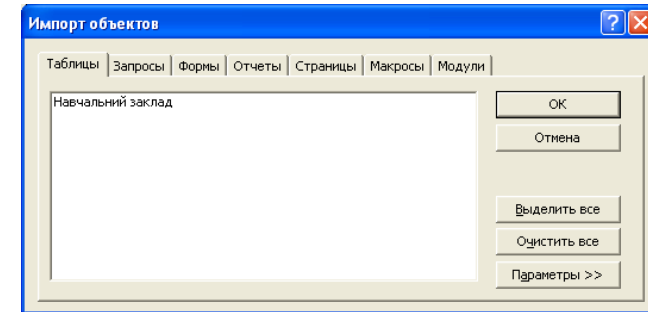


Рис. 5.7. Вікно імпорту таблиці *Навчальний заклад* у базу даних.

Приєднання здійснюється так сама як і імпорт, але замість пункту *Імпорт* потрібно вибрати пункт *Зв'язок з таблицями*. Під час приєднання таблиці в файлі відкритої БД буде знаходитися тільки посилання. Змінювати структуру приєднаної таблиці за допомогою *Конструктора* не можна. У вікні бази даних приєднана таблиця зліва від назви буде мати стрілку (рис. 5.8).

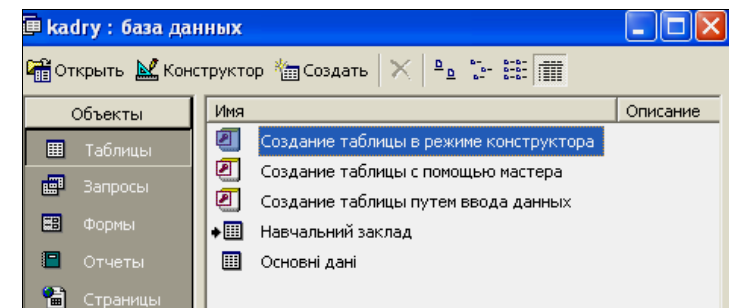


Рис. 5.8. Приєднана таблиця *Навчальний заклад* до бази *kadry*.

5.2.5. Міжтабличні зв'язки

Після розбиття даних на таблиці, а також виділення ключових полів потрібно задати інформацію про те, як зв'язані таблиці між собою, для того, щоб забезпечити логічну цілісність бази даних. Зв'язок між таблицями дає можливість виключити викидання або зміну даних у ключовому полі головної таблиці, якщо з цим полем зв'язані довільні поля іншої таблиці, а також під час викидання або зміни даних головної таблиці здійснити автоматично викидання або зміну відповідних даних у полях зв'язаних таблиць. Одна з таблиць вважається головною, а друга зв'язаною. Головна таблиця це та, яка бере участь у зв'язках своїм ключовим полем. Зв'язки між таблицями можуть бути трьох видів: один-до-багатьох, один-до-одного, багато-до-багатьох. У MS Access для кожного виду зв'язків існує три способи об'єднання. Вибраний тип об'єднання визначає результат вибірки даних із зв'язаних таблиць.

Внутрішнє об'єднання (*Inner Join*). У цьому випадку в результуючу таблицю входять тільки ті записи, зв'язані поля яких співпадають. Ця операція повністю співпадає з операцією природного з'єднання реляційної алгебри.

Ліве зовнішнє з'єднання (*Left Join*). З'єднуються всі записи таблиці з боку "один" і тільки ті записи з боку "багатьох" значення зв'язаного поля яких співпадають із значенням відповідного поля першої таблиці. Тобто під час лівому з'єднанні створюються кортежі внутрішнього з'єднання і додаються до всіх рядки лівого операнда, для яких не має відповідних рядків в правому операнді. Поля кортежів результуючої таблиці будуть приймати нульові значення.

Праве зовнішнє з'єднання (*Right join*). Виконується аналогічно до лівого, але до результатів внутрішнього з'єднання додаються рядки для рядків правого операнда, для яких не було відповідного рядка в лівому операнді.

Для того, щоб встановити зв'язки між таблицями, потрібно закрити всі відкриті таблиці і в меню *Сервіс* вибрати пункт *Схема даних*. На екрані відкриється вікно *Схема даних*. У це вікно потрібно додати таблиці, між якими ми хочемо встановити зв'язок.

Додавання таблиць здійснюється за допомогою меню зв'язки – додати таблицю або відповідну кнопку панелі інструментів. У вікні, що відкриється потрібно два рази клацнути на назвах таблиць, які хочемо зв'язати. Вікно додавання таблиць потрібно закрити. Далі для зв'язку полів потрібно перетягнути ключове поле однієї таблиці до поля іншої таблиці. У результаті отримаємо зв'язки між таблицями (рис. 5.9).

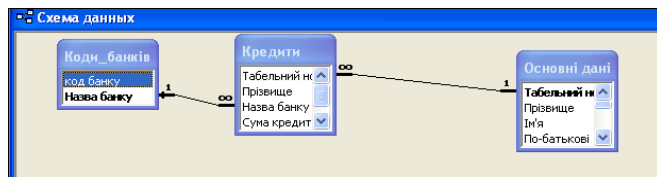


Рис. 5.9. Схема зв'язків між таблицями бази.

т.д. Коли значення параметра стане більшим від значення виразу A2, тоді здійснюється вихід з циклу і виконується наступний після NEXT оператор.

Залежно від версії VBA оператор циклу може функціонувати у режимі з передумовою або післяумовою. Розглянемо дію операторів FOR і NEXT на прикладах.

Приклад 1. Розробити функцію користувача для обчислення суми N перших членів арифметичної прогресії, перший член якої дорівнює a_1 , а різниця d.

Програмний код функції користувача:

```
Public Function CYM(N As Integer, d As Single, a1 As Single) As Single
Dim I As Integer
Dim Sum As Single
Dim aI As Single
Sum = 0
For I = 1 To N Step 1
aI = a1 + (I - 1) * d
Sum = Sum + aI
Next I
CYM = Sum
End Function
```

Приклад 2. Розробити функцію користувача для обчислення добутку N перших членів арифметичної прогресії, перший член якої дорівнює a_1 , а різниця d.

Програмний код функції користувача:

```
Public Function DOB(N As Integer, d As Single, a1 As Single) As Single
Dim I As Integer
Dim DOB1 As Single
Dim aI As Single
DOB1 = 1
For I = 1 To N
aI = a1 + (I - 1) * d
DOB1 = DOB1 * aI
Next I
DOB = DOB1
End Function
```

2. Цикли з передумовою

У VBA цикли з передумовою мають формати:

```
DO WHILE <логічний вираз>
<послідовність операторів>
LOOP
DO UNTIL <логічний вираз>
<послідовність операторів>
```

Розрізняють такі основні види циклів: 1) цикл за лічильником, 2) цикл за вхідним значенням, 3) цикл за вихідним значенням (ітераційний цикл), 4) цикл з накопиченням.

Розглянемо будову циклу. Циклові передує етап *підготовки циклу*. Тут задають початкові значення змінних, які є у циклі. Ця частина виконується один раз. Довільний цикл складається з таких етапів:

1. **Перевірка умови циклу.** Перевіряється умова, яка забезпечує вихід з циклу після досягнення мети. Якщо умова складена неправильно, тоді можна ніколи не вийти з циклу. Така ситуація називається зациклованням і її потрібно уникати.

2. **Виконання послідовності команд.** Виконується послідовність команд (основні дії) заради яких конструювали цикл.

3. **Зміна значення параметра.** Згідно з умовою задачі змінюється значення змінної, яка є в умові. Ця змінна називається параметром циклу.

Залежно від того, коли перевіряється умова, тобто яким буде порядок виконання трьох етапів, розрізняють два види циклів: цикл з передумовою (1, 2, 3) та цикл з післяумовою (2, 3, 1). Для циклу з передумовою перевірка умови здійснюється на початку циклу, а для циклу з післяумовою – у кінці циклу.

Різниця між такими циклами полягає у тому, що цикл з післяумовою виконується хоча б один раз, а цикл з передумовою може не виконуватися жодного разу.

1. Оператор циклу за лічильником

Оператор циклу за лічильником (FOR) призначений для організації повторень однотипних команд, якщо їх кількість у циклі є наперед відома. У VBA оператор циклу FOR має формат:

```
FOR <I> = <A1> TO <A2> STEP <A3>
    <послідовність операторів>
NEXT <I>
```

Оператор FOR-TO-STEP утворює заголовок циклу, NEXT – який фіксує звершення виконання тіла циклу (області дії) і змінює значення параметра циклу I на величину A3. Тіло циклу (область дії) – це послідовність команд, які розташовані між операторами FOR та NEXT.

Змінну I називають параметром циклу (керуюча змінна). A1, A2, A3 – арифметичні вирази, змінні або константи. A1 задає початкове значення параметра циклу, A2 – кінцеве, A3 – значення кроку, на яке щоразу змінюємо значення параметра циклу.

Для оператора. Параметрові циклу I присвоюється значення виразу A1. Якщо це значення є меншим або дорівнює значенню A2, тоді виконується послідовність команд, які утворюють тіло циклу. Після цього значення параметра збільшується на A3 і знову порівнюється зі значенням виразу A2 і

Якщо перетягнуто поле, яке не є ключовим, до іншого поля, яке теж не буде ключовим, тоді утворений зв'язок буде не визначений. Він не буде гарантувати цілісності даних. Такі зв'язки на кінцях лінії, що зв'язує дві таблиці не мають символів. Після встановлення зв'язків через меню *Зв'язки, Змінити зв'язки* (або контекстне меню), на екрані відкривається вікно *Зміна зв'язків* (рис. 5.10).

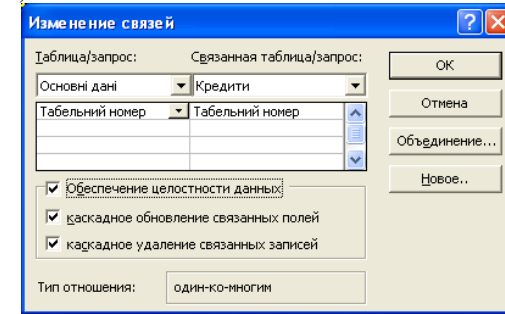


Рис. 5.10. Вікно зміни зв'язків.

У нижній частині цього вікна висвічується інформація про тип зв'язку. Крім того, у цьому вікні розміщені три прапорці, які задають способи за безпечення цілісності даних. Після включення опцій забезпечення цілісності даних стають доступними дві нові опції, які дають можливість ввести ще деякі додаткові обмеження на таблицю. Цілісність даних – це множина правил, яка забезпечує допустимі міжтабличні зв'язки і не дає можливості на випадковій зміні або викиданні зв'язаних даних.

Якщо після включення цих опцій знищимо запис у головній таблиці або змінимо одне із значень ключа то Access автоматично виконає зміни у підпорядкованій таблиці, щоб забезпечити цілісність даних. Для визначення способу з'єднання таблиць на кнопці об'єднання вікна відбувається зміна зв'язку. На екрані відкривається діалогове вікно параметри об'єднань (рис. 5.11), в якому можна вибрати один із способів з'єднань.

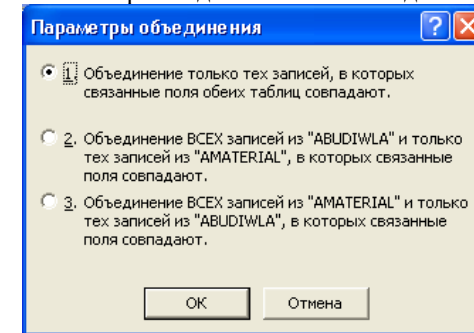


Рис. 5.11. Вікно параметрів об'єднань.

5.3. Запити

Query-by-Example (QBE) — це графічна версія реляційної мови SQL для створення запитів. Ця мова створювалася фірмою IBM під керівництвом (Zluff) Злуфа в 1975 р. та дістала широке розповсюдження і в даний час реалізована в більшості СУБД.

Назва запиту відповідає способу його створення. Під час створення нового запиту в режимі Конструктора відкривається вікно, у верхній половині якого вказується список полів таблиці, які необхідно включити до запиту. Нижня половина вікна – це таблиця QBE, в якій задаються умови для запиту. У рядках умови необхідно задати значення полів, які необхідні для запиту.

У таблиці QBE є такі рядки:

- *Поле* – у цьому рядку розміщуються назви полів. У випадку обчислювального поля, в клітинці розміщується вираз, який після обчислення стає значенням цього поля. Комірки, що знаходяться нижче, належать до цього поля.
- *Сортування* – визначає спосіб впорядкування записів у відповідному полі.
- *Виведення на екран* – визначає, чи буде включатися поле до динамічного набору даних.
- *Умова вибору* – вміщує критерії, згідно з якими записи вибираються у динамічний набір даних.

У Microsoft Access за допомогою мови QBE можна реалізувати такі типи запитів:

- запити на вибірку;
 - запити з обчислювальними полями;
 - запити для впорядкування записів у таблиці;
 - групувальні запити;
 - запити з умовами вибору (параметрами);
 - запити на створення таблиці;
 - перехресні запити;
 - запити вилучення;
 - запити на оновлення записів у таблицях;
 - запити на додавання записів;
 - запити, що здійснюють вибірку записів, що повторюються;
 - запити, що здійснюють вибірку записів, які не мають підпорядкованих;
- Розглянемо докладніше окремі типи запитів.

5.3.1 Запити на вибірку

Запити на вибірку містять формулювання запиту до бази даних, який визначається як набір критеріїв для вибірки з однієї або декількох таблиць. Такі запити є найпоширенішими .

```
End Select
End Function
```

Приклад 2. Розробити функцію користувача для обчислення значення функції

$$y_c = \begin{cases} \text{Петро, якщо } x = 1, 2, 3, 4, 5 \\ \text{Богдан, якщо } x = 6, 7, 8, 9, 10 \end{cases}$$

Програмний код функції користувача:

```
Public Function yc(x As Integer) As String
Select Case x
Case 1, 2, 3, 4, 5
yc = "Петро"
Case 6, 7, 8, 9, 10
yc = "Богдан"
Case Else
yc = "Введіть правильно код"
End Select
End Function
```

Приклад 3. Обчислення коренів квадратного рівняння.

Програмний код функції користувача:

```
Public Function kb_pib(a As Single, b As Single, _
c As Single, k As Integer) As Variant
Dim D As Single
Dim N As Integer
D = b ^ 2 - 4 * a * c
Select Case k
Case 1
N = k
Case Else
N = -k / k
End Select
If D >= 0 Then
kb_pib = (-b + N * Abs(D ^ (1 / 2))) / 2 * a
Else
kb_pib = "Рівняння дійсних коренів не має "
End If
End Function
```

Лабораторна робота 3.

Циклічні алгоритми та їх реалізування у VBA

Циклом називають послідовно повторювану кількість кроків алгоритму. Кількість повторень у циклі повинна бути скінченною.

Приклад. Розробити функцію користувача для обчислення значення функції

$$y = \begin{cases} \sin x, & \text{якщо } x \geq 0 \\ \cos x, & \text{якщо } x \leq 0 \end{cases}$$

Програмний код функції користувача:

```
Public Function y(x As Single) As Single
If x >= 0 Then
y = Sin(x)
Else
y = Cos(x)
End If
End Function
```

2. Оператор розгалуження Select Case

Загальний формат оператора:

```
SELECT CASE <Вираз-селектор>
CASE <список значень 1>
<послідовність операторів 1>
CASE <список значень 2>
<послідовність операторів 2>
CASE ELSE
<послідовність операторів 3>
END SELECT
```

Дія оператора. Якщо значення виразу-селектора збігається з довільним значенням зі списку і, тоді виконується послідовність команд 1.

Списки значень в операторі задаються декількома способами. Вони подані такими прикладами:

```
CASE 4, 5, 6, 7 – цілі значення від 4 до 7
CASE 4 To 7 – дійсні значення від 4 до 7
CASE X>0 – значення більші від нуля
CASE "A" To "Z" – "A", "B", ..., "Z"
```

Приклад 1. Розробити функцію користувача для обчислення значення функції

$$y = \begin{cases} \sin x, & \text{якщо } x \geq 0 \\ \cos x, & \text{якщо } x \leq 0 \\ 3, & \text{в інших випадках} \end{cases}$$

Програмний код функції користувача:

```
Public Function y(x As Single) As Single
Select Case x
Case -2 To 0
y = Cos(x)
Case 0 To 2
y = Sin(x)
Case Else
y = 3
```

Створити запит можна власноруч, за допомогою *Конструктора* або за допомогою *Майстра*.

Для того, щоб створити новий запит за допомогою *Конструктора*, потрібно у вікні бази даних вибрати об'єкт *Запити*, *Створення запиту в режимі Конструктора*. Відкриється вікно *Запит на вибірку* і діалогове вікно *Додавання таблиці* (рис. 5.12). Мишкою відзначається потрібна таблиця і виконується команди *Додати*, *Закрити*.

Після того, як вибрана таблиця буде додана, до неї можна буде звертатися з запитом. У нижній частині вікна запиту з'явиться таблиця QBE (рис. 5.13). У рядок *Поле* (в кожному стовпці) таблиці QBE потрібно задати назви полів, які необхідні для реалізації запиту, а також ті, котрі бажано вивести на друк для повноти інформації по запиту. Нижче буде задаватися ім'я таблиці, а далі під полем вибираються сортування і умова для вибору записів. Створений запит можна під час необхідності зберегти через пункт меню *Файл*.

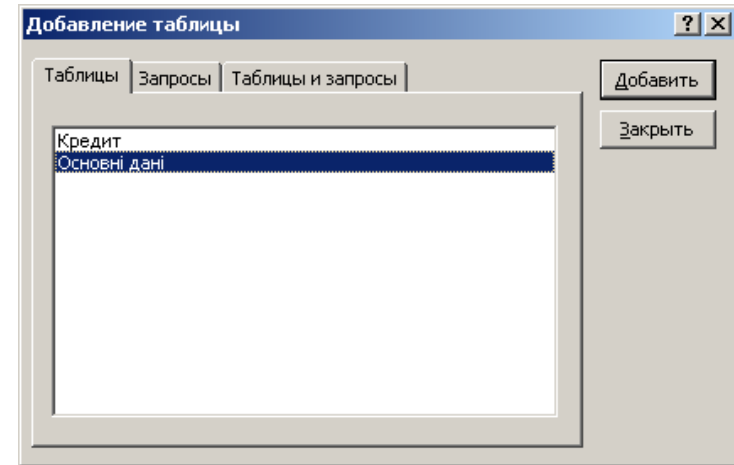


Рис. 5.12. Додавання таблиці *Основные дані*.


Приклад.

На основі таблиці *Основные дані*, що знаходиться у базі даних **kadry**, створити запит для вибірки інформації за статтю працівників.

Для цього у базі даних вибираємо об'єкт *Запити*, *Створення запиту за допомогою Майстра*. В якості вихідного елемента вибираємо таблицю *Основные дані*. Для повноти інформації про працівника додаємо у запит деякі доступні поля (*Табельний номер*, *Прізвище* тощо). Для стовпця *Стать* у полі *Умова відбору* вводимо умову – "жін" (як приклад). Закриваємо запит із збереженням змін (див. рис. 5.13). У результаті отримуємо:

Поле:	Табельний номер	Прізвище	Ім'я	По-батькові	Стать
Имя таблицы:	Основні дані	Основні дані	Основні дані	Основні дані	Основні дані
Сортировка:					
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Условие отбора:					жін"
или:					

Рис. 5.13. Сформована таблиця QBE-запиту на вибірку з таблиці *Основні дані* за статтю "жін".

Запустити запит на виконання піктограмою . У результаті отримаємо наступне (рис. 5.14):

Основні дані Запрос : запрос на выборку					
	Табельний нс	Прізвище	Ім'я	По-батькові	Стать
▶	4	Дідухів	Галина	Іванівна	жін
	5	Залужна	Христина	Семенівна	жін
	10	Пронів	Олена	Василівна	жін
*					

Рис. 5.14. Результат роботи QBE-запиту, відображеного на рис. 4.2.

5.3.2. Запити з обчислювальними полями

У таблицях недоцільно зберігати дані, які можна отримати за допомогою обчислень з інших полів. Для цього можна використовувати запити з обчисленням, де формуються обчислювальні поля. В обчислювальних полях можна здійснювати дії із значеннями різних полів того самого запису або відповідних записів із різних таблиць. У добре спроектованій базі даних зберігаються тільки основні дані, з яких, завдяки обчисленням, можна отримати нові дані. Це дає можливість зберігати місце на диску і кожен раз після виконання запиту отримувати нові результати обчислень.

Приклад.

На основі таблиці *Основні дані* порахувати суму профспілкових внесків з особи кожного місяця, якщо відомо, що кожен член профспілки сплачує 1% від нарахувань.

Для створення такого запиту таблицю *Основні дані* перенесемо у вікно бланку запитів, далі виберемо на сітці QBE потрібне поле *Посадовий оклад*

Щоб додати в таблицю стовпчик потрібно в рядку *Поле* сітки QBE вписати вираз

[Посадовий оклад] * 0,01

Цей стовпчик назвемо "Профвнески". Для цього перед арифметичним виразом потрібно написати назву, після якої поставити дві крапки. Наприклад,

Профвнески: [Посадовий оклад] * 0,01

На сітці QBE цей запит матиме наступний вигляд (рис. 5.15).

(NOT (не), AND (і) та OR (або)). Пріоритет виконання логічних операцій такий: NOT, AND, OR.

Означення 1. Результатом виконання операції (NOT A) є TRUE, якщо значення логічного виразу A є FALSE і навпаки.

Означення 2. Результатом виконання операції (A AND B) є TRUE лише тоді, коли значення обох логічних виразів є TRUE.

Означення 3. Результатом виконання операції (A OR B) є TRUE, якщо значення хоча б одного з логічних виразів є TRUE.

Складений логічний вираз $1 < X < 5$ записують так: $(1 < X) \text{ AND } (X < 5)$.

1. Оператор умовного переходу

Оператор умовного переходу має такий повний формат запису

```
IF <логічний вираз> THEN
  <послідовність операторів 1>
ELSE
  <послідовність операторів 2>
END IF
```

Дія оператора. Якщо, в результаті аналізу, логічний вираз приймає значення TRUE, тоді виконується послідовність операторів 1, якщо, у результаті аналізу, логічний вираз приймає значення FALSE, тоді виконується послідовність операторів 2.

Конструкція IF...THEN, яка займає кілька рядків (відома також як блок IF), має такий формат:

```
IF <логічний вираз> THEN
  <оператор 1>
```

```
...
  <оператор n>
END IF
```

Конструкція IF...THEN...ELSEIF...ELSE. – це конструкція, яку зручно застосовувати для вибору одного з декількох напрямків виконання обчислень.

Конструкції IF...THEN...ELSEIF...ELSE можуть бути однорядковими і блоковими. Блокову конструкцію IF...THEN...ELSEIF...ELSE простіше створити, прочитати і налагодити.

Конструкція IF...THEN...ELSEIF...ELSE має формат:

```
IF <логічний вираз_1> THEN
  <послідовність операторів 1>
ELSEIF <логічний вираз_2> THEN
  <послідовність операторів 2>
ELSEIF <логічний вираз_3> THEN
  <послідовність операторів 3>
ELSE
  <послідовність операторів 4>
END IF
```

$$11) a[x, y, z, b] = \frac{2\cos|x|^{1/3} - x^2/6}{1 - b + \sin^2 y^3} + \ln^2|z|^{0.6}, \quad b[x, y, z] = \frac{x^2 + z^2/tg^2|x|^{0.3}}{3 + x + y^2/2 + z^3/3} + \ln^{0.3}\left|\frac{z}{x}\right|^{1/3};$$

$$12) a[x, y, z, b] = \frac{1 + \sqrt{\sin^2|x + y|^{0.4}}}{2 + b^2 + \sin^2 y^3} + tg \frac{3x}{y}, \quad b[x, y, z] = \cos^2\left(\arctg \frac{x^2 + y}{z + 1}\right) + \frac{x}{y} e^{3x+y};$$

$$13) a[x, y, z, b] = \ln\left(\frac{y - \sqrt{|x^2 - b|}}{z + 4y^2}\right)^{2/3}, \quad b[x, y, z] = 1 - \frac{x + y}{|z|^{0.34}} + \frac{y^2}{3!} + \frac{z^3}{5!} + \frac{e^{x-y}}{z + y};$$

$$14) a[x, y, z, b] = \frac{\sqrt{|x^2 - l|^{0.3}} - \sqrt[3]{|y + 2b|}}{1 + \frac{x^l}{l} + \frac{y^2}{2} + \frac{z^3}{3}}, \quad b[x, y, z] = \frac{|y + x|^{0.2}}{|z|^{1.34}} + \frac{(y - z)^2}{1 + \sin^2 y} + \frac{|z - y|^3}{1 - \cos x^2};$$

$$15) a[x, y, z, b] = \frac{3 + e^{y-x} + \sqrt{|y^2 + b|^{0.3}}}{1 + x^2|y - tg^2 z^2|}, \quad b[x, y, z] = y \left| \frac{|z|^{0.3}}{x} + tg^2 \frac{x + z^2}{2x - 1.4} \right|^{1/3} - ze^{x^2 - y};$$

$$16) a[x, y, z, b] = (1 + y)^2 \frac{|x + y|^{0.3}/b^2 + z}{1 + e^{-(x-z)} + |y|^{0.43}}, \quad b[x, y, z] = x \left(\frac{y + \arctg|x^2 + z|^{0.1}}{2 + \sin^2 y^3} + e^{-\frac{x+z}{y}} \right);$$

$$17) a[x, y, z, b] = y^2 + \frac{z + \sin^2|y + b|^{1.3}}{z^2 + \frac{x^2}{|y + x^3/3|} - \ln|x|}, \quad b[x, y, z] = \frac{1 + \cos^2(y - 2x^3)}{2 + |x|^{3/2} \cdot \sin^2|z|^{0.2}} + \ln^2|z - y|;$$

$$18) a[x, y, z, b] = \frac{2\cos|x^2 - b/6|^{1/3}}{1 + xb + \sin^2 y^3} + \ln^3 z^2, \quad b[x, y, z] = 1 - \frac{x}{l} + \frac{y^2}{3} + \frac{z^3}{5} + \frac{\sin(x - z)^2}{y + z};$$

$$19) a[x, y, z, b] = \frac{1 + \sqrt{\sin^2|x + y|^{0.3}}}{0.3 + b^2 + \sin^2|z|^{1/3}} + tg \frac{3x}{y}, \quad b[x, y, z] = \cos^2\left(\arctg \frac{x^3 + y}{1 + |z|^{0.5}}\right) + e^{-\frac{3z-y}{x}};$$

$$20) a[x, y, z, b] = \ln\left|b^2 - \sqrt[3]{\frac{y^2 - x}{x + 2.1}} \frac{x - y^2}{b + 2z^2}\right|^{2/3}, \quad b[x, y, z] = \ln|z|^{1/3} + \frac{y^2 + \cos^2|x|^{0.3}}{3 + y + x^2/2 + z^3/3};$$

$$21) a[x, y, z, b] = \frac{\sqrt{\cos^2|x^3 + b|^{1.2}}}{3 + x^2 + \sin^2 y^3} + \frac{3 - x}{y + 1}, \quad b[x, y, z] = \left(3x + \sin^2 \frac{|x + z|^{1/3}}{2x + 1.34}\right)^2 - ye^{x^2 - z}.$$

Лабораторна робота 2.

Розгалужені алгоритми та їх реалізування у VBA

Розгалуження – це алгоритмічна конструкція, де аналізується значення логічного виразу (умови) і залежно від його значення виконується та, або інша послідовність операторів.

Логічний вираз може приймати значення TRUE або FALSE. Логічні вирази бувають прості та складені. Простий логічний вираз – це два арифметичні вирази з'єднані символом логічного відношення (=, >, <, >=, <=, <>), а складений – це прості логічні вирази поєднані логічними операціями

Поле:	Прізвище	Ім'я	По-батькові	Посадовий оклад	Професійні [Пос: ▾]
Імя таблиці:	Основні дані	Основні дані	Основні дані	Основні дані	
Сортировка:					
Вивод на екран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Умовне отбора:					
или:					

Рис. 5.15. Сформований QBE-запит на обчислення профспілкових внесків.

Запустити запит на виконання піктограмою .

У результаті запиту отримаємо нову таблицю (рис. 5.16).

Запрос 1 : запрос на выборку					
Объект	Прізвище	Ім'я	По-батькові	Посадовий оклад	Професійні
Табл	Бачинський	Назар	Романович	2 000,00р.	20,00 грн.
Запр	Ваків	Дмитро	Васильович	1 500,00р.	15,00 грн.
Форм	Двигун	Сергій	Борисович	1 800,00р.	18,00 грн.
Отче	Дідухів	Галина	Іванівна	2 000,00р.	20,00 грн.
Стра	Залужна	Христина	Семенівна	3 000,00р.	30,00 грн.
Макр	Кожан	Лев	Маркович	1 200,00р.	12,00 грн.
Моду	Мацігін	Григорій	Петрович	3 000,00р.	30,00 грн.
	Мозговий	Іван	Іванович	3 500,00р.	35,00 грн.
	Павлик	Віктор	Климович	1 500,00р.	15,00 грн.
	Пронів	Олена	Василівна	1 300,00р.	13,00 грн.

Рис. 5.16. Результати, отримані на основі QBE-запиту, зображеного на рис. 3.15.

5.3.3. Запити для впорядкування записів в таблиці

За допомогою запитів записи таблиці можна сортувати за відповідними полями в такій послідовності, яка потрібна користувачу. Для того щоб створити такий запит потрібно у вікні БД вибрати об'єкт *Запит*, далі вибрати *Створення в режимі Конструктора*. У вікні *Додавання таблиці* вибрати таблицю, яку потрібно відсортувати, наприклад *Основні дані*. Впорядкуємо прізвища за алфавітним порядком по зростанню. Для цього в рядку *Поле* таблиці QBE задамо поле *Прізвище*, нижче в рядку *Сортування* виберемо режим сортування *"за зростанням"* в рядку *Вивід на екран* включимо прапорець. В таблиці QBE-запит буде мати наступний вигляд (рис. 5.17).

Поле:	Прізвище	Ім'я	По-батькові
Імя таблиці:	Основні дані	Основні дані	Основні дані
Сортировка:	по возрастанию		
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Условие отбора:			
или:			

Рис. 5.17. QBE –запит на сортування таблиці *Основні дані* за полем *Прізвище* в режимі *"за зростанням"*.

Виконання цього запити виводить прізвище (в алфавітному порядку) ім'я та по-батькові працівників.

5.3.4. Групувальні запити

Групування використовуються для об'єднання записів, які мають однакові значення в полях, що групуються в один запис. Створювати запит можна також вказуючи поля для групування записів і виконання відповідних обчислень. Такі запити можна створити як з однієї, так і декількох таблиць, що зв'язані між собою.

У Access можна використовувати наступні групувальні функції:

- SUMM** – обчислення суми значень поля;
- AVG** – обчислення середнього арифметичного значення поля;
- MIN** – пошук мінімального значення поля;
- MAX** – пошук максимального значення поля;
- COUNT** – підрахунок кількості значень у полі;
- StDev** – обчислення середньо квадратичного відхилення для значень поля;
- Var** – обчислення дисперсії значень поля;
- First** – вибір значення з першого запису;
- Last** – вибір значення з останнього запису, для встановлення поточного

порядку сортування.

Наприклад, потрібно створити запит для підрахунку кількості чоловіків у таблиці *Основні дані*. Для того, щоб створити такий запит, потрібно у вікні бази даних вибрати об'єкт *Запит*, далі вибрати *Створення в режимі Конструктора*. У вікні *Додавання таблиці* вибрати таблицю, яку потрібно відсортувати, тобто *Основні дані*. У нижній частині QBE-таблиці у рядку *Поле* виберемо *Стать*. Після цього в меню *Вид* виберемо пункт *Групова операція*. У бланку запитів QBE появиться рядок *Групова операція* проти кожного поля. Далі в стовпці того поля, за яким буде здійснюватися групування (у нашому випадку це поле *Стать*), потрібно натиснути на кнопку «▼» – групова операція. Після розкриття списку на екрані буде видано перелік функцій, за допомогою яких можна виконувати групувальні операції.. Вибрати потрібну операцію. У нашому випадку візьмемо функцію COUNT. У рядку *Умова відбору* потрібно задати умову відбору, тобто

[Стать]=чол

Лапки можна не ставити, їх впише Access. Таблиця QBE з цим запитом буде мати такий вигляд (рис. 5.18).

Поле:	Стать	
Имя таблицы:	Основні дані	
Групповая операция:	Count	
Сортировка:		
Вывод на экран:	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Условие отбора:	[Стать]="чол"	
или:		

Рис. 5.18. QBE-запит на підрахунок кількості чоловіків в таблиці *Основні дані*.

РОЗДІЛ 8. ЛАБОРАТОРНИЙ ПРАКТИКУМ

8.1. Лабораторний практикум з VBA

Лабораторна робота 1.

Обчислення арифметичних виразів

1. Зразок виконання індивідуального завдання

Задано $x=2, y=4, z=-3$. Розробити програмний код для обчислення значень функцій $a[x,y,z,b]$ і $b[x,y,z]$:

$$a[x,y,z,b] = (x+y)^2 \cdot \frac{(x+y^2) \cdot |b^2+z|^{0.3}}{e^{z-2} + y^2} \quad \text{і} \quad b[x,y,z] = y \cdot \left(\frac{x^2 + \text{tg}(y+z)^2}{0.345y + \sin^2 x^2} + e^{-\frac{x+z}{z}} \right).$$

Програмний код функції користувача:

```
Function Fn_b(x, y, z) As Single
Dim f1, f2, f3 As Single
f1 =x ^ 2 + Tan(y + z) ^ 2
f2 =0.345 * y + Sin(x ^ 2) ^ 2
f3 =Exp(-(x + y) / z)
Fn_b =y * (f1 / f2 + f3)
End Function

Function Fn_a(x, y, z, b) As Single
Dim f1, f2, f3 As Single
f1 =(x + y) ^ 2
f2 =(x + y ^ 2) * Abs(b ^ 2 + z) ^ 0.3
f3 =Exp(z - 2) + y ^ 2
Fn_a =f1 * f2 / f3
End Function
```

2. Індивідуальні завдання

- 7) $a[x,y,z,b] = \frac{\sqrt{|x^2-z|^{0.3}} - \sqrt[3]{|y+2b|}}{1 + \frac{x^2}{1} + \frac{y^2}{2} + \frac{z^3}{3}}$, $b[x,y,z] = x \left(\frac{y + \arctg|x^2+z|^{0.1}}{3 + \sin^2 y^3} + e^{-\frac{x+z}{y}} \right)$;
- 8) $a[x,y,z,b] = |x|^{0.43} + \frac{e^{y-x} + \sqrt{|y^2+b|^{0.22}}}{1 + x^2|y-tg^2z|}$, $b[x,y,z] = \frac{2z + \cos|y-3x|^{1/3}}{2.1 + \sin^2|z^3-y|^{0.2}} + \ln^2|z-x|$;
- 9) $a[x,y,z,b] = (x+y)^2 + \frac{x+z^3/(b^2+y)^2}{1 + e^{-(x-y)} + |z|^{0.34}}$, $b[x,y,z] = 1 + \frac{|y-x|^2}{|z|^{1.34}} + \frac{(z-x)^2}{\sin^2 y} + \frac{|y-z|^3}{\cos y^2}$;
- 10) $a[x,y,z,b] = y^2 + \frac{x^2 + \sin^2 b}{y^2 + \frac{x^2}{y+x^3/3} - \ln|x|}$, $b[x,y,z] = \frac{1}{|z|^{0.6}} + \sin^2 \frac{x+z^2}{2x-y} - ze^{-\frac{x-y}{z}}$;

звукового супроводу. У випадку, коли loop=1 – відбувається одноразове відтворення звукового фрагменту, наприклад, привітання або коментар.

Значення атрибуту volume=число задає гучність звучання, наприклад:

```
<BGSOUND src="Шлях до файлу звуку" loop=1 volume=4>
```

```
<MARQUEE> </marquee>
```

елемент, призначений для створення рухомого рядка. Використовуючи цей елемент, Вам необхідно звернути увагу на вдалиий підбір значень атрибутів.

Колір фону для рухомого рядка задається значенням атрибуту bgcolor="колір"

Якщо колір фону (тла) вибраний, тоді рухомий рядок виглядає як кольорова смуга, вздовж якої з певною швидкістю рухається текст. Ширину смуги рухомого рядка задають значенням атрибуту

```
height=ширина у пікселях
```

```
height=ширина у %
```

Якщо ширина смуги задається у пікселях, тоді її варто вибирати з діапазону значень 30 ... 50. Значення атрибуту height, яке задане у відсотках означає, що це відсоток від висоти відтворюваної частини Web-сторінки у вікні броузера (однієї екранної сторінки). Для вирівнювання тексту рухомого рядка до верхнього краю, за шириною і до нижнього краю використовується атрибут:

```
align="top"
```

```
align="middle"
```

```
align="bottom"
```

Наприклад:

```
<MARQUEE bgcolor="orange" align="middle">
```

```
Текст у рухомому рядку</marquee>
```

У результаті виконання запити на екрані отримаємо (рис. 5.19).



Рис. 5.19. Порахована кількість чоловіків на підставі QBE-запиту, що зображений на рис. 5.18

5.3.5. Запити з умовами вибору (параметрами)

У розглянутих запитах для умов відбору використовувались константи, але замість констант у виразах умов відбору можна використовувати змінні (параметри). Параметри – це довільні текстові рядки в квадратних дужках. Для того, щоб встановити умову вибірки, застосовують вирази, в яких можна використовувати операції порівняння ">", "<", ">=", "<=", "=", логічні операції AND, OR, а також службові слова, розглянуті в SQL.

Таблиця 5.1

Оператори порівняння, логічні оператори та спеціальні знаки

Оператори	Значення оператора	Приклад	Значення прикладу
Оператори порівняння			
=	Рівність	=Бойчук	Тільки ім'я Бойчук
>	Більше ніж	>5000	
<	Менше ніж	<05.04.2003	Менше ніж (більше ніж) 5 квітня 2003
>=	Більше рівне	>=М	Більше або рівно літері М
<=	Менше рівне	<=05.04.2003	Більше або рівно 5 квітня 2003 року
<>	Не рівно	<>UA	Не рівно PL
Between	Між двома значеннями	Between 15 and 25	Число від 15 до 25
In	Список значень	In(UA, GB, USA)	UA, GB, USA
Is Null	Поле порожнє	Is Null	Запис, який не має значення в полі
Is Not Null	Поле не є порожнє	Is Not Null	Запис, який має значення в полі
Like	Описує зразок	Like c*	Запис починається з букви "с"
Логічні оператори			
And	Обидва значення справджуються	>=1 And <=10	Між 1 і 10
Or	Одне зі значень справджується	UA or USA	UA або USA
Not	Неправда	Not Like ma???	Записи, які не розпочинаються літерами „ma” з трьома іншими знаками

Продовження таблиці 5.1

Спеціальні знаки			
?	Одинарний знак	8?-791	Номер телефону з будь яким другим знаком
*	Знаки	(8245)*	Текст, що починається з (8245). Наприклад, номер факсу
[назва поля]	Інші поля з запиту	<[TYZN_PLATA]	Записи, в яких значення поля є менші від значення в полі „TYZN_PLATA”

Наприклад, для того, щоб вибрати співробітників *Посадовий оклад* яких більший ніж 1500 і менший ніж 2000, інтервал відбору можна записати у такому вигляді:
BETWEEN 1500 AND 2000

Таблиця QBE буде мати такий вигляд (рис. 5.20).

Поле:	Прізвище	Посадовий оклад	
Имя таблицы:	Основні дані	Основні дані	
Сортировка:			
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Условие отбора:		Between 1500 And 200	
или:			

Рис. 5.20. QBE-запит на отримання прізвищ співробітників, зарплата яких знаходиться в межах 1500 та 2000 грн.

Якщо параметри нижньої та верхньої межі потрібно зробити змінними, тоді оператор BETWEEN можна записати таким чином:

BETWEEN [нижня межа] AND [верхня межа]

Під час запуску на виконання запиту, що містить параметри, на екран видається діалогове вікно, в яке потрібно ввести значення параметрів (рис. 5.21).

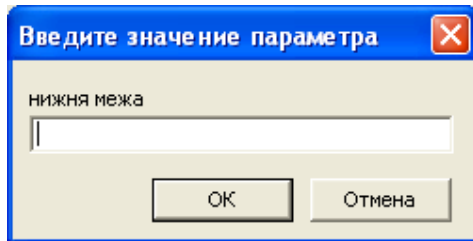


Рис. 5.21. Вікно для введення параметрів у запиті.

Введені у вікно значення параметра підставляються в умову відбору. Запити такого типу мають переваги, тому, що зручніше задавати

Якщо клацнути лівою кlawішею "мишки" на написові "Вміст файлу правої частини", тоді браузер завантажить файл framefile_3.htm, так як цей запис є гіперпосиланням на файл framefile_3.htm. HTML-код цього файлу (і решти, які реалізують гіперпосилання у відповідних частинах) є простим, щоб легше було проаналізувати, власне, саму організаційну структуру зв'язків.

На рис. 7.14 поданий вигляд вікна браузера після того, як клацнули лівою кlawішею "мишки" на написові "Вміст файлу правої частини".

Зверніть увагу, що у рядку заголовку на рис. 5.13, 5.14 відтворюється заголовок з файлу Main_file.htm. Це єдина інформація з цього файлу, яку можна побачити у вікні браузера, а усі дані елементів TITLE інших файлів не будуть відтворюватися.

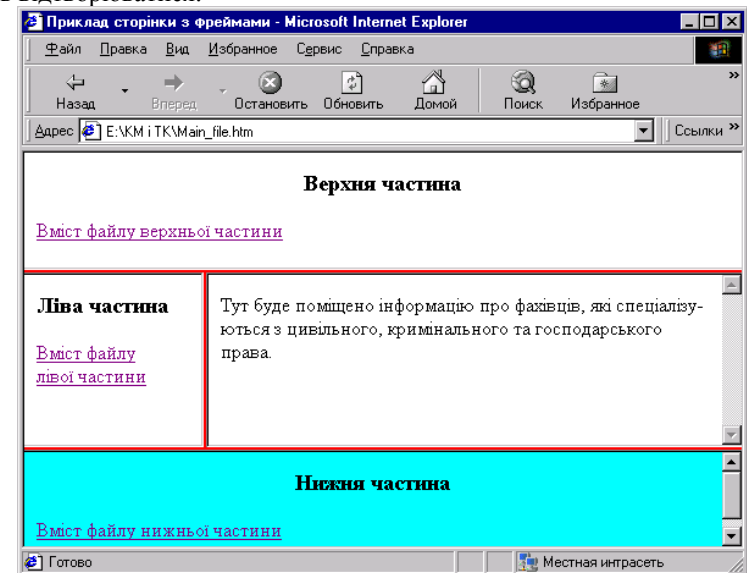


Рис. 7.14. Вигляд вікна браузера після реалізування гіперпосилання у правому фреймі

7.10. Відтворення звуку і рухомих об'єктів

<BGSOUND>

елемент, призначений для створення звукових ефектів. Елемент BGSOUND записується як вкладений у елемент HEAD. Для елемента BGSOUND необхідним атрибутом є атрибут src, значення якого вказує місце на магнітних носіях, де зберігається звуковий файл.

Значення атрибуту loop задає кількість повторень звукового фрагменту.

Значення атрибуту loop=-1 або loop="infinite", як правило, використовують у тому випадку, коли потрібним є безперервне відтворення

Тут буде поміщено інформацію про фахівців, які спеціалізуються з цивільного, кримінального та господарського права.

```

</body>
</html>
framefile_4.htm:
<HTML>
<HEAD>
<TITLE> Файл framefale_4</title> </head>
<BODY>

```

Тут буде поміщено адресу електронної пошти фірми та іншу довідкову інформацію.

```

</body>
</html>

```

Створенням цих файлів процес проектування Web-сторінки з фреймовою структурою закінчується.

Подібну структуру мають багато Web-документів. Такі "ланцюжки" сторінок зручно використовувати для створення віртуальних книг, галерей зображень, тобто тоді, коли користувачам потрібно послідовно переглядати ряд сторінок. У цьому випадку важливим є правильно організувати посилання.

На рис. 7.13 поданий вигляд вікна броузера у якому відтворено HTML-код основного файлу (Main_file.htm).

Файл Main_file.htm організовує (здійснює поділ на області вікна броузера) фреймову структуру Web-сторінки, а файли frame_top.htm, frame_left.htm, frame_right.htm, frame_bottom.htm призначені для організації підтримки відповідних фреймів.

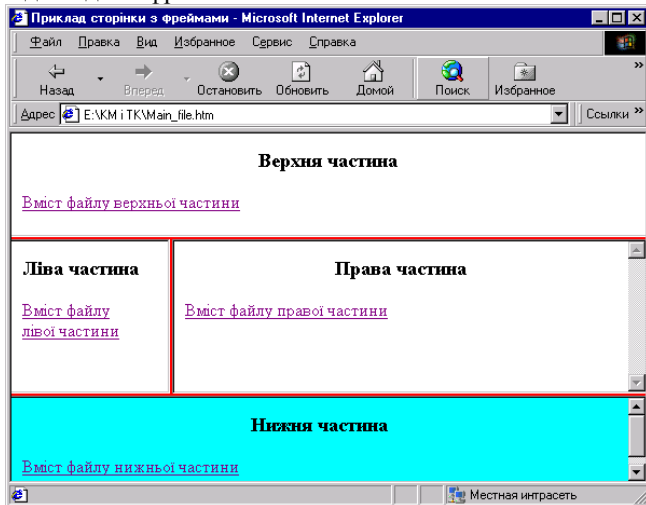


Рис. 7.13. HTML-код основного файлу (Main_file.htm) відтворений у вікні броузера.

значення для умов відбору під час запуску запиту на виконання, а не в таблиці QBE в режимі Конструктора.

Запити з параметрами зручно використовувати в ролі джерела даних для форми. У формі створюється елемент керування, що має назву параметра. Якщо в запиті використовується форма, то діалогове вікно для введення параметра не відображається. Введення параметра здійснюється у відповідному елементі керування форми.

5.3.6 Запити на створення таблиці

Запити на створення таблиці від звичайних запитів на вибірку відрізняються тим, що замість тимчасового зберігання даних утворюють в результаті нову таблицю, в якій зберігаються результати запиту.

Особливим випадком застосування запитів, що створюють таблиці, є запам'ятовування даних, зібраних з різних таблиць в одній допоміжній таблиці. В деяких випадках створення таких тимчасових таблиць є необхідне, коли система Access істотно сповільнює швидкість роботи під час виконання запитів, побудованих на запитах, для яких джерелами є об'єкти запитів. У новій таблиці тип даних, розмір і поля унаслідуються з полів-джерел, інші атрибути приймають стандартні установки.

Для того, щоб створити запит на створення нової таблиці виконаємо наступні дії.

Нову таблицю будемо створювати (для прикладу) на основі двох таблиць. Тому, створимо за допомогою Конструктора ще одну таблицю у тій же базі **kadry** і назвемо її *Кредит*. У цій таблиці *Кредит* створимо тільки поля: *Табельний номер*, *Прізвище*, *Назва банку*, *Сума кредиту*, *Дата завершення кредиту*. Зв'язок між таблицями *Основні дані* та *Кредит* здійснимо за полем *Табельний номер*.

Виберемо таблиці *Основні дані* та *Кредит*, а в них поля *Табельний номер*, *Прізвище*, *Ім'я*, *По-батькові* та *Сума кредиту*. В таблиці QBE цей запит буде мати наступний вигляд (рис. 5.22).

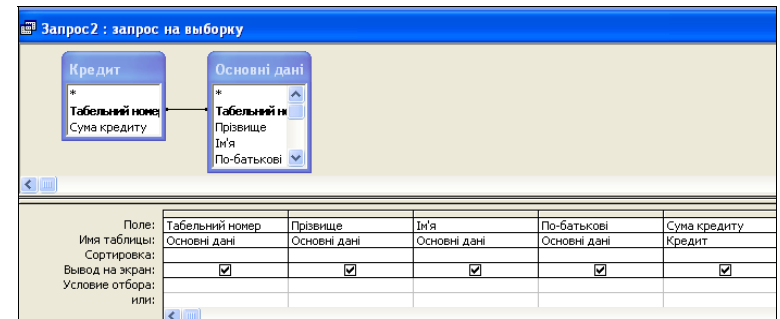


Рис. 5.22. QBE-запит на створення нової таблиці на основі полів з таблиць *Основні дані* та *Кредит*.

Щоб на основі запити утворилась нова таблиця в базі **kadry** потрібно скористатися піктограмою *Тип запиту* на панелі інструментів і серед пунктів випадаючого меню вибрати *Створення таблиці* (рис. 5.23).

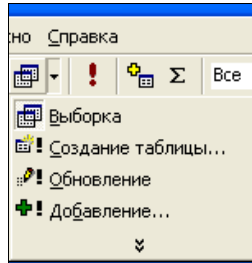


Рис. 5.23. Вигляд випадаючого меню піктограми *Тип запиту*.

Після вибору пункту меню *Створення таблиці* відкриється вікно Створення таблиці (рис. 5.24), де можна надати новій таблиці ім'я і вибрати потрібну базу для збереження.

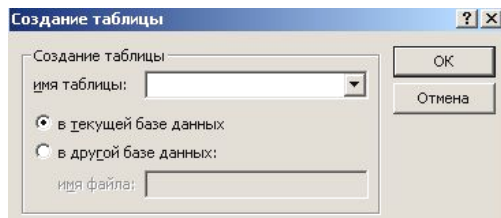


Рис. 5.24. Завершення створення нової таблиці.

Після введення назви отриману таблицю можна використати для створення запитів. Після запуску цього запити на екран буде видано підказку про підтвердження створення нової таблиці. У результаті виконання запити отримаємо наступну таблицю (рис. 5.25).

Табельний нс	Прізвище	Ім'я	По-батькові	Сума кредиту
1	Бачинський	Назар	Романович	2 000,00 грн.
2	Баків	Дмитро	Васильович	1 230,00 грн.
3	Двигун	Сергій	Борисович	7 600,00 грн.
4	Дідухів	Галина	Іванівна	2 300,00 грн.
5	Залужна	Христина	Семенівна	3 000,00 грн.
6	Кожан	Лев	Маркович	2 400,00 грн.
7	Мацігін	Григорій	Петрович	0,00 грн.
8	Мозговий	Іван	Іванович	0,00 грн.
9	Павлик	Віктор	Климович	0,00 грн.
10	Пронів	Олена	Василівна	5 600,00 грн.

Рис. 5.25. Результат виконання QBE-запити поданого на рис. 5.22

```
<HEAD>
<TITLE> Правий фрейм</title> </head>
<BODY>
<H3 align="center"> Права частина </h3>
<A href="framefile_3.htm" target="right">
Вміст файлу правої частини</a>
</body>
</html>
frame_bottom.htm:
<HTML>
<HEAD>
<TITLE> Нижній фрейм</title> </head>
<BODY bgcolor="aqua">
<H3 align="center"> Нижня частина </h3>
<A href="framefile_4.htm" target="bottom">
Вміст файлу нижньої частини</a>
</body>
</html>
```

Для того, щоб така фреймова структура Web-сторінки функціонувала, необхідно створити файли *framefile_1.htm*, *framefile_2.htm*, *framefile_3.htm*, *framefile_4.htm* на які здійснюється посилання. Імовірно, HTML-коди цих файлів можуть бути і такими:

```
framefile_1.htm:
<HTML>
<HEAD>
<TITLE> Файл framefile_1</title> </head>
<BODY>
Тут буде розміщено логотип і назва фірми з надання правових послуг,
юридична адреса, прізвище власника.
</body>
</html>
framefile_2.htm:
<HTML>
<HEAD>
<TITLE> Файл framefile_2</title> </head>
<BODY>
Тут буде розміщено перелік послуг, які надає фірма.
</body>
</html>
framefile_3.htm:
<HTML>
<HEAD>
<TITLE> Файл framefile_3</title> </head>
<BODY>
```

```

<FRAME src="frame_top.htm" name="top"
scrolling="no" frameborder=1 border=4 bordercolor="red">
<FRAMESET cols=25%,75%>
<FRAME src="frame_left.htm" name="left" frameborder=1
border=4 bordercolor="red">
<FRAME src="frame_right.htm" name="right"
scrolling="yes" frameborder=1 border=4 bordercolor="red">
</frameset>
< FRAME src="frame_bottom.htm" name="bottom"
frameborder=1 border=6 bordercolor="red">
</frameset>
</html>

```

Це приклад документу фреймової структури, який ділить вікно броузера на чотири області. Для вертикального поділу середньої області використано вкладену структуру для елемента FRAMESET, а для правої частини обов'язково створюються смуги прокручування.

Зважаючи на те, що у HTML-файлах, які підтримують створені фрейми, є гіперпосилання на файли, у яких міститься логічний зміст цих посилань, тоді у гіперпосиланнях необхідно використати атрибут target. Значенням атрибуту target буде назва фрейму, яка була визначена атрибутом name елементу FRAME основного документу.

Приклад HTML-коду файлу, який підтримує верхній фрейм frame_top.htm буде таким:

```

<HTML>
<HEAD>
<TITLE> Верхній фрейм</title> </head>
<BODY>
<H3 align="center"> Верхня частина </h3>
<A href="framefile_1.htm" target="top">
Вміст файлу верхньої частини </a>
</body>
</html>

```

frame_left.htm:

```

<HTML>
<HEAD>
<TITLE>Лівий фрейм</title> </head>
<BODY>
<H3 align="center"> Ліва частина </h3>
<A href="framefile_2.htm" target="left">
Вміст файлу лівої частини</a>
</body>
</html>

```

frame_right.htm:

```

<HTML>

```

5.3.7. Перехресні запити

Перехресні запити – це один з різновидностей запитів на вибірку. Ці запити призначені для поглибленого аналізу інформації, що зберігаються в таблицях. Результати перехресного запиту групуються за двома наборами даних, один з яких розміщений у лівому стовпці (стовпцях) таблиці, а другий – у верхньому рядку. У решту стовпців таблиці розміщуються результати статистичних розрахунків (Sum, Count тощо), які виконуються над даними.

Приклад.

Створити запит, результатом якого будуть суми кредитів, отриманих у різних банках певними працівниками підприємства.

1. Створимо спочатку таблицю *Коди банків*, у яку введемо тільки два поля: *Код банку* та *Назва банку* (рис. 5.26).

Коди банків : таблиця		
	код банку	Назва банку
+	5	Аваль
+	4	Ексімбанк
+	3	Кредобанк
+	1	Укргазбанк
▶	2	Укрсоцбанк
*		п

Рис 5.26. Таблиця кодів банків.

2. Зв'яжемо таблиці *Основні дані*, *Коди банків* та *Кредити* наступним чином (рис. 5.27).

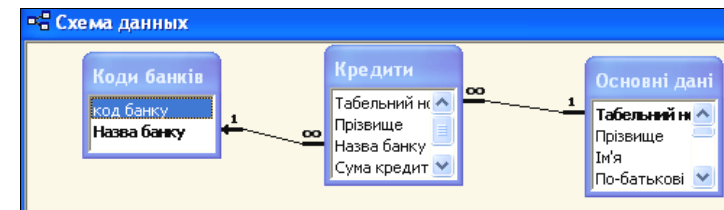



Рис. 5.27. Схема зв'язку необхідних таблиць.

3. Тепер створимо перехресний запит, виконавши наступні дії: вибираємо об'єкт *Запит* бази **kadry**, *Створення запиту в режимі Конструктора, Додавання трьох таблиць*, пункт головного меню *Запит, Перехресний*.

4. Побудуємо наступну QBE-таблицю (рис. 5.28).

Поле:	Прізвище	Назва банку	data: Сума кредит
Имя таблицы:	Основні дані	Коди банків	Кредити
Групповая операция:	Группировка	Группировка	Sum
Перекрестная таблица:	Заголовки строк	Заголовки столбц	Значение
Сортировка:	по возрастанию		
Условие отбора:			
или:			

Рис. 5.28. Таблица QBE перехресного запиту.

5. Запустимо створений запит на виконання за допомогою кнопки  і отримаємо наступний результат (рис. 5.29).

	Прізвище	Аваль	Укргазбанк	Укрсоцбанк
▶	Бачинський	2 000,00 грн.		
	Ваків	1 230,00 грн.	227 442,00 грн.	4 440,00 грн.
	Варіван	60 000,00 грн.		
	Двигун			7 600,00 грн.
	Дідухів	2 300,00 грн.		
	Кожан		3 400,00 грн.	
	Мозговий		55 550,00 грн.	
	Пронів	5 600,00 грн.		

Рис. 5.29. Зведена перехресна таблиця кредитів, отриманих співробітниками у різних банках.

5.3.8. Запити на вилучення


У базі даних може зберігатися велика кількість даних, які в даний момент неактуальні. Наявність таких даних в базі істотно сповільнює роботу системи. Очевидно, що деякі з неактуальних даних потрібно зберігати протягом певного часу. Однак найкраще зберігати такі дані в окремих таблицях або навіть в іншій базі даних. Після архівації потрібно усунути непотрібні дані з поточних таблиць. Найшвидше усунення непотрібних даних можна виконати за допомогою запиту на вилучення, який дає можливість вилучати з таблиці цілі записи на підставі будь-яких заданих критеріїв. Єдине обмеження на вилучення записів пов'язане з порушенням цілісності даних.

Для надійності спочатку створюють запит на вибірку, за допомогою якого переглядають ті запити, які можуть бути вилучені. Якщо в результаті запиту-вибірki присутні лише ті запити, які підлягають вилученню, то такий запит-вибірку можна перетворити в запит-вилучення.

Приклад.

Вилучити записи з прізвищами усіх жінок з таблиці *Основні дані*, що входить до бази **kadry**.

Для цього потрібно виконати наступні дії.

1. Вибираємо об'єкт *Запит* бази **kadry**, далі *Створення запиту в режимі Конструктора*, *Додавання таблиці Основні дані*, пункт головного меню *Запит*, *Вилучення*, або піктограма .

елемент поділу вікна браузера на частини.

Спочатку необхідно визначитися, які області будуть потрібні і як поділити сторінку. Горизонтальний поділ вікна браузера здійснюється значенням атрибуту `rows`, а вертикальний – значенням атрибуту `cols`. Значення атрибутів можна задати у пікселях або у відсотку відношенні.

Розглянемо кілька прикладів:

- `cols=50%,50%` – вертикальний поділ області перегляду навпіл;
- `cols=25%,75%` – у вікні браузера ліва вертикальна область займає 1/4 частку простору, а права – 3/4;
- `rows=150,30%, *` – у вікні браузера верхня горизонтальна область займає 150 пікселів, середня – 30%, а нижня – решту простору сторінки.

Другим етапом є формування окремих HTML-файлів для кожної з областей. Ці файли створюються за такими ж правилами, як і інші гіпертекстові документи. Формуючи ці файли обов'язково треба врахувати розміри областей у вікні браузера, де вони будуть відтворюватися. Поки ці файли не будуть створені, відкривати документи фреймової структури нема змісту, Ви нічого не побачите.

`<FRAME>`

елемент формування посилання на кожний документ вмісту (підтримки) для кожної окремої області. Використовується як вкладений у елемент `FRAMESET`.

Кожний елемент `FRAME` повинен мати ім'я, яке задається значенням атрибуту `name`, наприклад:

`<FRAME src="файл.htm" name="ім'я фрейму">`

Тепер розглянемо атрибути елементу і їх значення.

`Scrolling` – атрибут управління смугами прокручування всередині окремої області. Він може приймати значення `yes` – смуги прокручування створюються обов'язково, `no` – смуги прокручування не створюються, `auto` – смуги прокручування створюються автоматично за необхідністю. Якщо атрибут `scrolling` не використовується – браузер створює смуги прокручування для тих областей, у яких документ не поміщається повністю.

`Frameborder` – атрибут, який створює рамку довкола фрейму. Значення атрибуту, яке дорівнює 1 – створює рамку, а значення 0 – не створює рамки.

`Marginheight` – атрибут, який визначає границі відступу від верхнього і нижнього країв фрейму. Значення атрибуту задається у пікселях.

`Margiwidth` – атрибут, який визначає величину границі відступу від лівого і правого країв фрейму.

Використання елементів `FRAMESET` та `FRAME` розглянемо на такому прикладі:

```
<HTML>
<HEAD>
<TITLE> Приклад сторінки з фреймами </title>
</head>
<FRAMESET rows=20%,60%,20%>
```


HTML допускає використання графічних файлів, які збережені тільки у форматі з розширеннями .gif або .jpeg.

Іноколи, з тих або інших причин, браузер не може відтворити зображення. У HTML для елемента IMG призначений атрибут alt – альтернатива (alternative). Атрибут alt дає змогу виводити рядок тексту на місці графічного зображення, якщо браузер не може відтворити зображення, наприклад:

Це пов'язано ще і з тим, що Web-сторінка може завантажуватися браузером доволі довго і поки графічні файли завантажуються, користувач має змогу побачити, які зображення будуть на сторінці.

Висоту і ширину області Web-сторінки, у якій буде відтворено графічне зображення, визначають атрибути height та width. У тому випадку, коли використаний один з атрибутів, зображення масштабується таким чином, щоб висота або ширина відповідали значенню відповідного атрибуту, а інший розмір встановлюється автоматично зі збереженням пропорцій.

Зауважимо, що змінюючи значення атрибутів height та width можна доволі змінити розміри графічного зображення у вікні браузера, але розміри графічного файлу залишаються без змін. Зміна значень атрибутів height і width ніяк не вплине на час завантаження браузером конкретного графічного зображення.

Якщо між початковим та кінцевим дескрипторами елемента А використано елемент IMG, тоді мова йде про зображення-посилання (clickable image). У цьому випадку гіперпосиланням буде графічне зображення, наприклад:

Гіперпосиланням, у даному прикладі, буде графічне зображення (фотографія) групи zed. Якщо клацнути на цьому зображенні лівою клавішею "мишки", тоді браузер завантажить файл Page_zed у якому, імовірно, буде подана інформація про групу.

7.9. Фрейми

Під поняттям фрейми розуміють області (поля), які створюються у вікні браузера для одночасного відтворення кількох документів. Для створення сторінок з фреймами необхідно створити кілька документів, які відрізняються за функціональним призначенням:


- документи, які визначають фреймову структуру Web-сторінки;
- документи, які призначені для заповнення інформацією кожної з областей.

<FRAMESET> </frameset>

2. Заповнимо QBE-запит (рис. 5.30).

Поле:	Стать	
Имя таблицы:	Основные дані	
Удаление:	Условие	
Условие отбора:	"жін"	
или:		

Рис. 5.30. Заповнений QBE-запит на видалення усіх записів жінок з таблиці *Основные дані*.

3. Запустимо запит на виконання піктограмою  і отримаємо повідомлення (рис. 5.31).

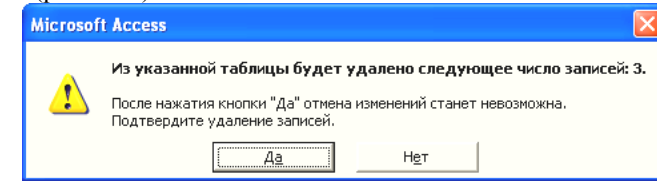


Рис. 5.31. Попередження про можливе видалення з таблиці *Основные дані* записів, які задовільняють певній умові.

4. Якщо дати підтвердження на видалення, тоді вибрані для видалення записи не підлягають відновленню.

5.3.9. Запити на оновлення записів у таблицях


Запити на оновлення записів – це процедури автоматичного перетворення даних у таблиці. Виконання запитів на оновлення спричинює зміни в початкових таблицях. Запити на вибірку таких змін не виконують.

Загальну схему створення запитів на оновлення продемонструємо на прикладі.

Приклад.

На основі таблиці *Основные дані*, співробітникам господарського підрозділу підвищити посадовий оклад на 10%.


Для цього потрібно виконати наступні дії.

1. Вибираємо об'єкт *Запит* бази **kadry**, далі *Створення запиту в режимі Конструктора*, *Додавання таблиці Основные дані*, пункт головного меню *Запит, Оновлення*, або піктограму .

2. Заповнимо QBE-запит (рис. 5.32).

Поле:	Посадовий оклад	Підрозділ
Имя таблицы:	Основные дані	Основные дані
Обновление:	[Посадовий оклад]+[Посадовий оклад]*0,1	
Условие отбора:		"господарський"
или:		

Рис. 5.32. Таблиця QBE запиту на зміну посадових окладів працівникам господарського підрозділу.

3. Запустимо запит на виконання піктограмою  і отримаємо повідомлення (рис. 5.33).

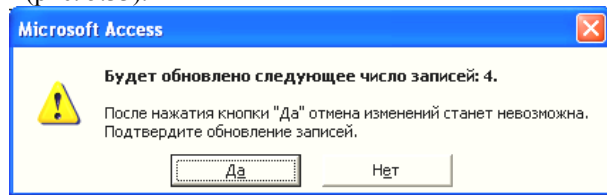


Рис. 5.33. Попередження про можливе оновлення записів таблиці *Основні дані*, що задовольняють певну умову.

4. Якщо дати підтвердження на оновлення, то вибрані для оновлення записи не підлягають відновленню.

5.3.10. Запити на додавання записів


Запити на додавання записів – це такі запити, які доповнюють існуючі таблиці даними що вибрані з інших таблиць або запитів, які додають нові записи. Під час виконання такого типу запитів повинна зберігатись узгодженість типу даних, що додаються з типом полів таблиці, до якої додаються дані. Також повинно зберігатись значення ключа цієї таблиці. Таблиця, в яку додаються записи, може бути як у файлі поточної бази даних, так і у зовнішньому файлі бази даних. Записи додаються в кінець таблиці.

Загальну схему створення запитів на додавання записів продемонструємо на прикладі.

Приклад.

На основі таблиці *Основні дані* додати до таблиці *Кредити* записи з *Табельним номером* 13 та 14.

Для цього потрібно виконати наступні дії.

1. Вибраємо об'єкт *Запит* бази **kadry**, далі *Створення запиту в режимі Конструктора, Додавання таблиці Основні дані*, пункт головного меню *Запит, Додавання*, або піктограму .

2. Відкриється діалогове вікно додавання записів (рис. 5.34), де потрібно вказати ім'я таблиці, до якої будуть добавлені записи.

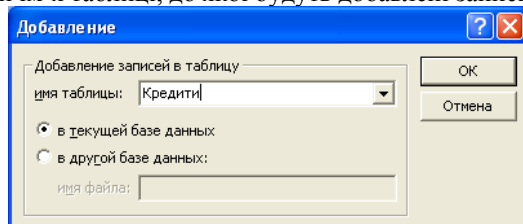


Рис. 5.34. Вигляд діалогового вікна вибору таблиці для додавання записів.

На рис. 7.12 подано відтворений у вікні броузера HTML-код використання внутрішніх посилань.

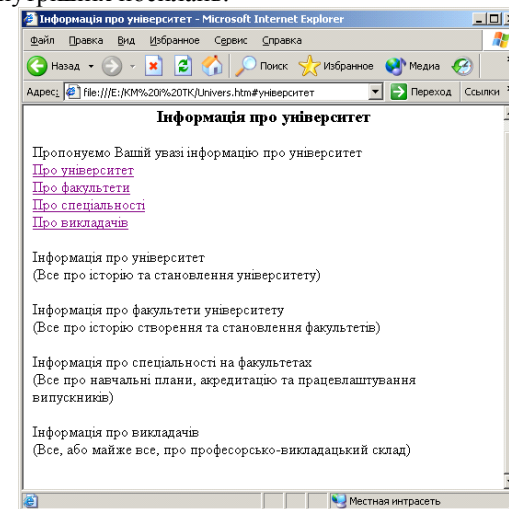


Рис. 7.12. Відтворений у вікні броузера HTML-код використання внутрішніх посилань.

елемент для створення на Web-сторінці графічного зображення (image). Цей елемент не має кінцевого дескриптора – уся необхідна інформація задається відповідними атрибутами. Використавши елемент IMG, як вкладений в елемент A, можна створити посилання на графічний файл або зображення.

Уявити собі сучасну Web-сторінку без графічних зображень практично неможливо. Такий стан речей обумовлений рядом переваг, які дає впровадження графічних об'єктів у структуру Web-сторінки, а саме:

- графічні зображення ділять текст на невеликі фрагменти, що покращує сприйняття змісту Web-сторінки;
- графіка ділить сторінку на логічно пов'язані між собою фрагменти, що робить Web-сторінку зручнішою для перегляду, розширює можливості дизайну сторінки;
- використавши графічні зображення, можна подати інформацію так, як її неможливо подати у вигляді тексту.

Для елемента IMG необхідним атрибутом є атрибут src, який означає джерело, ресурс (source) і вказує на те місце, де зберігається графічний файл або графічне зображення

Значенням атрибуту src може бути відносний або повний URL графічного об'єкту, наприклад:

Створюючи Web-сторінки їх автори передбачають для користувачів можливість швидкого переходу від одного розділу сторінки до іншого, використовуючи внутрішні посилання. Для внутрішніх посилань, як і для зовнішніх, необхідно створити текст посилання, на якому треба клацнути лівою клавішею "мишки", щоб здійснити перехід до того місця сторінки, на яке робиться посилання. До того ж, те місце, на яке здійснюється посилання, необхідно певним чином позначити. Для цього використовується атрибут name:

name="значання"

Наприклад:

 текст посилання

Символ # перед ім'я розділу є обов'язковим. Тепер, клацнувши лівою клавішею "мишки" на текст посилання, браузер виконає пошук у межах Web-сторінки, поки не знайде той розділ, якому присвоєно ім'я ім'я розділу. Цей розділ повинен бути певним чином описаний:

 необхідний розділ

Відзначимо, що значення атрибуту name (у даному прикладі ім'я розділу) браузером не відтворюється.


Для розуміння механізму реалізування внутрішніх посилань розглянемо наступний приклад:

```
<HTML>
<HEAD>
<TITLE>Інформація про університет</title>
</head>
<BODY>
<H3 align="center">Інформація про університет</h3>
Пропонуємо Вашій увазі інформацію про університет<BR>
<A href="#університет">Про університет </a><BR>
<A href="#Факультет">Про факультети </a><BR>
<A href="#спеціальність">Про спеціальності </a><BR>
<A href="#викладач">Про викладачів </a><BR><BR>
<A name="університет">Інформація про університет </a>
<BR>(Все про історію та становлення університету)<BR><BR>
<A name="Факультет">Інформація про факультети університету </a>
<BR>(Все про історію створення та становлення факультетів)<BR><BR>
<A name="спеціальність">Інформація про спеціальності на факультетах </a>
<BR>(Все про навчальні плани, акредитацію та працевлаштування випускників)<BR><BR>
<A name="викладач">Інформація про викладачів </a>
<BR>(Все, або майже все, про професорсько-викладацький склад)<BR>
<BR>
</body>
</html>
```

3. Заповнимо QBE-запит на додавання запитів до таблиці *Кредити* (рис. 5.35).

Поле:	Табельний номер	Прізвище	
Имя таблицы:	Основні дані	Основні дані	
Сортировка:			
Добавление:	Табельний номер	Прізвище	
Условие отбора:	13 Or 14		
или:			

Рис. 5.35. Таблиця QBE-запиту на додавання записів з табельними номерами 13 та 14 до таблиці *Кредити*.

4. Запустимо запит на виконання піктограмою  і отримаємо повідомлення (рис. 5.36).

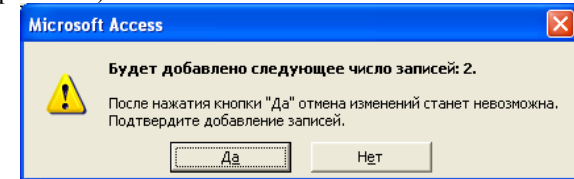


Рис. 5.36. Попередження про можливе додавання записів у таблиці *Кредити*.

5. Якщо дати підтвердження на додавання, то вибрані для додавання записи доповнять таблицю *Кредити*.

Як видно з самого QBE-запиту у таблицю *Кредити* будуть вставлені два нові записи з заповненими полями *Табельний номер* та *Прізвище*. Інші поля у новостворених записах будуть незаповненими.

Access не вимагає щоб у вбудованому запиті кількість полів була рівна кількості полів у таблиці, куди додаються дані. Можна записувати дані тільки до вибраних полів, не беручи до уваги інших. Важливим є те, щоб у запиті було поле, що відповідає полю ключа таблиці, куди додаються дані. Якщо головним ключем є поле типу лічильника, то в такій ситуації Access сам визначає значення головного ключа.

5.3.11. Запити, що виконують вибірку записів, які повторюються

Цей запит визначає, чи містить таблиця значення, що повторюються в одному або декількох полях.

Побудувати такий запит можна за допомогою *Майстра*. Загальну схему створення запитів на додавання записів продемонструємо на прикладі.

Приклад.

Серед записів таблиці *Кредити* вибрати такі записи, що повторюються за полем *Прізвище*.

Для цього потрібно виконати наступні дії.

1. У вікні бази даних натиснути кнопку *Запит, Створити*. Відкриється вікно, в якому із списку вибрати пункт *Записи, що повторюються* (рис. 5.37).

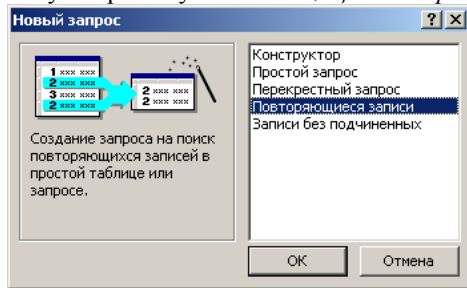


Рис. 5.37. Вигляд вікна для вибору записів, що повторюються.

2. У вікні, що відкриється, потрібно вибрати назву таблиці (у нашому випадку *Кредити*), для якої будемо вибирати записи, що повторюються (рис. 5.38).

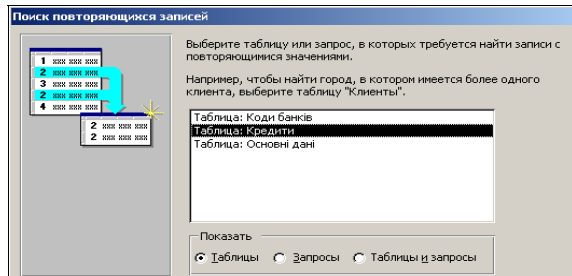


Рис. 5.38. Вибір таблиці, в якій буде пошук записів, що повторюються.

3. У вікні, котре відкриється (перший крок створення запиту), задаємо поля, за якими можливий пошук записів, що повторюються (у нашому випадку поле *Прізвище*), та додаткові поля (це вже у наступному кроці), котрі бажані у результатуючій таблиці (у нашому випадку, наприклад, поля *Табельний номер, Назва банку, Сума кредиту*, рис. 5.39).

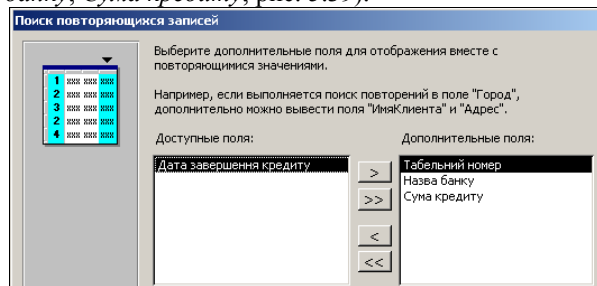


Рис. 5.39. Вибір додаткових полів, що будуть відображатися разом з полями, що повторюються.

```
<BODY>
<H3 align="center">HELLO !</h3>
<HR align="justify" size=4 color="red">
<H3 align="center">Я, Петрів Петро, вітаю Вас на своїй Web-сторінці</h3>
Пропоную Вашій увазі інформацію про себе: де я навчаюся, коло моїх наукових інтересів, улюблена музика, захоплення.
Я формував свою Web-сторінку використавши гіперпосилання.<BR>
<A href="page_2.htm">Де я навчаюся </a><BR>
<A href="page_3.htm ">Мої наукові інтереси </a><BR>
<A href="page_4.htm ">Улюблена музика </a> <BR>
<A href="page_5.htm ">Захоплення </a> <BR> <BR>
Можете контактувати зі мною електронною поштою <BR>
<A href="mailto:zed23_04@forest.lviv.ua">Петрів Петро </a>
</body>
</html>
```

Тут текстом посилання на файл *page_2.htm*, у якому міститься інформація про університет, де навчається *Петрів Петро*, є текст *Де я навчаюся*. При відтворенні HTML-коду у вікні броузера цей текст буде поданий без змін, але з ефектом підкресленого тексту (*Де я навчаюся*). Тепер, клацнувши лівою клавішею "мишки" на цьому підкресленому тексті, броузер здійснить перехід до файлу *page_2.htm* і відтворить його HTML-код. Якщо ж підвести вказівник манипулятора "мишка" на текст *Петрів Петро*, тоді у нижньому правому куті вікна броузера появиться напис *mailto:zed23_04@forest.lviv.ua*, де: *mailto:* вкаже на використання протоколу електронної пошти; *zed23_04* – ім'я поштової скриньки; *forest.lviv.ua* – доменне ім'я сервера на якому створена поштова скринька.

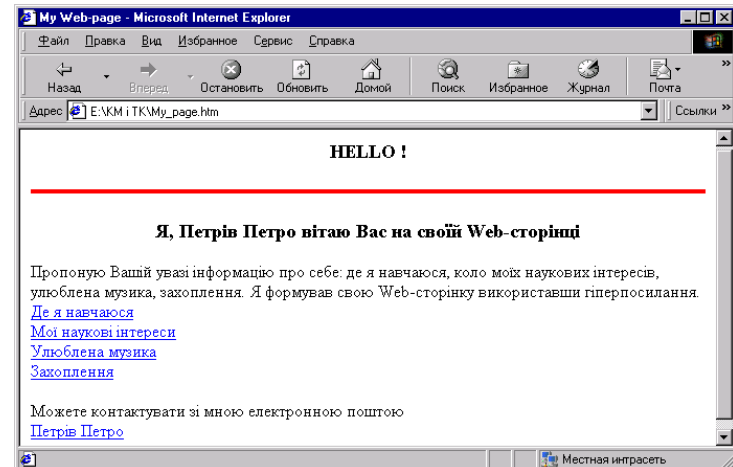


Рис. 7.11. HTML-код Web-сторінки з використання гіперпосилань відтворений у вікні броузера.

диску, тоді адреса посилання обов'язково починається словом file, тобто містить вказівку на протокол:

href="file://Диск:/шлях доступу до файлу"

href="file:///Диск:/шлях доступу до файлу"

За замовчуванням використовується посилання на файли поточної теки (теєї, де розміщений файл самої Web-сторінки). У цьому випадку просто вказується ім'я файлу.

Дозволяється використовувати відносні посилання на теки. Це дає можливість легко змінювати місце розташування комплексу сторінок на диску. Якщо у поточній теці є інша, у якій є необхідні файли, посилання формується за таким шаблоном:

href=../Тека/Ім'я файлу.тип

На структуру вкладених тек вказує крапка перед похилою рискою. Якщо ж необхідно вказати на теку, яка знаходиться на тому ж рівні вкладеності, що і поточна, тоді додають ще одну крапку:

href=../Тека/Ім'я файлу.тип

Якщо в якості значення атрибуту href застосована так звана схема доступу, використовують такий шаблон:

href="Протокол://Адреса посилання"

Схема доступу визначає тип сервера, який доступний за допомогою даного посилання. Для користувача це є можливість доступу до сервісів Internet:

- file – доступ до файлу на локальному диску;
- ftp – доступ до архівів файлів за протоколом передавання файлів;
- http – доступ до WWW;
- mailto – відсилання повідомлення електронною поштою;
- news – доступ до новин USENET;
- nntp – доступ до новин USENET за протоколом NNTP;
- telnet – під'єднання за протоколом Telnet;
- wais – під'єднання до системи пошуку WAIS.

У випадку, коли гіперпосилання використовується для вказання адреси електронної пошти, реалізування цього гіперпосилання забезпечує не перехід до нового документу, а активування діалогу для висилання повідомлення вказаному адресатові. Як правило, таке посилання розміщують у кінці сторінки, щоб забезпечити можливість спілкування з її автором. Наприклад:

Петрів Петро

Використання гіперпосилань і те, яким чином броузер відтворить HTML-код Web-сторінки віртуального студента (рис. 7.11), продемонструємо на прикладі.

```
<HTML>
<HEAD>
<TITLE>My Web-page</title>
</head>
```

4. У вікні, що відкриється, потрібно задати назву шойно створеного запиту та зберегти його.

5. Запит автоматично стартує. Результат роботи запиту буде наступний (рис. 5.40).

Поиск повторений для Кредити : запрос на выборку			
Прізвище	Табельний ном	Назва банку	Сума кредиту
Ваків	2	Укрсоцбанк	4 440,00р.
▶ Ваків	2	Укргазбанк	3 000,00р.
Ваків	2	Аваль	1 230,00р.
Ваків	2	Укргазбанк	2 222 222,00р.
Ваків	11	Укргазбанк	2 220,00р.
*	0		0,00р.

Рис. 5.40. Результат пошуку записів, що повторюються за полем *Прізвище* у таблиці *Кредит*.

Цей запит в таблиці QBE буде мати наступний вигляд (рис. 5.41).

Поле:	Прізвище	Табельний номер	Назва банку	Сума кредиту
Інша таблиця:	Кредити	Кредити	Кредити	Кредити
Сортировка:	по возрастанию			
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Условие отбора:	In (SELECT [Прізвище] FROM [Кредити] As Tmp GROUP BY [Прізвище] HAVING Count(*)>1)			
или:				

Рис. 5.41. Вигляд вікна QBE-запиту на пошук записів, що повторюються за значенням поля *Прізвище* у таблиці *Кредит*.

5.4. Форми в MS ACCESS

В попередніх розділах були розглянуті такі об'єкти Access як таблиці та запити, які користувач не завжди бачить. Непідготовлений користувач використовує базу даних через інтерфейс, який створив програміст. Одним з основних елементів такого інтерфейсу є форми. Форми дають можливість користувачам вводити дані у таблиці бази даних без безпосереднього доступу до самих таблиць, а також виводити або переглядати результати не у вигляді таблиць, а у вигляді якісно оформлених форм.

За допомогою форм можна змінювати рядки SQL-операторів запитів і запускати їх на виконання.

5.4.1. Створення нової форми

Для створення нової форми в Access існує сім способів. Щоб почати створення форми одним із способів потрібно у вікні бази даних вибрати форму і клацнути на кнопку "Створити". На екрані з'явиться вікно (рис. 5.42) *Нова форма*, в якому будуть виведені способи створення форм. У нижній частині цього вікна можна вибрати джерело даних для форми.

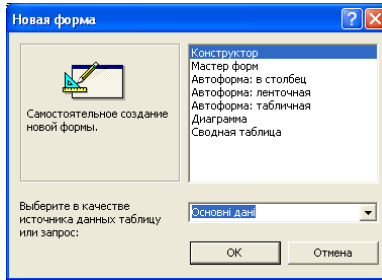


Рис. 5.42. Возможні способи створення форми.

Розглянемо детальніше кожен із способів створення форми.

5.4.2. Режим Конструктора

Після вибору в діалоговому вікні пункту *Конструктор* на екрані відкривається незаповнена форма (рис. 5.43).

Форма, як і будь-який документ, має свою структуру. Щоб побачити основні розділи форми, потрібно у меню *Вид* вибрати усі необхідні пункти, тобто *Лінійка*, *Сітка*, *Колонтитули*, *Заголовок/примітки форми* (рис 5.44).

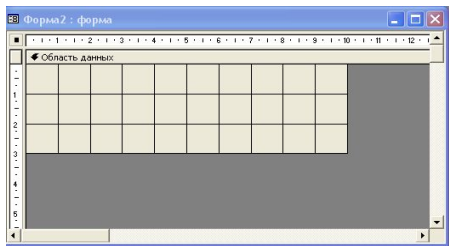


Рис. 5.43. Вигляд незаповненої форми

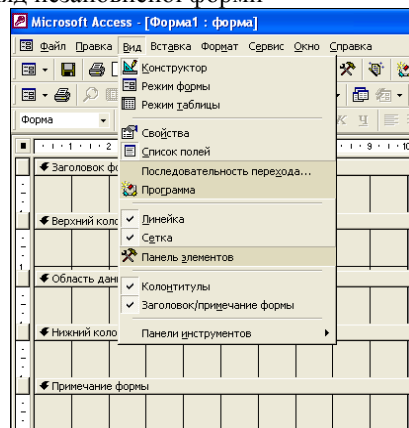


Рис. 5.44. Розділи незаповненої форми.

7.8. Гіперпосилання

Можливість переходу від однієї Web-сторінки до іншої забезпечується наявністю в HTML одного з найважливіших елементів – елементу реалізування гіперпосилання. Гіперпосилання – фрагмент тексту або графічний об'єкт, який є вказівником переходу до іншого Web-документу. Існує три основні види гіперпосилань: внутрішні, зовнішні та відносні.

Внутрішні посилання (internal links) – це посилання на об'єкти у межах одного документу. Застосовуючи їх користувач може здійснювати переміщення однією і тією ж Web-сторінкою. Цей тип посилань доцільно використовувати у довгих сторінках з метою швидкого пересування між її розділами.

Зовнішні (external links) або локальні (distant links) посилання – це посилання на інші Web-сервери.

Відносні (relative links) або локальні (local links) посилання – це посилання на інші Web-сторінки (або сервіси Internet), які розміщені на тому ж сервері, що і Web-сторінка, яка містить посилання. Такі посилання називаються відносними тому, що їх адреси подаються відносно до адреси Web-сторінки, яка містить посилання.

`<A> `

елемент, який реалізує перехід до іншого документу або графічного об'єкту. Найчастіше використовують такі два формати для запису елементу гіперпосилання:

Текст ` текст посилання `;

` `.

Перший формат використовується у тому випадку, коли гіперпосилання реалізується безпосередньо у тексті Web-сторінки. Значення атрибуту href є вказівником на ресурс Internet, файл на локальному диску або на якір (позначку) всередині сторінки. Текст, який розміщений всередині елементу A, є видимою частиною гіперпосилання. У вікні броузера цей текст відзначений підкресленням та кольором і на ньому потрібно клацнути лівою клавішею "мишки", щоб здійснити перехід за адресою ресурсу, яка є значенням атрибуту href. Після використання гіперпосилання цей фрагмент тексту змінює колір.

Конкретизуємо, що будемо розуміти під виразом "Адреса посилання", яка є значенням атрибуту href. Це URL (Uniform Resource Locator) – уніфікований локатор ресурсів. URL – це адреса Web-сторінки, котра відтворюється у полі Адрес (Location або URL), яке розміщене у лівому верхньому куті вікна броузера. URL складається з двох основних елементів: типу протоколу і адреси, за якою здійснюється посилання. Тип протоколу визначає тип ресурсу Internet, з яким встановлюється зв'язок, а адреса – це ім'я файлу, каталогу або сервера, на ресурси яких здійснюється посилання.

Другий формат використовується у тому випадку, коли видима частина гіперпосилання є графічним об'єктом. Якщо графічний об'єкт відзначений рамкою, тоді вона також змінює колір після реалізування гіперпосилання. Якщо адреса посилання вказує на рисунок, котрий знаходиться на локальному

COLGROUP, а частина властивостей може бути іншою. У таблиці можуть бути визначені властивості для довільної кількості стовпців. Якщо кількість реальних стовпців буде меншою, ніж кількість оголошених, тоді останні визначення будуть непотрібні. Це не буде помилкою.

Використовуючи елементи COLGROUP і COL важливо дотримуватися певних правил. У розумінні опису властивостей елемент COLGROUP має вищий пріоритет, тому елементи COL використовуються всередині елементу COLGROUP. Формуючи таблицю можна використати кілька елементів COLGROUP. Якщо кількість стовпців у одному елементі COLGROUP задається значенням атрибуту span, тоді використовувати у ньому елемент COL немає змісту. Якщо елементи COL наявні, тоді атрибут span відповідного елементу COLGROUP ігнорується. Тобто, кількість стовпців визначається кількістю елементів COL. Для окремих елементів можна використовувати власні атрибути span. Наприклад:

```
<HTML>
<HEAD>
<TITLE>Елементи COLGROUP, COL</title>
</head>
<TABLE border=4 align="center" bgcolor="aqua">
<COLGROUP span=1 width=80 bgcolor="orange">
<COLGROUP bgcolor="yellow">
<COL span=2 width=100> <COL width=80>
<COLGROUP bgcolor=#C0C0C0> <COL width=120>
<TR> <TD>Комірка 1.1
<TD>Комірка 1.2 <TD>Комірка 1.3 <TD>Комірка 1.4
<TD>Комірка 1.5
<TR> <TD>Комірка 2.1
<TD>Комірка 2.2 <TD>Комірка 2.3 <TD>Комірка 2.4
<TD>Комірка 2.5
</table>
</body>
</html>
```

На рис. 7.10. поданий вигляд таблиці, яка створена з використанням елементів COLGROUP і COL у вікні браузера.

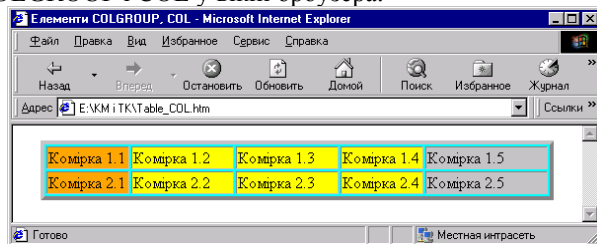



Рис. 7.10. Вигляд таблиці, яка створена з використанням елементів COLGROUP і COL у вікні браузера.

Розглянемо основні складові елементи структури форми.

Область виділення форми. Для того, щоб розпочати роботу з усією формою, її потрібно зробити активною, тобто клацнути лівою клавшею мишки в області виділення форми (знаходиться на перетині горизонтальної та вертикальної лінійок). Щоб зробити активним розділ форми достатньо клацнути мишкою по заголовку розділу. В кожен момент активним є лише один розділ. Якщо потрібно відредагувати властивості форми, то спочатку її виділяють, а потім клацають на кнопки *Властивості* (кнопка панелі інструментів ).

Область виділення розділу служить для виділення розділу форми. Якщо розділ активний, то колір заголовку інвертується (фон стає червоним, а шрифт білим).

Горизонтальна та вертикальна лінійки вирівнювання використовуються для вирівнювання елементів форми один відносно другого і для задання їх точного положення в розділі. За допомогою лінійок вирівнювання і курсора мишки можна виділяти групу елементів керування в формі. Для цього курсор мишки поміщають в область лінійки. Далі натискають ліву кнопку мишки і тримаючи її переміщують курсор по лінійці. Після відпускання кнопки мишки, всі елементи смуги будуть виділеними. Появу і зникнення у вікні *Конструктора* форм лінійок вирівнювання можна здійснювати за допомогою пункту меню *Вид* і підменю *Лінійка*.

Область даних — тут розміщуються елементи керування.

Лінії координатної сітки — використовуються так само як і лінійки вирівнювання для вирівнювання елементів керування. Виведення сітки можна включати або виключати за допомогою меню *Вид* і підменю *Сітка*. Густоту вузлів сітки можна змінювати за допомогою властивостей форми.

Число поділок по X і число поділок по Y. За замовчуванням число поділок як по X, так і по Y дорівнює 10. Це означає, що лінії сітки розміщені через 10 мм.

Кожна форма може мати п'ять розділів.

Заголовок форми — у заголовку розміщується інформація, спільна для всіх записів. Заголовок розміщується у верхній частині вікна або на початку першої сторінки під час друкування.


Верхній і нижній колонтитули — інформація з верхнього або нижнього колонтитула розміщується на початку або в кінці кожної сторінки. У режимі форми колонтитули не відображаються.

Область даних — у цьому розділі знаходиться основна інформація.

Примітки форми — тут міститься інформація, спільна для всіх записів. Примітка розміщується внизу вікна в режимі форми і в кінці останньої сторінки під час друку.

Форма, що починає створюватися містить один розділ — *область даних*. Решту розділів, під час потреби, додаються в форму, причому додавання здійснюються попарно. Наприклад, верхній колонтитул з нижнім, заголовок з примітками. Додати новий розділ у форму можна за допомогою меню *Вид* і

підпунктів меню *Колонтитули* чи *Заголовок / Примітки*. Це можна зробити також за допомогою контекстного меню. Елементи керування, що додаються в розділ, можна переміщати тільки в цьому розділі. Перекидання елементу керування з одного розділу в інший здійснюється через буфер обміну.

Разом з вікном нової форми на екрані відкривається панель елементів (рис. 5.45). Вона містить елементи, які можна розмістити на формі. Відкрити або заховати панель елементів можна за допомогою кнопки  на панелі інструментів MS Access в режимі *Конструктора* форми або за допомогою меню *Вид* і підменю *Панель інструментів*.

Майстер елементів. Кнопка може бути включеною або виключеною. Якщо кнопка включена, то після вибору елемента і розміщення його на формі на екрані монітора будуть появлятися відповідні *Майстри*.

Надпис. Використовуються для побудови елементів керування *надпис*. Надписи не є зв'язаними елементами. Надпис може бути приєднаний до будь-якого елемента. Тоді надпис називається підписом. Надпис на відміну від інших елементів не може бути активним. Текст в написі задається властивістю. **Підпис.** Підпис може містити до 2048 символів. Як буде відображатися текст надпису можна задати властивостями, які можна задати у вікні Властивості, що відкривається через контекстне меню.

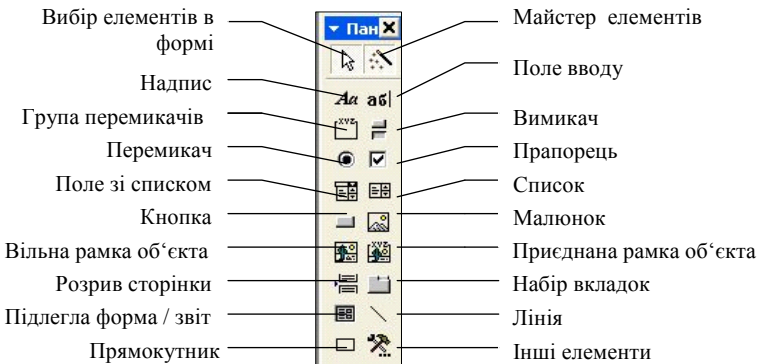


Рис. 5.45. Структура панелі елементів.

Дамо коротко пояснення тим кнопкам, що розміщені на панелі елементів.

Вибір елементів у формі. Ця кнопка використовується для того, щоб вибрати один або декілька елементів форми.

Поле. Відображає значення виразів відповідного типу, їх редагування. Поле може бути приєднаним, вільним, або обчислювальним. Обчислювальне поле містить у властивості дані вираз. Вираз повинен починатися із знаку = (дороівнює) і може бути числовим, текстовим або логічним. У приєднаному полі в режимі *Конструктора* відображається назва поля таблиці або запити,

```
<TFOOT>
<TR> <TD>Нижній блок таблиці<TD>&nbsp;
<TBODY><TR> <TD>Факультет 1 <TD>Група 1.1
<TR> <TD>Факультет 2 <TD>Група 2.1
<TBODY><TR> <TD>Факультет 3 <TD>Група 3.1
<TR> <TD>Факультет 4 <TD>Група 4.1
<TR> <TD>Факультет 5 <TD>Група 5.1
</table>
</body>
</html>
```

На рис. 7.9. поданий вигляд таблиці, яка створена з використанням елементів THEAD, TFOOT та TBODY у вікні браузера.

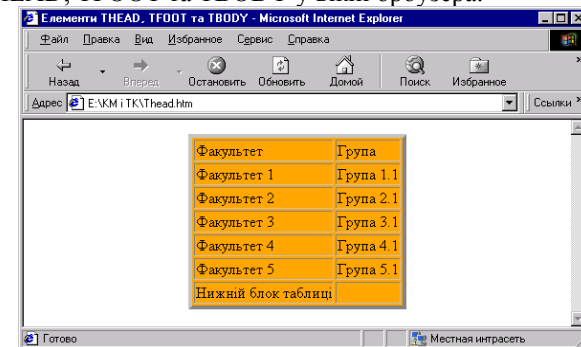


Рис. 7.9. Вигляд таблиці, яка створена з використанням елементів THEAD, TFOOT та TBODY у вікні браузера Internet Explorer.

При формуванні цієї таблиці до рядків і комірок не застосовувалися елементи форматування, тому таблиця має простий вигляд. Однак, використання елементів груп рядків дають можливість визначення додаткових стилів. Використовуючи ці елементи, необхідно дотримуватися таких правил:

у таблиці можна використати по одному елементу THEAD і TFOOT, але декілька TBODY;

послідовність використання елементів є такою: THEAD, TFOOT та TBODY;

в усіх блоках повинна бути однакова кількість стовпців.

```
<COLGROUP>
```

елемент, призначений для формування групи стовпців з однаковими властивостями. Кожний елемент COLGROUP надає однакові властивості заданій кількості стовпців, які задаються використанням атрибуту span. Усі ці стовпці при відтворенні таблиці браузером будуть однаковими.

```
<COL>
```

елемент, призначений для надання однакових властивостей кожному стовпцеві. Використання елемента дає можливість задання частини спільних властивостей для усіх стовпців, які відносяться до одного елемента

Продовження таблиці 7.3

166	¦	¦	Вертикальна риска	
167	§	§	Знак параграфу	§
169	©	©	Знак копірайт	©
171	«	«	Ліві типографські лапки	"
172	¬	¬	Знак заперечення	¬
174	®	®	Знак "зарєєстровано"	®
176	°	°	Знак градуса	°
177	±	±	Знак "плюс-мінус"	±
178	²	²	До степеня 2	²
179	³	³	До степеня 3	³
181	µ	µ	Знак "мікро"	μ
182	¶	¶	Знак абзацу	¶
183	·	·	Крапка – маркер	·
184	¹	¹	До степеня 1	¹
187	»	»	Праві типографські лапки	"
188	¼	¼	Одна четверта	¼
189	½	½	Одна друга	½
215	×	×	Множення	×
247	÷	÷	Ділення	÷

Увага, подвійні лапки, кутові дужки і амперсанд (" , < , > , &) в HTML є службовими символами. Якщо виникає необхідність використовувати ці символи у звичайному тексті Web-сторінки, тоді вони повинні бути записані тільки за допомогою кодів.

У HTML є можливість групувати рядки таблиці використовуючи такі елементи:

<THEAD>

елемент блоку заголовку;

<TFOOT>

елемент звичайних блоків рядків;

<TBODY>

елемент нижнього блоку рядків.

У кожному блоці може бути довільна кількість рядків (елементів TR). Ці три елементи можуть використовуватися як з кінцевими дескрипторами, так і без них. Використання елементів THEAD, TFOOT та TBODY продемонструємо на такому прикладі:

<HTML>

<HEAD>

<TITLE>Елементи THEAD, TFOOT та TBODY</title>

</head>

<TABLE border=4 align="center" bgcolor="orange">

<THEAD><TR> <TD>Факультет <TD>Група

яке буде джерелом даних. Вільне поле містить рядок "вільний", а в обчислювальному полі відображається вираз у властивості Дані. Для елемента *поле* джерелом даних може бути поле МЕМО. У вікні властивості для цього поля можна задати і інші властивості.

Група. Ця кнопка використовується у формах для побудови елемента керування *група* і надає користувачу можливість вибору одного з декількох варіантів. Група перемикачів складається з рамки групи і набору параметрів. У кожен момент користувач може вибрати тільки один параметр.

Вимикач. Ця кнопка використовується для відображення або введення даних логічного типу включено/виключено (або 1 та 0, або Так/Ні). Приєднаний вимикач джерелом даних має поле таблиці логічного типу.

Перемикач. Так само як кнопка вимикач, ця кнопка використовується для відображення і введення логічних даних. Особливістю перемикачів є те, що з групи перемикачів можна вибрати тільки один.

Прапорець. Значення цього елемента аналогічні до значень елементів вимикач і перемикач, але на відміну від них допускають множинний вибір.

Поле із списком. Поле із списком є комбінацією з двох елементів керування поля і списку. За допомогою цього елемента можна вибрати значення з наперед заданих, із випадючого списку. Поле із списком займає менше місця у формі, оскільки відкривається після клацання на кнопці, що його розкриває.

Список. Так само як попередня кнопка, цей елемент використовується для вибору значень з наперед заданих. Список дає можливість зразу оглядати частину рядків, що в ньому містяться.

Кнопка. Кнопка використовується для виконання дій. Основна властивість кнопки – це подія натискування кнопки (Click). За допомогою цієї кнопки можна здійснювати запуск макроса або процедури VBA.

Малюнок. Ця кнопка дає можливість розмістити на формі в рамці малюнок. Для малюнку можна задавати адресу гіперпосилання. Якщо клацнути на ньому, то буде здійснено перехід за заданою адресою. Для керування рисунком можна задавати властивості, які можна відкрити через контекстне меню.

Вільна рамка. Ця кнопка використовується для включення в форму об'єктів OLE. Вільна рамка об'єкта не має джерела даних. Об'єкт OLE після включення стає частиною форми.

Приєднана рамка. Ця кнопка використовується так само як вільна рамка. Об'єкти OLE повинні міститися у таблиці бази даних.

Розрив сторінки. Цей елемент використовується для того, щоб розпочати друкування з нової сторінки. У конструкторі форм він відображається зліва як точкова лінія. У режимі перегляду розрив не відображається. Розрив сторінки використовується для побудови багатосторінкових форм.

Підлегла форма. В MS Access всередині форми можна розміщувати іншу форму. Форма, що містить іншу форму, називається *головною*, а форма, що міститься в іншій – *підлеглою*. Головна форма може містити будь-яке число

підпорядкованих форм. Елемент *підлегла* форма використовується для побудови таких форм.

Лінія. Використовується для включення у форму відрізків прямих ліній. Лінії використовують для оформлення зовнішнього вигляду форми. Лініям можна задавати властивості – колір, товщину та розмір.

Прямокутник. Цей елемент використовується для включення у форму прямокутників. Він використовується так само як лінія.

Active X. Ця кнопка, не відповідає жодним елементам керування. Клацнувши на цій кнопці, на екрані відкриється вікно, що буде містити список всіх зареєстрованих у системі елементів Active X.

Набір вкладок. Елемент керування *набір вкладок* використовується для розміщення елементів керування на декількох сторінках. Цей елемент використовується тоді, коли елементи не поміщаються на одній сторінці. Доступ до вкладок можна зробити через ярлички. З самого початку набір вкладок має тільки дві сторінки. За допомогою контекстного меню і пункту *додати вкладку* або *викинути вкладку* можна збільшити або зменшити кількість вкладок. Пункт *послідовність вкладок* контекстного меню дає можливість керувати порядком розміщення сторінок у наборі вкладок.

Вибір елемента керування здійснюється одним клацанням на відповідному значку панелі інструментів, далі наступне клацання в полі форми де цей значок повинен бути встановлений. Для відміни вибору потрібно клацнути на кнопці *вибору об'єкта* і ще раз клацнути на кнопці вибраного елемента або натиснути на клавішу ESC на клавіатурі.

Разом з елементом у поле форми встановлюється надпис, який приєднаний до нього. Редагування властивостей елемента керування здійснюється через контекстне меню. Цьому елементу можна дати змістовний підпис, який буде відповідати програмі. Основними елементами оформлення форми є текстові надписи і рисунки. Для створення в формі текстових надписів використовуються два елементи *Надпис* і *Поле*. У ролі надпису можна давати будь-який текст. У елементі *Поле* відображається вміст одного з полів таблиці, на якій побудована форма. Під час переходу на інший запис текст може змінюватися.

Для створення графічних елементів використовуються такі елементи керування як *Рисунок*, *Довільна рамка об'єкту* і *Приєднана рамка об'єкту*. Рисунок вибирається з файлу і встановлюється в форму. Елемент *Довільна рамка об'єкту* може містити будь-який об'єкт OLE. Вміст цього елемента береється з таблиці бази даних. Під час переходу між записами вміст цього елемента буде змінюватися.

Кнопка *Вибору об'єкта*, як правило, завжди активізована. Коли вибирається один з елементів керування, клацаючи лівою клавішею мишки на відповідній кнопці, кнопка відпускається. Ще раз клацнувши на кнопці вибору об'єктів, можна відмінити зроблений вибір.

Кнопка *Майстра* – якщо ця кнопка активізована, то під час переносі елемента керування на форму запускається відповідний *Майстер*. Інакше новий елемент поміщається тільки у форму.

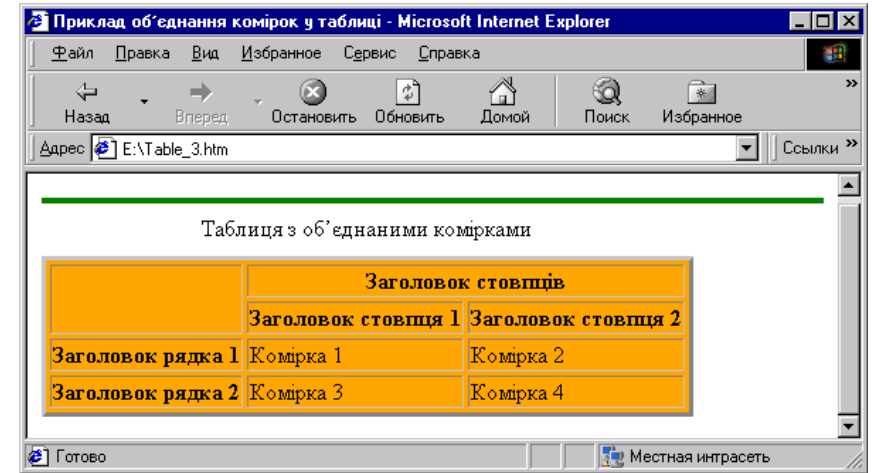


Рис. 7.8. Вигляд таблиці з об'єднаними комірками рядків і стовпців у вікні браузера Internet Explorer.

Зауважимо, що у комірці, яка не містить тексту, занесено символ нерозривного пробілу. Це необхідно для того, щоб сітка таблиці була правильно відтворена браузером.

У HTML і браузерах реалізована можливість відтворення символів за їх кодами. Символи можуть бути загальноприйнятими, введеними з клавіатури, нестандартними і використаними в HTML у якості службових. Усіх їх будемо називати спеціальними символами. Довільний символ можна задати числовим кодом або мнемонічним кодом. У табл. 7.3 подано коди найчастіше уживаних символів (за стандартом ISO 8859-1). Символом, який відділяє код символу від тексту є ";" (крапка з комою).

Таблиця 7.3

Коди символів				
Код символу	Числовий код	Мнемонічний код	Назва	Символ
1	2	3	4	5
34	"	"	Праві лапки	"
38	&	&	Амперсанд	&
60	<	<	Знак "<"	<
62	>	>	Знак ">"	>
153	™	™	Торгівельна марка	™
160	 	 	Нерозривний пробіл	
162	¢	¢	Цент	¢
163	£	£	Фунт	£
164	¤	¤	Знак валюти	¤
165	¥	¥	Йена	¥

На рис. 7.7 подано вигляд таблиці з заголовками рядків і стовпців у вікні браузера.

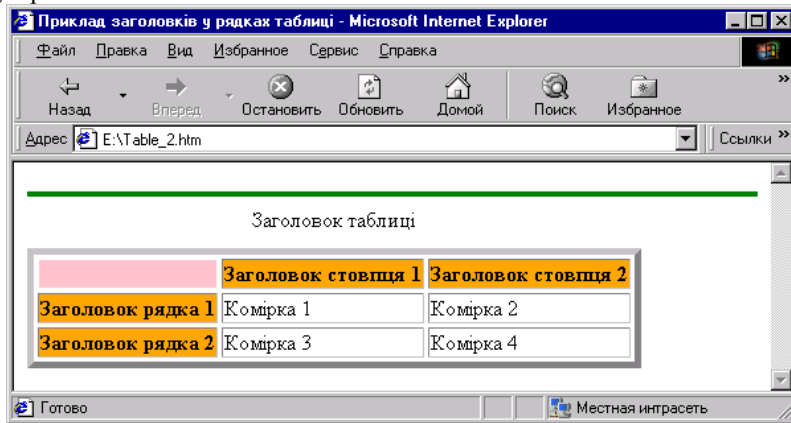


Рис. 7.7. Вигляд таблиці з заголовками рядків і стовпців у вікні браузера Internet Explorer.

Інколи виникає необхідність в об'єднанні комірок рядків та стовпців, як це подано у наступному прикладі:

```
<HTML>
<HEAD>
<TITLE>Приклад об'єднання комірок у таблиці</title>
</head>
<BODY>
<HR align="justify" size=4 color="green">
<TABLE border=3 cellspacing=3 bgcolor="orange">
<CAPTION> Таблиця з об'єднаними комірками </caption>
<TR><TH rowspan=2>&nbsp;&nbsp;&nbsp;<TH colspan=2> Заголовок стовпців
<TR><TH> Заголовок стовпця 1 <TH> Заголовок стовпця 2
<TR><TH> Заголовок рядка 1
<TD>Комірка 1
<TD>Комірка 2
<TR><TH> Заголовок рядка 2
<TD>Комірка 3
<TD>Комірка 4
</table>
</body>
</html>
```

На рис. 7.8 подано вигляд таблиці з об'єднаними комірками рядків і стовпців у вікні браузера.

5.5. Приклади роботи з MS ACCESS

Завдання 1.

Створити базу даних **Кадри**, яка буде містити наступні поля: Табельний номер, Прізвище, Ім'я, По-батькові, Стать, Дата народження, Сімейний стан, Кількість дітей, Дата прийняття на роботу, Посадовий оклад, Підрозділ, Примітка, Фото. На основі даної таблиці створити форму з наступними елементами керування: наступний запис, попередній запис, друк запису. Створити запит на вибірку усіх осіб жіночої статі. Створити запит для вибірки тих працівників, зарплати яких знаходиться у межах від 1000 грн. до 1500 грн. Створити запит для вибірки даних з сортуванням та умовою відбору за датою прийняття на роботу. На основі побудованої бази створити звіт. Хід роботи:

1. Створюємо базу даних **Кадри**, яка буде містити таблицю *Основні дані*. Для цього використовуємо пункт *Створення таблиці в режимі Конструктора* на закладці *Таблиця*. Обов'язковими полями є:

- Табельний номер – тип даних Числовий;
- Прізвище – тип даних Текстовий;
- Ім'я – тип даних Текстовий;
- По – батькові – тип даних Текстовий;
- Стать – тип даних Текстовий;
- Дата народження – тип даних Дата/час;
- Сімейний стан – тип даних Текстовий;
- Кількість дітей – тип даних Числовий;
- Дата прийняття на роботу – тип даних Дата/час;
- Посадовий оклад – тип даних Грошовий;
- Підрозділ – тип даних Текстовий;
- Примітка – тип даних Мемо;
- Фото – тип даних Поле об'єкту OLE;

У результаті отримуємо наступне (рис. 5.46):

Імя поля	Тип даних
Табельний номер	Числовий
Прізвище	Текстовий
Ім'я	Текстовий
По-батькові	Текстовий
Стать	Текстовий
Дата народження	Дата/время
Сімейний стан	Текстовий
Кількість дітей	Числовий
Дата прийняття на роботу	Дата/время
Посадовий оклад	Денежный
Підрозділ	Текстовий
Примітка	Поле MEMO
Фото	Поле объекта OLE

Рис. 5.46. Створення таблиці *Кадри* в режимі *Конструктора*.

Закриваємо режим *Конструктора* та відкриваємо таблицю в режимі вводу даних. Вводимо 10 довільних записів.

2. На основі таблиці *Основні дані* створюємо форму *Основні дані* для вводу даних, яка буде містити наступні поля:

- Табельний номер;
- Прізвище;
- Ім'я;
- По-батькові;
- Стать;
- Дата народження;
- Сімейний стан;
- Кількість дітей;
- Дата прийняття на роботу;
- Посадовий оклад;
- Підрозділ;
- Примітка;
- Фото;

У результаті отримуємо (рис. 5.47):

Рис. 5.47. Форма *Основні дані* з відкритим першим записом.

3. Створюємо елементи керування формою. Для цього переходимо в режим *Конструктора* форми, за допомогою мишки змінюємо розміри *Області даних*, на панелі елементів вибираємо функційний елемент *Кнопка* та

```
<TD>Комірка 2
<TR><TD>Комірка 3
<TD>Комірка 4
</table>
<HR align="justify" size=4 color="green">
</body>
</html>
```

На рис. 7.6 подано вигляд простої таблиці у вікні браузера.

Як видно з поданого прикладу, текст, поміщений після елементів TD, є не чим іншим, як вмістом комірки.

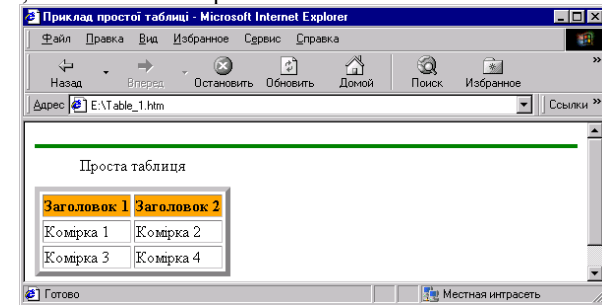


Рис. 7.6. Вигляд простої таблиці у вікні браузера Internet Explorer.

Розглянемо трохи складніший приклад, додавши до заголовків стовпців ще і заголовки рядків:

```
<HTML>
<HEAD>
<TITLE>Приклад заголовків у рядках таблиці</title>
</head>
<BODY>
<HR align="justify" size=4 color="green">
<TABLE border=5 cellspacing=3>
<CAPTION> Заголовок таблиці </caption>
<TR><TH bgcolor="pink">
<TH bgcolor="orange"> Заголовок стовпця 1
<TH bgcolor="orange"> Заголовок стовпця 2
<TR><TH bgcolor="orange"> Заголовок рядка 1
<TD>Комірка 1
<TD>Комірка 2
<TR><TH bgcolor="orange"> Заголовок рядка 2
<TD>Комірка 3
<TD>Комірка 4
</table>
</body>
</html>
```

напрямі і може набувати таких значень:

- top – вирівнювання до верхнього краю комірки;
- bottom – вирівнювання до нижнього краю комірки;
- center – вирівнювання до центру;
- baseline – вирівнювання за першим рядком.

<TR>

елемент, який формує рядок таблиці. Він не має кінцевого дескриптора. Рядок закінчується там, де починається наступний, тобто наступний елемент TR. Всередині елемента розміщуються елементи TH і TD, які визначають окремі комірки. Для вирівнювання даних у горизонтальному і вертикальному напрямках використовуються атрибути align і valign.

<TH>

елемент комірки-заголовку, який є заголовком стовпця або рядка таблиці. Цей елемент повинен розташовуватися всередині елемента TR. Комірка-заголовок відрізняється від звичайної тим, що браузер відтворює текст всередині комірки напіввідзначеним шрифтом.

Комірки можна об'єднувати використовуючи атрибути rowspan та colspan. Задавши значення атрибуту

rowspan=n і умову, що n>1, відповідна комірка займе не один, а n рядків і буде мати розмір у n разів більший, ніж звичайна комірка даного стовпця. Так само, використовуючи атрибут colspan, можна формувати комірки у декількох стовпцях.

Розміри комірок можна формувати точно задавши її висоту і ширину:

- width=ширина
- height=висота

<TD>

елемент, який визначає параметри звичайної комірки таблиці. Для цього елемента допускається використання тих же атрибутів, що і для елемента TH. Обидва елементи, TH і TD, можуть не мати кінцевих дескрипторів. Функцію кінцевих дескрипторів виконує наступний елемент, який визначає структуру таблиці.

Розглянемо приклад HTML-коду формування простої таблиці:

```
<HTML>
<HEAD>
<TITLE>Приклад простої таблиці</title>
</head>
<BODY>
<HR align="justify" size=4 color="green">
<TABLE border=4 cellspacing=3>
<CAPTION> Заголовок таблиці </caption>
<TR><TH bgcolor="orange"> Заголовок 1
<TH bgcolor="orange"> Заголовок 2
<TR><TD>Комірка 1
```

вставляємо його в область даних форми, у стовпці Категорії вибираємо *Переходи по записам*, у стовпці Дії – *Наступний запис – Готово* (рис. 5.48).

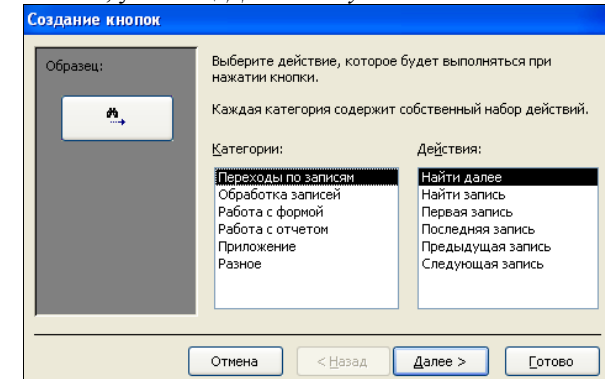


Рис. 5.48. Вікно створення об'єкту *Кнопка* на формі.

Повторяємо операцію для встановлення функційних кнопок *Попередній запис* та *Друк запису*. Результат виконання подано на рис. 5.49.

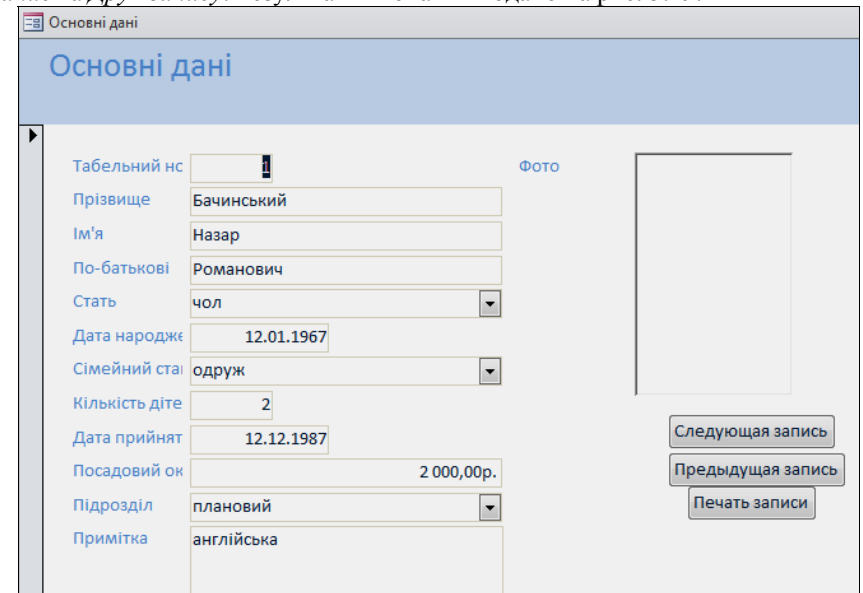


Рис. 5.49. Вікно форми зі створеними функціональними кнопками.

5. Створюємо запити для вибірки інформації за статтю працівників. Для цього переходимо до об'єкта *Запити – Створення запиту за допомогою Майстра*. В якості вихідного елемента вибираємо таблицю *Основные дані*,

додаємо всі доступні поля, Далі – Далі – вибираємо *Змінити макет запиту – Готово* (рис. 5.50).

Поле: Имя таблицы: Сортировка: Вывод на экран: Условие отбора: или:	Табельний номер Основні дані	Прізвище Основні дані	Ім'я Основні дані	По-батькові Основні дані	Стать Основні дані	Дата народження Основні дані	Сімейний стан Основні дані
	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Рис. 5.50. Фрагмент таблиці створення запиту за технологією QBE.

Для стовпця *Стать* у полі *Умова вибірки* вводимо умову – жін (як приклад). Закриваємо запит із збереженням змін. В результаті отримуємо (рис. 5.51).

абельний	Прізвище	Ім'я	По-батькові	Стать	Дата народ	Сімейний с	Кількість ді
4	Дідухів	Галина	Іванівна	жін	01.01.1991	неодруж	0
5	Залужна	Христина	Семенівна	жін	03.03.1990	одруж	1
10	Пронів	Олена	Василівна	жін	08.08.1960	одруж	4

Рис. 5.51. Фрагмент таблиці з результатами запиту.

6. Створюємо запит для вибірки даних зарплати у межах від 1000 грн. до 1500 грн. Для цього переходимо до об'єкту *Запити – Створення запиту за допомогою Майстра*. В якості вихідного елемента вибираємо таблицю *Основні дані*, додаємо всі доступні поля, Далі – Далі – вибираємо *Змінити макет запиту – Готово*. Для стовпця *Посадовий оклад* у полі *Умова вибірки* вводимо умову – *Between 1000 And 1500* (як приклад). Закриваємо запит із збереженням змін. У результаті отримуємо (рис. 5.52).

бельний	Прізвище	Ім'я	По-батькові	Стать	Дата народ	Сімейний с	Кількість ді	Дата прийм	Посадовий о
2	Ваків	Дмитро	Васильович	чол	11.03.1989	неодруж	0	12.04.2009	1 500,00р.
6	Кожан	Леє	Маркович	чол	04.04.1990	одруж	3	05.05.2008	1 200,00р.
9	Пазлик	Віктор	Климович	чол	07.07.1970	одруж	1	05.08.1952	1 500,00р.
10	Пронів	Олена	Василівна	жін	08.08.1960	одруж	4	09.12.1960	1 300,00р.

Рис. 5.52. Фрагмент результатів запиту за вибіркою за полем *Посадовий оклад*.

6. Створюємо запит для вибірки даних з сортуванням та умовою відбору за датою прийняття на роботу. Для цього переходимо у поле – *Запроси – Создание запроса с помощью мастера* - в якості вихідного елемента вибираємо таблицю *Основні дані* – додаємо всі доступні поля – *Далее - Далее* – вибираємо *Изменить макет запроса – Готово*. У полі *Сортировка* стовпця *Прізвище* вибираємо – *По возрастанию* – у полі *Условие отбора* стовпця *Дата прийняття на роботу* вводимо умову *Between #01.01.2007# And #01.01.2009#* –

Використання цього атрибуту робить комірки більшими. Між рамкою таблиці і даними у комірках завжди буде певна віддал.

Для всієї таблиці можна задати колір фону (тла):

`bgscolor="колір"`

Замість кольору допускається використання графічного зображення фону (тла):

`background="шлях до файлу фону"`

Атрибут `frame` (використовується тільки для елемента TABLE) дозволяє задавати вигляд рамки таблиці:

`frame="значення"`

Атрибут `frame` має такі стандартні значення:

`void` – рамка відсутня;

`above` – верхня сторона рамки;

`below` – нижня сторона рамки;

`hsides` – частини рамки зверху і знизу;

`vsides` – частини рамки зліва і справа;

`lhs` – ліва частина рамки;

`rhs` – права частина рамки;

`border` або `box` – рамка показана повністю.

Якщо атрибут `frame` не використовується – рамка навколо таблиці викреслюється повністю.

Атрибут `rules` визначає вигляд сітки всередині таблиці, тобто між сусідніми комірками. Для атрибуту `rules` використовуються такі значення:

`none` – сітка не відтворюється;

`groups` – сітка навколо групи комірок;

`rows` – горизонтальні лінії між рядками;

`cols` – вертикальні лінії між стовпцями;

`all` – звичайна сітка.

`<CAPTION>` `</caption>`

елемент для формування заголовку таблиці. Незважаючи на те, що цей елемент розміщується всередині елемента TABLE, заголовок відтворюється на екрані поза рамкою таблиці. За замовчуванням заголовок таблиці розміщується над таблицею і вирівнюється до центру таблиці. Положенням заголовку можна керувати використовуючи атрибут `align`:

`align="top"` – заголовок над таблицею;

`align="bottom"` – заголовок під таблицею;

`align="left"` – заголовок вверху і вирівняний до лівого краю;

`align="right"` – заголовок вверху і вирівняний до правого краю.

`<TC>` `</tc>`

елемент, призначений для формування заголовку таблиці, який буде розміщений над таблицею і вирівняний до лівого краю.

Для вирівнювання даних у комірках таблиці у горизонтальному напрямі призначений атрибут `align`, який може набувати значень `left`, `center`, `right`.

Атрибут `valign` надає можливість вирівнювати дані у вертикальному

7.8. Таблиці

Таблиці є зручним засобом форматування і публікування даних на Web-сторінках. Окрім того, таблиці дають можливість урізноманітнювати дизайн сторінки: вирівнювати між собою частини сторінки, розміщувати поруч рисунки і текст, керувати кольоровим оформленням тощо.

Для формування таблиці використовується принцип вкладеності елементів. Всередині основного елемента таблиці (TABLE) використовується ряд елементів, які визначають рядки (TR), а всередині цих елементів використовуються елементи для опису кожної комірки у рядку (TD,TH).

При аналізі структури існуючої таблиці або при створенні нової необхідно пам'ятати, що послідовність елементів описує таблицю зверху вниз і справа наліво. Наприклад, якщо після елемента TABLE поміщено елемент TR, це означає, що починається опис нового рядка таблиці. Усе, що розміщено за цим елементом буде подано в одному рядку (справа наліво). Це може бути послідовність елементів TD (комірок) або інша таблиця. Після того як зустріється наступний елемент TR, починається опис наступного рядка і так до кінця таблиці.

```
< TABLE > </table>
```

зовнішній елемент таблиці. Він визначає загальні властивості таблиці і відокремлює структуру таблиці від решти частин Web-сторінки. Елемент TABLE допускає використання ряду атрибутів. Таблицю можна вирівнювати у горизонтальному напрямі використовуючи атрибут align:

align="left" – до лівого краю;

align="center" – до центру;

align="right" – до правого краю.

Ширину таблиці можна задати точно у пікселях або у відсотковому відношенні до ширини сторінки у вікні броузера:

width=ширина таблиці у пікселях

width=ширина таблиці у відсотках%

Для керування виглядом рамки використовуються два атрибути. Це пов'язано з тим, що браузер створює зображення рамки імітуючи її тривимірність різницею в освітленості граней. На рамці розрізняється фронтальна і бокова похила грані. Ширину бокової грані задає атрибут border=ширина

Зауважимо, що задавши значення атрибуту border, яке дорівнює нулю, рамка пропадає зовсім.

Ширина фронтальної грані задається атрибутом cellpadding=ширина

Якщо значення цього атрибуту дорівнює нулю, тоді рамка виходить тонкою, загостреною.

Для усіх комірок таблиці можна задати розмір порожнього простору, який охоплює дані у комірках в пікселях або у відсотках:

cellpadding=кількість пікселів

(як приклад). Закриваємо запит із збереженням змін. У результаті отримуємо (рис. 5.53).

Ідентифікатор	Прізвище	Ім'я	По-батьков	Стать	Дата народ	Сімейний с	Кількість ді	Дата прийн
8	Двигун	Сергій	Борисович	чол	12.04.1990	одруж	1	09.09.2008
5	Залужна	Христина	Семенівна	жін	03.03.1990	одруж	1	04.04.2008
6	Кожан	Лев	Маркович	чол	04.04.1990	одруж	3	05.05.2008
7	Мацігін	Григорій	Петрович	чол	05.05.1985	неодруж	0	06.07.2007

Рис. 5.53. Фрагмент результатів запиту за вибіркою за полем *Дата прийняття на роботу*.

7. Створюємо звіт на основі побудованої бази. Для цього вибираємо об'єкт Звіти – *Створення звіту за допомогою Майстра*, додаємо всі доступні поля – *Далі – Готово*. У результаті отримуємо (рис. 5.54).

Основні дані

Прізвище	<input type="text" value="Бачинський"/>
Табельний номер	<input type="text" value="1"/>
Ім'я	<input type="text" value="Назар"/>
Стать	<input type="text" value="чол"/>
Сімейний стан	<input type="text" value="одруж"/>
Дата прийняття на	<input type="text" value="12.12.1987"/>
Підрозділ	<input type="text" value="плановий"/>
Прізвище	<input type="text" value="Ваків"/>
Табельний номер	<input type="text" value="2"/>
Ім'я	<input type="text" value="Дмитро"/>
Стать	<input type="text" value="чол"/>
Сімейний стан	<input type="text" value="неодруж"/>
Дата прийняття на	<input type="text" value="12.04.2009"/>
Підрозділ	<input type="text" value="господарський"/>
Прізвище	<input type="text" value="Двигун"/>
Табельний номер	<input type="text" value="3"/>
Ім'я	<input type="text" value="Сергій"/>
Стать	<input type="text" value="чол"/>
Сімейний стан	<input type="text" value="одруж"/>
Дата прийняття на	<input type="text" value="09.09.2008"/>
Підрозділ	<input type="text" value="обчислювальний центр"/>

Рис. 5.54. Фрагмент звіту, створеного за таблицею *Основні дані*.

Завдання 2.

Створити базу даних **Навчальний заклад**, яка містить наступні поля: Номер з/п, Назва групи, Прізвище студента, Стать, Рік народження, Адреса, Поштовий індекс, Домашній телефон, Оцінка 1, Оцінка 2, Оцінка 3, Оцінка 4, Оцінка 5. На основі даної таблиці створити форму з наступними елементами керування: перший запис, останній запис, видалення запису. Створити запити за назвою групи, кількістю запитів у базі. Створити у формі поле, де буде обчислюватися середній бал студента, а також надрукувати звіт по базі даних. Хід роботи:

1. Створюємо базу даних **Навчальний заклад**, яка буде містити таблицю *Навчальний заклад*. Для цього використовуємо пункт *Створення таблиці* в режимі *Конструктора* на закладці *Таблиця*. Обов'язковими полями є:

- Номер з/п – тип даних *Лічильник*;
- Назва групи – тип даних *Текстовий*;
- Прізвище студента – тип даних *Текстовий*;
- Стать - тип даних *Текстовий*;
- Рік народження – тип даних *Дата\час*;
- Адреса – тип даних *Текстовий*;
- Поштовий індекс – тип даних *Числовий*;
- Домашній телефон – тип даних *Числовий*;
- Оцінка 1 – тип даних *Числовий*;
- Оцінка 2 – тип даних *Числовий*;
- Оцінка 3 – тип даних *Числовий*;
- Оцінка 4 – тип даних *Числовий*;
- Оцінка 5 – тип даних *Числовий*.

У результаті отримуємо (рис. 5.55).

Навчальний заклад : таблиця	
Имя поля	Тип данных
Назва групи	Текстовый
Номер з/п	Счетчик
Прізвище студента	Текстовый
Стать	Текстовый
Рік народження	Дата/время
Адреса	Текстовый
Поштовий індекс	Числовой
Домашній телефон	Числовой
Оцінка 1	Числовой
Оцінка 2	Числовой
Оцінка 3	Числовой
Оцінка 4	Числовой
Оцінка 5	Числовой

Рис. 5.55. Фрагмент вікна створення таблиці *Навчальний заклад* у режимі *Конструктора*.

```
</head>
<BODY bgcolor=#FFFFFF>
<H2 align="center">Складний список</h2>
<HR align="justify" size=4 color="red">
<H3 align="center">Використання елементів UL та OL</h3>
<HR align="justify" size=4 color="red">
<UL> <LI>Пункт 1
<OL> <LI>Пункт 1.1
<LI>Пункт 1.2 </ol>
<LI>Пункт 2
<OL> <LI>Пункт 2.1 <LI>Пункт 2.2
</ol>
<LI>Пункт 3
<OL> <LI>Пункт 3.1 <LI>Пункт 3.2
</ol>
</ul>
<HR align="justify" size=4 color=#000000>
</body>
</html>
```

У даному прикладі нумеровані списки є вкладеними у пункти нумерованого списку.

На рис. 7.5 видно, як браузер Internet Explorer відтворює цей HTML-код.

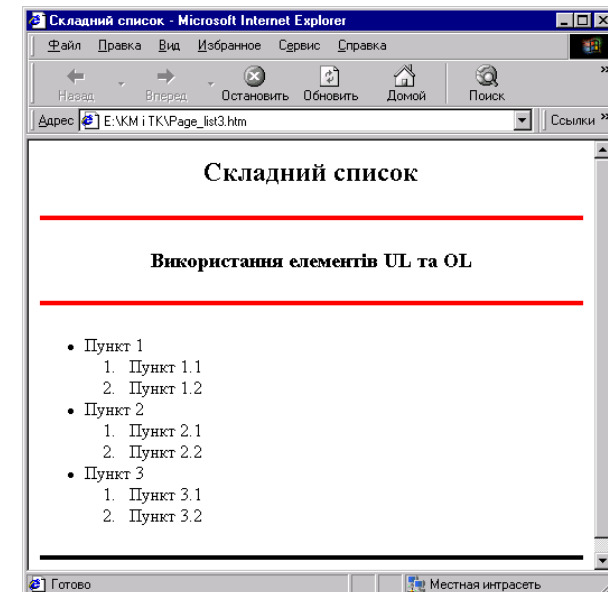


Рис. 7.5. Відтворений у вікні браузера Internet Explorer HTML-код використання елементів UL та OL.

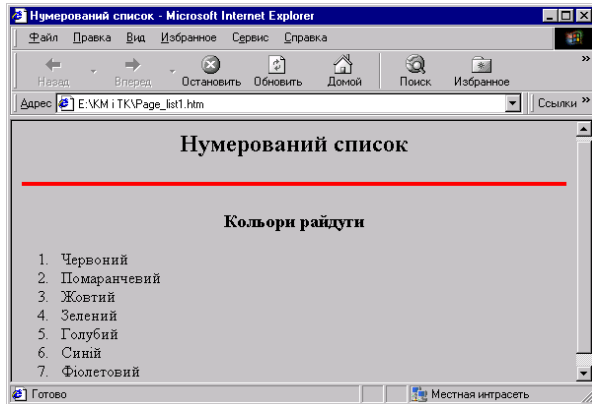


Рис. 7.4. HTML-код формування простого нумерованого списку, який відтворений у вікні браузера Internet Explorer.

```
<DL><DT><DD></dl>
списки з визначенням (Definition List).
Наприклад (фрагмент HTML-коду):
<DL>
<DT> Пункт 1
<DD> Визначення пункту 1
<DD> Наступне визначення пункту 1
<DT> Пункт 2
<DD> Визначення пункту 2
<DT> Пункт 3
<DD> Визначення пункту 3
</dl>
```

Кожен пункт списку може бути доповнений одним або декількома блоками тексту. Кожен блок автоматично розміщується з нового рядка. Термін "визначення" означає, що абзаци, розташовані у списку, можуть бути визначеннями, доповненнями, тлумаченнями пунктів. За своєю суттю пункт є заголовком, а визначення – довільним текстом тлумачення заголовку.

Інший спосіб створення списків полягає у використанні принципу вкладеності елементів. Кожний елемент, який визначає пункт списку, може містити у собі ще один список.

Вкладений список розміщується з невеликим відступом вправо у відношенні до списку верхнього рівня. Кожний із списків може бути створений і відлагоджений окремо, а пізніше усі складові можна об'єднати в один складний список.

```
Наприклад:
<HTML>
<HEAD>
<TITLE>Складний список</title>
```

Закриваємо режим *Конструктора* та відкриваємо таблицю у режимі вводу даних. Вводимо 10 довільних записів (рис. 5.56).

Назва групи	Номер з/п	Прізвище студента	Стать	Рік народження	Адреса	Поштовий індекс	Домашній телефон	Оцінка 1
A-11	1	Василів	чол	01.01.1993	м. Львів	39000	0	3
A-11	2	Горбачук	чол	01.02.1993	м. Львів	39000	1212120	4
A-11	3	Маланок	чол	01.03.1993	м. Стрий	24440	2323230	3
A-11	4	Петренко	чол	02.04.1993	м. Київ	56660	3433430	5
A-11	5	Іванчук	чол	03.05.1993	м. Полтава	54440	4545450	2
A-11	6	Татарко	жін	12.05.1993		67860	5656560	4
A-11	7	Бартків	жін	23.06.1993	м. Львів	89760	6767670	5
A-11	8	Дорош	жін	22.09.1992	м. Львів	67890	7878780	3
A-11	9	Гаркава	жін	14.12.1992	м. Київ	34560	8989890	4
A-11	10	Топорін	чол	22.12.1992	м. Хмельницьк	23450	9090900	3
A-11	11	Бойко	чол	23.08.1993	м. Одеса	13130	9898980	5
A-11	12	Лиско	жін	11.07.1993	м. Полтава	58580	7676760	3
A-11	13	Марків	чол	01.04.1993	м. Донецьк	67670	8787870	4
A-11	14	Торунь	жін	02.06.1993	м. Золочів	84840	6565650	4
A-11	15	Глинський	чол	03.07.1993	м. Стрий	66880	5454540	3
A-12	16	Фаріон	жін	09.09.1993	м. Київ	78780	4343430	5
A-12	17	Булавін	чол	09.12.1993	м. Хмельницьк	89890	3232320	5

Рис. 5.56. Фрагмент створеної таблиці *Навчальний заклад*.

2. На основі таблиці *Навчальний заклад* створюємо за допомогою *Майстра* форму *Навчальний заклад* для вводу даних, яка буде містити наступні поля:

- Номер з/п;
- Назва групи;
- Прізвище студента;
- Стать;
- Рік народження;
- Адреса;
- Поштовий індекс;
- Домашній телефон;
- Оцінка 1;
- Оцінка 2;
- Оцінка 3;
- Оцінка 4;
- Оцінка 5;

У результаті, після виходу з режиму *Конструктора*, отримуємо наступну форму з записами (рис. 5.57).

Назва групи	Номер з/п	Прізвище студента	Стать
A-11	1	Василів	чол
A-11	2	Горбачук	чол
A-11	3	Маланок	чол
A-11	4	Петренко	чол
A-11	5	Іванчук	чол
A-11	6	Татарко	жін
A-11	7	Бартків	жін
A-11	8	Дорош	жін

Рис. 5.57. Фрагмент вікна форми *Навчальний заклад*.

3. Створюємо елементи керування формою. Для цього переходимо в режим *Конструктора* форми, за допомогою мишки змінюємо розміри *Області даних*, на панелі елементів вибираємо функційний елемент *Кнопка* та вставляємо його у область даних форми. Відкриється вікно створення кнопок (рис. 5.58).

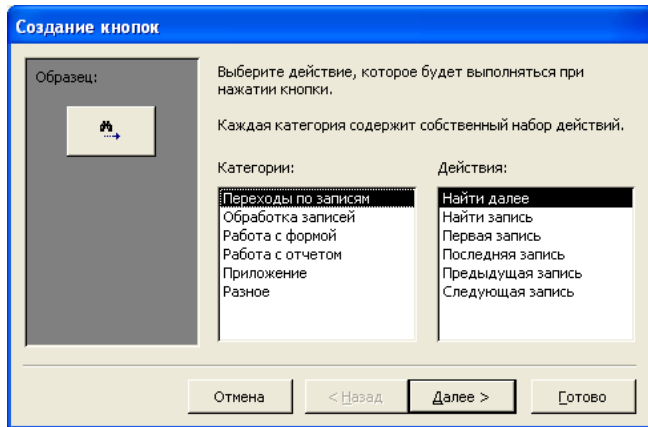


Рис. 5.58. Вигляд вікна створення кнопок.

У стовпці *Категорії* вибираємо пункт *Переходи по записях*, у стовпці *Дії* – *Перший запис* – *Готово*.

Повторюємо операцію для встановлення функційних кнопок *Останній запис* та *Видалити запис*. Результат виконання подано на рис. 5.59.

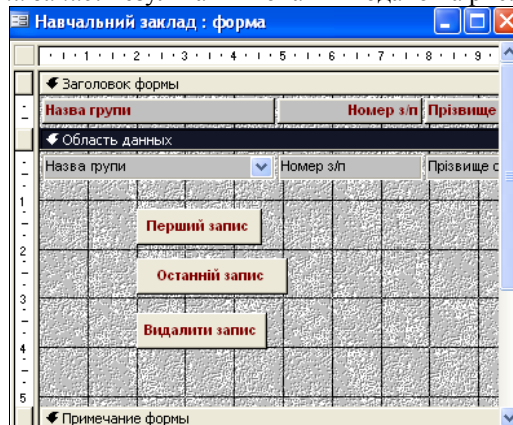


Рис. 5.59. Фрагмент форми *Навчальний заклад* з створеними кнопками.

4. Створюємо запити для вибірки інформації за конкретною групою. Для цього вибираємо об'єкт *Запити*, *Створення запиту* за допомогою *Майстра*. В якості вихідного елемента вибираємо таблицю *Навчальний*

Значення атрибуту type

Атрибут і його значення	Тип нумерації
Type="1"	1, 2, 3, 4, ...
Type="i"	i, ii, iii, iv, ...
Type="I"	I, II, III, IV, ...
Type="a"	a, b, c, d, ...
Type="A"	A, B, C, D, ...

Для задання початкового значення нумерації списку призначений атрибут start:

start=номер

Наприклад, якщо використовується нумерований список у вигляді букв, тоді значення атрибуту start=3 означає, що нумерація списку починається з символу C.

У нумерованих списках для елемента LI існує атрибут, який визначає номер для поточного пункту списків:

value=номер

Розглянемо HTML-код формування простого нумерованого списку:

```
<HTML>
<HEAD>
<TITLE>Нумерований список</title>
</head>
<BODY bgcolor=#C0C0C0>
<H2 align="center">Нумерований список</h2>
<HR align="justify" size=4 color="red">
<H3 align="center">Кольори райдуги</h3>
<OL> <LI>Червоний
<LI>Помаранчевий
<LI>Жовтий
<LI>Зелений
<LI>Голубий
<LI>Синій
<LI>Фіолетовий </ol>
</body>
</html>
```

HTML-код формування простого нумерованого списку, який відтворений у вікні броузера Internet Explorer подано на рис. 7.4.


```

<FONT color="maroon">
<H2 align="center">Маркований список</h2>
<HR align="justify" size=4 color="navy">
<H3 align="center">Кольори райдуги</h3>
<HR align="justify" size=3 color="red">
<UL> <LI>Червоний
<LI>Помаранчевий
<LI>Жовтий
<LI>Зелений
<LI>Голубий
<LI>Синій
<LI>Фіолетовий </ul>
</body>
</html>

```

Елемент UL є своєрідним обрамленням списку. Він дозволяє відділяти один список від іншого. Елемент LI позначає кожний пункт. HTML-код формування нумерованого списку, який відтворений у вікні браузера Internet Explorer подано на рис. 7.3.

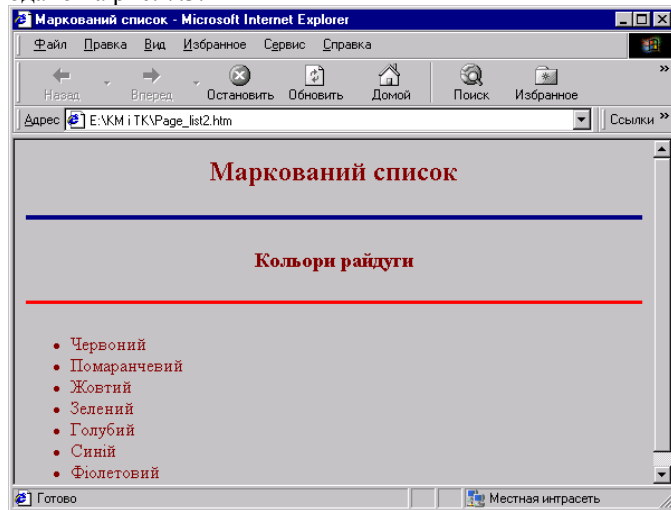


Рис. 7.3. HTML-код формування нумерованого списку, який відтворений у вікні браузера Internet Explorer.

```
<OL> <LI> </ol>
```

елемент формування нумерованого списку (Ordered List). Нумерований список обрамляється елементом OL, а кожний пункт позначається впорядкованою послідовністю символів: арабські або римські числа, букви латинського алфавіту. Спосіб нумерації визначається атрибутом type. Значення атрибуту type наведені у табл. 7.2.

заклад, додаємо всі доступні поля, Далі – Далі, вибираємо Змінити макет запити – Готово. Для стовпця Назва групи у полі Умова вибору вводимо умову – A-11 (рис. 5.60).

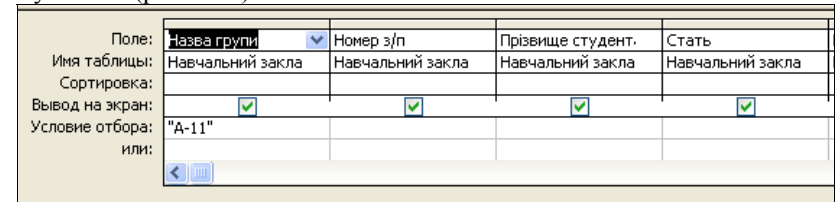


Рис. 5.60. Фрагмент заповненої таблиці Конструктора запити для вибору за кодом групи.

Закриваємо запит із збереженням змін. У режимі таблиці отримуємо (рис. 5.61).

Назва групи	Номер з/п	Прізвище студент.	Стать	Рік народження	Адреса	Поштовий індекс
A-11	1	Василів	чол	01.01.1993	м.Львів	39000
A-11	2	Горбачук	чол	01.02.1993	м.Львів	39000
A-11	3	Маланок	чол	01.03.1993	м.Стрий	24440
A-11	4	Петренко	чол	02.04.1993	м.Київ	56660
A-11	5	Іванчук	чол	03.05.1993	м.Полтава	54440
A-11	6	Татарко	жін	12.05.1993		67860
A-11	7	Бартків	жін	23.06.1993	м.Львів	89760
A-11	8	Дорош	жін	22.09.1992	м.Львів	67890
A-11	9	Гаркава	жін	14.12.1992	м.Київ	34560
A-11	10	Топорін	чол	22.12.1992	м.Хмельницьк	23450
A-11	11	Бойко	чол	23.08.1993	м.Одеса	13130

Рис. 5.61. Фрагмент результату виконання запити, поданого на рис. 3.60.

5. Створюємо запит для вибірки даних з підрахунком кількостей записів у таблиці. Для цього вибираємо об'єкт Запити, Створення запити за допомогою Майстра. У якості вихідного елемента вибираємо таблицю Навчальний заклад, додаємо всі доступні поля, Далі – Далі, вибираємо Змінити макет запити – Готово. На панелі інструментів активуємо кнопку Підсумки, у результаті додається ще один рядок – Групова операція. У всіх полях даного рядка вибираємо функцію Count (рис. 5.62).

Поле:	Стать	Рік народження	Адреса	Поштовий індекс
Ім'я таблиці:	Запрос_група	Запрос_група	Запрос_група	Запрос_група
Груповая операція:	Count	Count	Count	Count
Сортировка:				
Вивод на екран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Условие отбора:				
или:				

Рис. 5.62. Фрагмент заповненої таблиці Конструктора запити для визначення кількості записів в таблиці.

Закриваємо запит із збереженням змін. У режимі таблиці отримуємо (рис. 5.63).

Count-Назва гр	Count-Номер з	Count-Прізвищ	Count-Стать	Count-Рік наро	Count-Адреса
15	15	15	15	15	14

Рис. 5.63. Фрагмент результату запиту поданого на рис. 3.62.

У початковій таблиці, під час створення бази даних, не всі поля можуть бути заповненими, тому під час підрахунку кількості записів в базі бажано задавати групову операцію *Count* одночасно за декількома полями.

6. Відкриваємо форму для вводу обчислювального поля *Середній бал*. Додаємо поле *Середня успішність*. Відкриваємо властивість даного поля, вводимо умову $=([Оцінка\ 1]+[Оцінка\ 2]+[Оцінка\ 3]+[Оцінка\ 4]+[Оцінка\ 5])/5$ (рис. 5.64).

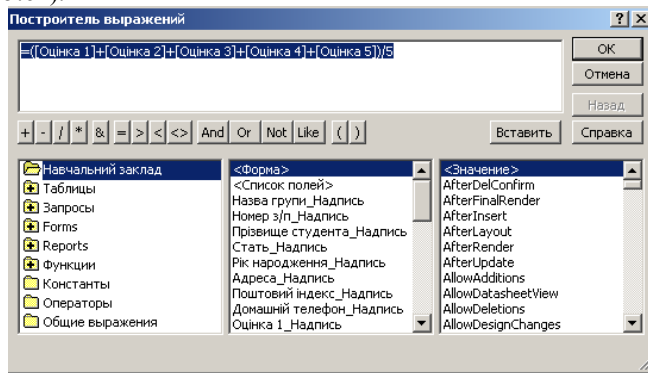


Рис. 5.64. Використання *Побудовувача виразів* для обчислення середнього балу студента.

Закриваємо *Конструктор* форми із збереженням. У режимі форми отримаємо результат (рис. 5.65).

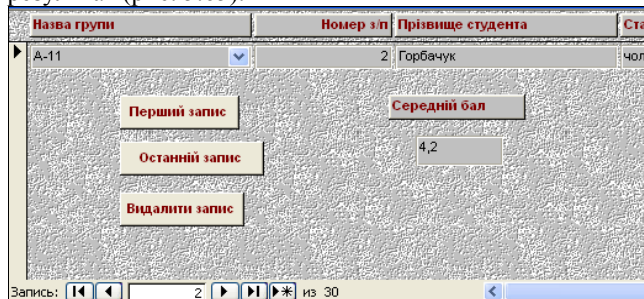


Рис. 5.65. Фрагмент форми з середнім балом для другого запису.

7. Створюємо звіт на основі побудованої бази. Для цього вибираємо об'єкт Звіти – *Створення звіту за допомогою Майстра*, додаємо всі доступні поля – *Далі* – *Готово* (рис. 5.66).

елемент визначення типу, розміру і кольору шрифтів. Усі параметри шрифтів задаються значеннями відповідних атрибутів: *size*, *color*, *face*. Абсолютний розмір шрифту задається значенням атрибуту *size*.

``

Абсолютні значення атрибуту *size* змінюється в межах від 1 до 7. Розмір шрифту (збільшення або зменшення) можна задати відносно до базового:

`size=+число`

`size=- число`

При визначенні значення атрибуту треба враховувати базовий розмір шрифту. Сумарне значення (сума "+число" або "- число" і базового значення) не повинно перевищувати одного з абсолютних значень.

Тому, для базового розміру, який дорівнює 3, відносний розмір може набувати значення у межах від -2 до +4 (найменше абсолютне значення може дорівнювати 1, найбільше – не перевищує 7).

Атрибут *face* дозволяє задавати визначений шрифт або декілька шрифтів, назви яких відділяються символом ";", наприклад:

``

Використання атрибуту *face* пояснюється тим, що Web-сторінки переглядає багато користувачів і немає гарантії, що у кожного з них наявний той або інший тип шрифту. Якщо у обчислювальній системі не встановлений потрібний шрифт, тоді браузер використовує стандартний шрифт, який встановлюється за замовчуванням.

7.7. Списки

Списки (list) відрізняються від звичайного тексту перш за все тим, що користувачам нема потреби думати про нумерацію пунктів списку. Якщо список доповнюється новими пунктами або скорочується, нумерація коректується автоматично. У випадку нумерованих (маркованих) списків програма ставить перед кожним пунктом маркери: крапки, прямокутники, ромби або інші символи. Елементи, призначені для створення списків, умовно поділяються на дві групи: перші визначають загальний вигляд списку, а другі – задають його внутрішню структуру. У списках використовуються стандартні атрибути.

` `

елемент найпростішого, нумерованого (маркованого) списку (Unordered List).

Розглянемо HTML-код формування нумерованого списку на наступному прикладі:

`<HTML>`

`<HEAD>`

`<TITLE>Маркований список</title> </head>`

`<BODY bgcolor=#C0C0C0>`

Таблиця 7.1

Назви і RGB-коди кольорів

Українська назва	Англійська назва	RGB-код
Аквамарин	Aqua	#00FFFF
Білий	White	#FFFFFF
Жовтий	Yellow	#FFFF00
Зелений	Green	#008000
Золотий	Gold	#FFD700
Індиго	Indigo	#480080
Каштановий	Maroon	#800000
Червоний	Red	#FF0000
Оливковий	Olive	#808000
Помаранчевий	Orange	#FFA500
Яскраво-червоний	Purple	#800080
Яскраво-зелений	Lime	#00FF00
Сріблястий	Silver	#C0C0C0
Сірий	Gray	#808080
Сизий	Teal	#008080
Синій	Blue	#0000FF
Ультрамарин	Navy	#000080
Фіолетовий	Violet	#EE80EE
Фуксиновий	Fuchsia	#FF00FF
Чорний	Black	#000000

Не можна стверджувати, що генерований код кольору у різних програмних кодах буде сумісним. Створивши деякий відтінок кольору в одній програмі і використавши його RGB-код у іншій, Ви не обов'язково отримаєте той самий відтінок.

Відтінки кольорів, створені у RGB-кодах, найкраще перевіряти, власне, на самих Web-сторінках.

7.6. Шрифти

Для визначення параметрів шрифту (типу, розміру, кольору) в HTML існують такі елементи.

<BASEFONT>

елемент, який визначає базовий (головний для усієї сторінки) розмір шрифту.

Всередині елемента необхідно записати атрибут size.

<BASEFONT size=базовий розмір шрифту>

Значення для атрибуту size змінюються у межах від 1 до 7. За замовчуванням використовується значення 3. Інші атрибути цього елемента такі ж, як і для елемента FONT.

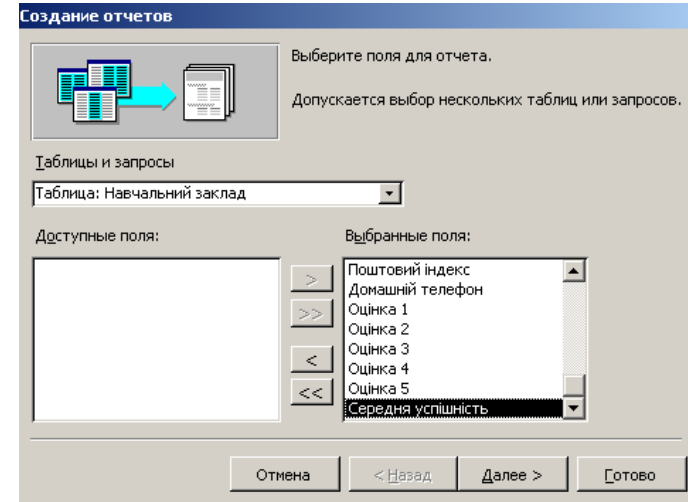


Рис. 5.66. Вікно Майстра створення звіту.

У результаті отримуємо (рис. 5.67).

Навчальний заклад	
Назва групи	A-11
Номер з/п	1
Прізвище студент	Василів
Стать	чол
Рік народження	01.01.1993
Адреса	м.Львів
Поштовий індекс	39000
Домашній телефо	0
Оцінка 1	3
Оцінка 2	3
Оцінка 3	4
Оцінка 4	4
Оцінка 5	4
Середня успішність	

Рис. 5.67. Фрагмент звіту на основі створеної бази даних **Навчальний заклад**.

Завдання 3.

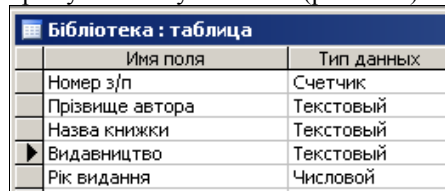
Створити базу даних **Бібліотека**, що буде містити таблицю *Бібліотека*, у котру входять наступні поля: Номер з/п, Прізвище автора, Назва книжки, Видавництво, Рік видання. На основі таблиці *Бібліотека* створити форму *Бібліотека для вводу даних*. У формі створити кнопки керування для переходу на перший запис, останній запис та друк форми. Створити запит на вибірку за автором, посортувати за роком видання, автором та назвою книжки. Створити звіт по базі даних.

Хід роботи:

1. Створюємо базу даних **Бібліотека**, яка буде містити таблицю *Бібліотека*. Для цього використовуємо пункт *Створення таблиці в режимі Конструктора* на закладці *Таблиця*. Обов'язковими полями є:

- Номер з/п – тип даних *Лічильник*;
- Прізвище автора – тип даних *Текстовий*;
- Назва книжки – тип даних *Текстовий*;
- Видавництво – тип даних *Текстовий*;
- Рік видання – тип даних *Числовий*.

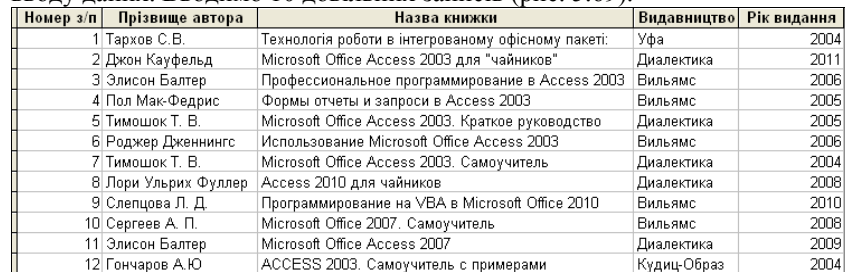
У результаті отримуємо наступне вікно (рис. 5.68).



Бібліотека : таблиця		
	Имя поля	Тип данных
	Номер з/п	Счетчик
	Прізвище автора	Текстовый
	Назва книжки	Текстовый
	Видавництво	Текстовый
	Рік видання	Числовой

Рис. 5.68. Таблиця *Бібліотека* у режимі *Конструктора*.

Закриваємо режим *Конструктора* та відкриваємо таблицю у режимі вводу даних. Вводимо 10 довільних записів (рис. 5.69).



Номер з/п	Прізвище автора	Назва книжки	Видавництво	Рік видання
1	Тархов С. В.	Технологія роботи в інтегрованому офісному пакеті:	Уфа	2004
2	Джон Кауфелд	Microsoft Office Access 2003 для "чайників"	Диалектика	2011
3	Элисон Балтер	Профессиональное программирование в Access 2003	Вильямс	2006
4	Пол Мак-Федрис	Формы отчеты и запросы в Access 2003	Вильямс	2005
5	Тимошок Т. В.	Microsoft Office Access 2003. Краткое руководство	Диалектика	2005
6	Роджер Дженнингс	Использование Microsoft Office Access 2003	Вильямс	2006
7	Тимошок Т. В.	Microsoft Office Access 2003. Самоучитель	Диалектика	2004
8	Лори Ульрих Фуллер	Access 2010 для чайников	Диалектика	2008
9	Слепцова Л. Д.	Программирование на VBA в Microsoft Office 2010	Вильямс	2010
10	Сергеев А. П.	Microsoft Office 2007. Самоучитель	Вильямс	2008
11	Элисон Балтер	Microsoft Office Access 2007	Диалектика	2009
12	Гончаров А. Ю.	ACCESS 2003. Самоучитель с примерами	Кудиц-Образ	2004

Рис. 5.69. Фрагмент створеної таблиці *Бібліотека*.

2. На основі таблиці *Бібліотека* створюємо за допомогою *Майстра* (рис. 5.70) форму *Бібліотека* для вводу даних, яка буде містити наступні поля:

- Номер з/п;
- Прізвище автора;

```
<H1 align="center">Заголовок 1, елемент Н1 </h1>
<H2 align="right">Заголовок 2, елемент Н2 </h2>
<H3>Заголовок 3, елемент Н3 </h3>
<H4>Заголовок 4, елемент Н4 </h4>
<H5>Заголовок 5, елемент Н5 </h5>
<H6>Заголовок 6, елемент Н6 </h6>
<HR align="justify" size=4 color="green">
</body>
</html>
```

У якому вигляді браузер Internet Explorer відтворює цей HTML-код подано на рис. 7.2.

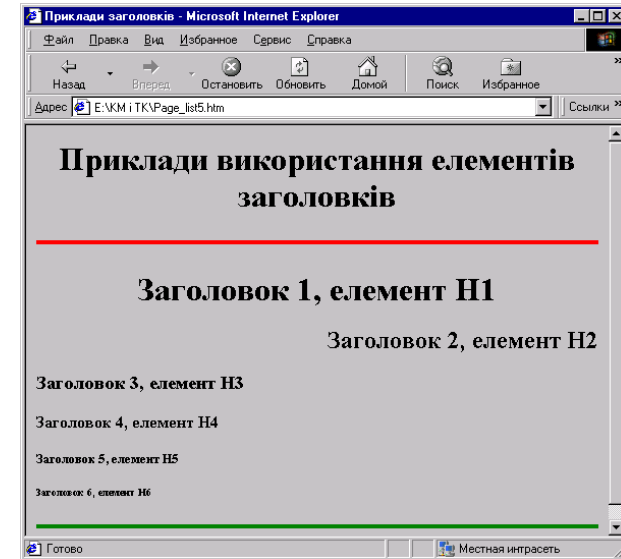


Рис. 7.2. HTML-код використання заголовків відтворений у вікні браузера Internet Explorer.

7.5. Управління кольором

Кольори використовують для подання фону (тла), шрифтів, горизонтальних ліній та інших елементів Web-сторінки. Кольори позначаються англійськими назвами або числовими шістнадцятковими кодами. Існує декілька атрибутів, значеннями яких є параметри кольорів. Найпростіший спосіб вибрати колір – написати його назву англійською мовою, як значення атрибуту color color="red"

У табл. 7.1. подані уживані назви кольорів. Але, у загальному випадку, колір визначається так званим RGB-кодом. Довільний колір визначається комбінацією червоного (R), зеленого (G) та синього (B) кольорів у певній пропорції.

зручно розміщувати у кінці рядка. Він не тільки позначить кінець абзацу, але і виконає функцію елемента
 (примусовий перехід на новий рядок).

Елемент абзацу допускає використання атрибуту align:

- <P align="left"> – вирівнювання до лівого краю;
- <P align="center"> – вирівнювання до центру;
- <P align="right"> – вирівнювання до правого краю.

елемент примусового переходу на новий рядок. Він має тільки початковий дескриптор. У місці його розміщення рядок закінчується, а решта тексту переходить у наступний рядок.

<HR>

елемент формування горизонтальної лінії (horizontal rule). Цей елемент є дуже популярним. По-перше, з його допомогою зручно ділити сторінку на частини. По-друге, запропонований розробникам Web-сторінок вибір подібних елементів є дуже обмежений.

Елемент немає кінцевого дескриптора, але допускає використання атрибутів: align, size, color. Для вирівнювання до лівого краю, до центру, до правого краю, за шириною:

<HR align="left">

<HR align="center">

<HR align="right">

<HR align="justify">

Задання товщини лінії

<HR size=товщина у пікселях>

Задання довжини лінії

<HR width=довжина у пікселях>

<HR width=довжина у відсотках %>

Задання кольору лінії

<HR color="колір">.

Для створення заголовків у HTML наявні шість рівнів заголовків, кожному з яких відповідає свій розмір шрифту. Використовуючи заголовки різних рівнів можна створити чітку ієрархічну структуру документу.

<Hn> </hn>

елемент заголовку. Тут n – рівень заголовку: 1 – найбільший, 6 – найменший. Для вирівнювання заголовків використовується атрибут align. Використання заголовків проілюструємо таким HTML-кодом.

<HTML>

<HEAD>

<TITLE>Приклади заголовків</title>

</head>

<BODY bgcolor="silver">

<H1 align="center">Приклади використання елементів заголовків</h1>

<HR align="justify" size=4 color="red">

- Назва книжки;
- Видавництво;
- Рік видання.

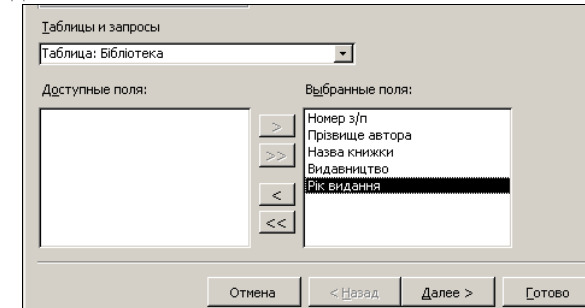


Рис. 5.70. Вікно *Конструктора* форм під час створення форми *Бібліотека*.

У результаті, після виходу з режиму *Конструктора*, отримуємо наступну форму з записами (рис. 5.71).

Номер з/п	Прізвище автора	Назва книжки	Видавництво	к видання
1	Тарков С. В.	Технологии работы в сети	Уфа	2004
2	Джон Кауфелд	Microsoft Office Access 2	Диалектала	2011
3	Этсон Балгер	Профессиональное прог.	Випльяс	2006
4	Пол Мак-Федрис	Формы отчеты и запросы	Випльяс	2005
5	Тимошок Т. В.	Microsoft Office Access 2	Диалектала	2005
6	Роджер Джозонгис	Использование Microsoft	Випльяс	2006
7	Тимошок Т. В.	Microsoft Office Access 2	Диалектала	2004
8	Лори Уларки Фултер	Access 2010 для чайников	Диалектала	2008
9	Святцова Л. Д.	Программирование на V	Випльяс	2010
10	Сергеев А. П.	Microsoft Office 2007. С.	Випльяс	2008
11	Этсон Балгер	Microsoft Office Access 2	Диалектала	2009
12	Гончаров А Ю	ACCESS 2003. Самоучит	Кулиц-Образ	2004
*	(Счетчик)			0

Рис. 5.71. Фрагмент вікна форми *Бібліотека*.

3. Створюємо елементи керування формою. Для цього переходимо в режим *Конструктора форми*, за допомогою мишки змінюємо розміри *Області даних*, на панелі елементів вибираємо функційний елемент *Кнопка* та вставляємо його в область даних форми. Відкриється вікно створення кнопок (рис. 5.72).

У стовпці *Категорії* вибираємо пункт *Переходи по записам*, у стовпці *Дії* – *Перший запис* – *Готово*.

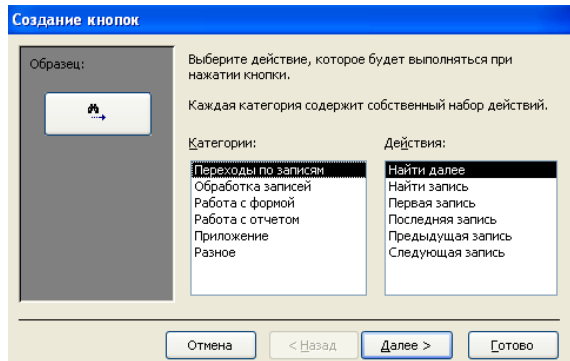


Рис. 5.72. Видяк вiкна створення кнопок.

Повторюємо операцію для встановлення функційних кнопок *Останній запис* та *Видалити запис*. Результат виконання подано на рис. 5.73.

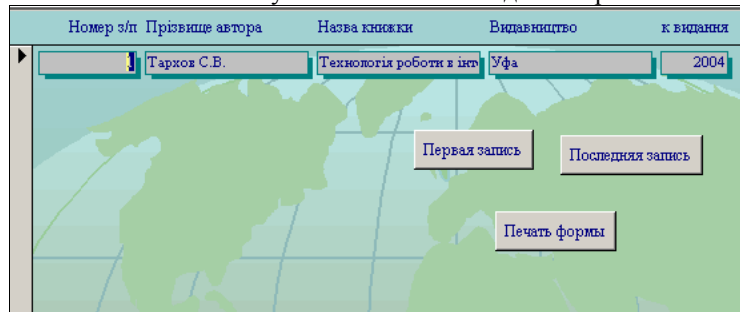


Рис. 5.73. Фрагмент заповненої форма *Бібліотека* зі створеними кнопками керування записами.

4. Налаштуємо параметри запуску бази даних, *Сервіс – Параметри запуску* – у полі *Вивід форми* вибираємо за основу форму *Бібліотека – ОК* (рис. 5.74).

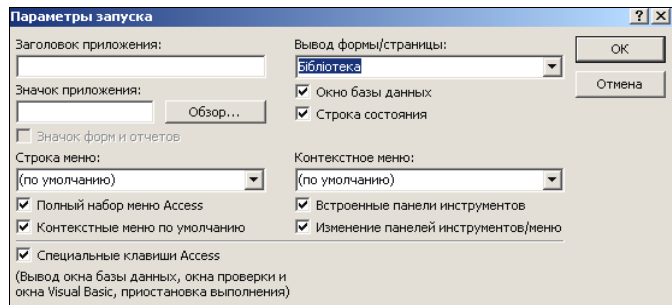


Рис. 5.74. Вікно налагодження параметрів запуску бази даних.

На відміну від EM елемент I, який визначає подання тексту у вигляді курсиву, інтерпретується браузером однозначно.

Форматування тексту розглянемо на такому прикладі.

```
<HTML>
<HEAD>
<TITLE>Приклад форматування тексту і символів</title>
</head>
<BODY bgcolor="silver">
<H1 align="center">Приклад форматування тексту і символів</h1>
<HR align="justify" size=4 color="red">
Подання тексту <B>піввідзначеним шрифтом </b>. <BR>
Подання тексту <I> курсивом</i>. <BR>
Створення ефекту <SUP>надрядкових</sup> символів. <BR>
Створення ефекту <SUB>підрядкових</sub> символів. <BR>
Подання тексту <U>підкресленням</u>. <BR>
Подання тексту <S>закресленням</s>. <BR>
<HR align="justify" size=4 color="green">
</body>
</html>
```

У якому вигляді браузер Internet Explorer відтворює цей HTML-код подано на рис. 7.1.

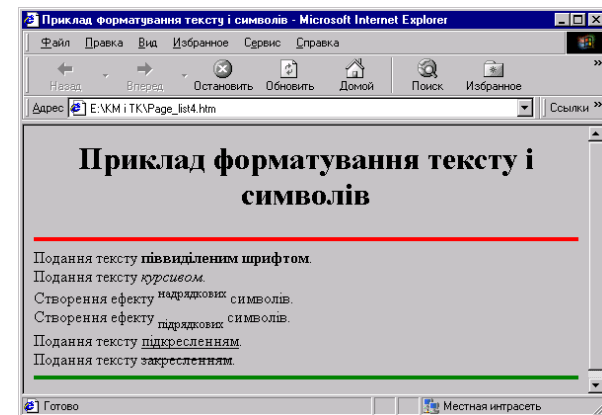


Рис. 7.1. HTML-код відтворений у вікні браузера Internet Explorer.

7.4. Розміщення тексту і заголовки

```
<P> </p>
```

елемент абзацу (paragraph). Допускається використання тільки початкового дескриптора, оскільки наступний <P> означає не тільки початок наступного абзацу, але і закінчення попереднього. Інколи початковий дескриптор

7. Створюємо запит для вибірки даних за автором та назвою книжки. Для цього переходимо у поле *Запити, Створення запиту за допомогою Майстра*. У якості вихідного елемента вибираємо таблицю *Замовлення*, додаємо всі доступні поля, *Далі – Далі*, вибираємо *Змінити макет запиту – Готово*. У полі *Прізвище автора*, рядку *Умова вибору* вводимо умову "Тимошок*" (як приклад). У полі *Назва книжки*, стрічці *Умова вибору* вводимо умову Microsoft Office Access 2003. Самоучитель (рис. 5.79).

Поле:	Номер з/п	Прізвище автора	Назва книжки	Видавництво	Рік видання
Имя таблицы:	Бібліотека	Бібліотека	Бібліотека	Бібліотека	Бібліотека
Сортировка:					
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Условие отбора:		Like "Тимошок*"	"Microsoft Office Acces		
или:					

Рис. 5.79. Фрагмент заповненої таблиці *Конструктора* запиту для вибору за прізвищем автора та назвою книги.

Закриваємо запит із збереженням змін. У режимі таблиці отримуємо (рис. 5.80).

Бібліотека Запрос : запрос на выборку		
Номер з/п	Прізвище автора	Назва книжки
1	Тимошок Т. В.	Microsoft Office Access 2003. Самоучитель
*	(Счетчик)	

Рис. 5.80. Фрагмент результату виконання запиту поданого на рис. 5.79.

8. Створюємо звіт на основі побудованої бази. Для цього вибираємо об'єкт *Звіти – Створення звіту за допомогою Майстра*, додаємо всі доступні поля, *Далі – Готово*. У результаті отримуємо (рис. 5.81).

Бібліотека				
Номер з/п	Прізвище автора	Назва книжки	Видавництво	Рік видання
1	Тархов С.В.	Технология работы в ин	Уфа	2004
2	Джон Кауфельд	Microsoft Office Access	Диалектика	2011
3	Элисон Балтер	Профессиональное пр	Вильямс	2006
4	Пол Мак-Федрис	Формы отчеты и запро	Вильямс	2005
5	Тимошок Т. В.	Microsoft Office Access	Диалектика	2005
6	Роджер Дженнингс	Использование Micros	Вильямс	2006
7	Тимошок Т. В.	Microsoft Office Access	Диалектика	2004
8	Лори Ульрих Фуллер	Access 2010 для чайн	Диалектика	2008
9	Слепцова Л. Д.	Программирование на	Вильямс	2010
10	Сергеев А. П.	Microsoft Office 2007.	Вильямс	2008
11	Элисон Балтер	Microsoft Office Access	Диалектика	2009
12	Гончаров А.Ю	ACCESS 2003. Самоуч	Кудиц-Образ	2004

Рис. 5.81. Фрагмент звіту на основі створеної бази даних *Бібліотека*.

<TITLE> </title>

елемент, призначений для розміщення заголовку Web-сторінки.

Заголовок повинен бути простим і коротким. Одночасно заголовок має нести інформаційне навантаження про призначення і логічний зміст Web-сторінки. При формуванні заголовків Web-документів дотримуйтесь наступних рекомендацій:

- уникайте загальних фраз;
- уникайте лозунгів;
- використовуйте для заголовків не більше, як 64 символи.

<META>

елемент, призначений для розміщення службової інформації про Web-сторінку, яка не відтворюється при її перегляді. В середині цього елемента немає тексту у звичному розумінні, тому і немає кінцевого дескриптора. Кожний елемент META містить два основних атрибути, перший з яких визначає тип даних, а другий – зміст.

Наприклад:

- name="Expires" content="Дата" (дата, яка визначає "термін дії" документу);
- name="reply-to" content="ім'я@Адреса" (адреса електронної пошти);
- name="Author" content="ім'я автора" (ім'я автора Web-сторінки);
- name="Keywords" content="Слово 1, Слово 2, Слово 3 ..." (набір ключових слів для пошуку);
- name="Generator" content="Назва HTML-редактора" (назва документу, в якому була створена Web-сторінка).

До елемента HEAD можна внести декілька елементів <META>, щоб перерахувати ключові слова, внести ім'я автора та іншу інформацію про Web-сторінку.

<BODY> </body>

елемент, який окреслює основну частину Web-сторінки, власне, саму Web-сторінку.

Між його початковим і кінцевим дескрипторами розміщується відтворена частина Web-сторінки: текст, зображення, мультимедіа і посилання на інші документи. Це – та довільна частина, яка створюється автором і відтворюється браузером. В середині елемента BODY розміщуються усі елементи, призначені для формування структури, вмісту та дизайну Web-сторінки. У початковому дескрипторі можна використовувати ряд атрибутів, які визначають параметри для всієї сторінки. Розглянемо деякі з них:

- background="Шлях до файлу фону" атрибут, який визначає фон (тло) сторінки;
- bgcolor="#RRGGBB" атрибуту простого оформлення фону (тла) заданням кольору;

<BODY>

У цьому місці необхідно розмістити оригінальний текст Web-сторінки, списки, таблиці, графічні об'єкти, посилання на інші Web-сторінки і багато іншого.

</body>

</html>

Якщо проаналізувати HTML-коди різноманітних Web-документів, можна відразу звернути увагу на подібність їх структур. Це пояснюється тим, що документи створюються за певними правилами. У синтаксичну основу мови HTML закладено офіційний стандарт ISO 8879:1986 "Information processing. Text and office systems. Standard Generalized Markup Language (SGML)".

Для розуміння структури Web-сторінки розглянемо елементи, які подані у HTML-кодї. Вивчаючи елементи будемо вказувати початковий і кінцевий дескриптори. Кількість випадків, коли допустимим є тільки початковий дескриптор, невелика і про такі випадки будемо говорити окремо. Для імен дескрипторів можна використовувати як великі, так і малі символи латинського алфавіту. Будемо дотримуватися такого підходу: імена початкових дескрипторів будемо записувати великими буквами, а кінцевих – малими. Це значною мірою допоможе Вам аналізувати та редагувати HTML-код Web-сторінки.

<HTML > </html >

елемент позначення документу мовою HTML.

Вище згадувалось про те, що одним із фундаментальних принципів HTML є багаторівневе вкладення елементів. Даний елемент має найвищий рівень вкладеності, так як між його початковим і кінцевим дескрипторами буде розміщений HTML-код всієї Web-сторінки. Взагалі, цей елемент можна трактувати як обов'язкову формальність. Елемент допускає вкладеність елементів HEAD, BODY, FRAMESET, які визначають загальну структуру Web-сторінки.

<HEAD> </head>

елемент області заголовку Web-сторінки. Як і попередній, цей елемент призначений для формування загальної структури документу. Хоча вміст елементу HEAD не відтворюється у тексті Web-сторінки, однак один його важливий компонент, елемент TITLE, відтворюється у вікні броузера при отриманні доступу до Web-сторінки.

Елемент HEAD допускає можливість вкладення ряду інших елементів, зокрема:

- TITLE (визначає назву документу);
- META (передає броузеру додаткову інформацію про документ);
- BASE (задає базову адресу документу);
- STYLE (визначає зовнішній вигляд документу);
- SCRIPT (визначає оператори інтерпретованих мов програмування, таких як JavaScript, VBScript).

З усіх перерахованих елементів обов'язковим є тільки TITLE.

Завдання 4.

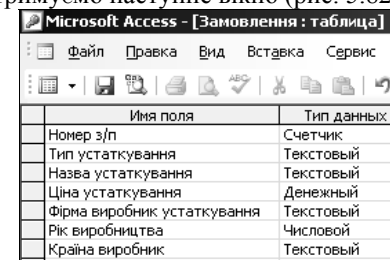
Створити базу даних **Замовлення**, яка буде містити таблицю *Замовлення* з такими полями: Номер з/п, Назва устаткування, Назва виробництва, Ціна устаткування, Фірма виробник устаткування, Рік виробництва, Країна виробник. На основі таблиці *Замовлення* створити форму *Замовлення* для вводу даних. На формі створити елемент керування для друку форми. Налаштувати параметри запуску бази даних вибравши за основу форму *Замовлення*. Створити запит для вибірки інформації упорядкованого за Назвою устаткування. Створити запит для вибірки даних за країною виробником. Створити запит для вибірки даних за найнижчою ціною серед автомобілів Bentley. Створити звіт на основі побудованої бази.

Хід роботи:

1. Створюємо базу даних *Замовлення*, яка буде містити таблицю *Замовлення*. Для цього використовуємо пункт *Створення таблиці в режимі Конструктора* на закладці *Таблиця*. Обов'язковими полями є:

- Номер з/п - тип даних Счетчик;
- Тип устаткування – тип даних Текстовий;
- Назва устаткування - тип даних Текстовий;
- Ціна устаткування - тип даних Денежний;
- Фірма виробник устаткування - тип даних Текстовий;
- Рік виробництва - тип даних Числовий;
- Країна виробник - тип даних Текстовий.

У результаті отримуємо наступне вікно (рис. 5.82).



Имя поля	Тип данных
Номер з/п	Счетчик
Тип устаткування	Текстовый
Назва устаткування	Текстовый
Ціна устаткування	Денежный
Фірма виробник устаткування	Текстовый
Рік виробництва	Числовой
Країна виробник	Текстовый

Рис. 5.82. Таблиця *Замовлення* в режимі *Конструктора*

Закриваємо режим *Конструктора* та відкриваємо таблицю в режимі вводу даних. Вводимо 10 довільних записів.

2. На основі таблиці *Замовлення* створюємо за допомогою *Майстра* (рис. 5.83) форму *Замовлення* для вводу даних, яка буде містити наступні поля:

- Номер з/п;
- Тип устаткування;
- Назва устаткування;
- Ціна устаткування;
- Фірма виробник устаткування;
- Рік виробництва;
- Країна виробник;

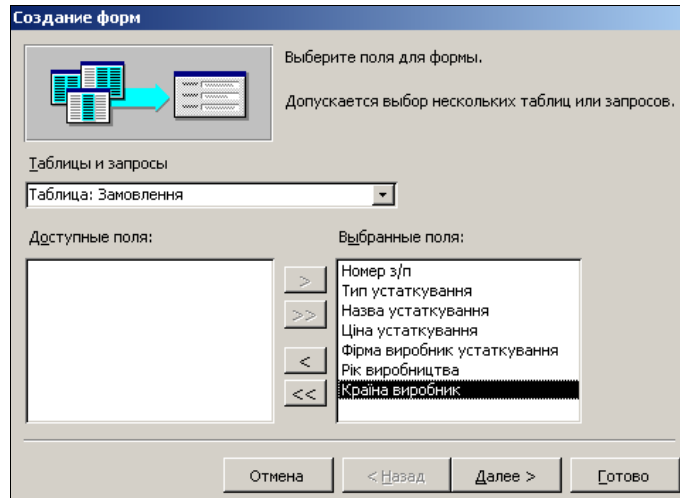


Рис. 5.83. Вікно Конструктора форм під час створення форми *Замовлення*.

У результаті, після виходу з режиму Конструктора, отримуємо наступну форму з записами (рис. 5.84).

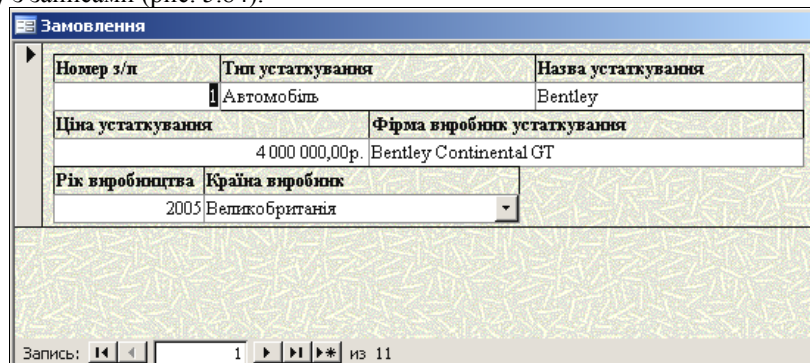


Рис. 5.84. Фрагмент вікна форми *Замовлення*.

3. Створюємо елементи керування формою. Для цього переходимо в режим Конструктора форми, за допомогою мишки змінюємо розміри *Області даних*, на панелі елементів вибираємо функційний елемент *Кнопка* та вставляємо його у область даних форми. Відкриється вікно створення кнопок (рис. 5.85).

У стовпці *Категорії* вибираємо пункт *Різні*, у стовпці *Дії* – *Друк таблиці* – *Готово*.

Програмісти, системні адміністратори та інженери спілкуються між собою "своєю", добре їм зрозумілою мовою. Такий стан речей здається влаштує усі зацікавлені сторони, тому, видається доцільним уточнення деяких термінів, тобто необхідно конкретизувати зміст того або іншого терміну з переліку основних.

- Елемент (element) конструкція мови HTML. Це контейнер, який містить дані і дозволяє формувати їх певним чином. Довільна Web-сторінка є набором елементів. Одна з головних ідей гіпертексту – можливість вкладення елементів.

- Дескриптор – початковий або кінцевий маркери елементів. У англійській літературі вживається, також, термін тег (tag). Це одне і те ж саме. Дескриптор – це основний елемент кодування прийнятий у стандарті HTML. Дескриптори визначають границі дії елементів і відділяють елементи один від одного. Дескриптори записуються у кутових дужках (< >). Наприклад, дескриптором формування абзацу є <P>, а дескриптором формування горизонтальної лінії – <HR>.

У HTML існує два типи дескрипторів: контейнери і одинарні дескриптори. Контейнер – це дескрипторна пара, яка складається з початкового (або відкриваючого) і кінцевого (або закриваючого) дескрипторів. Початковий дескриптор має вигляд <TAG>, де TAG – це ім'я реального дескриптора HTML, а кінцевий дескриптор має вигляд </tag>. Кінцевий дескриптор відрізняється від початкового тільки тим, що перед іменем дескриптора записується символ "/" (похила риска). Контейнери призначені для зберігання певної інформації, наприклад, тексту або інших HTML-дескрипторів. Отже, між початковим і кінцевим дескрипторами знаходиться вміст контейнера.

Зауважимо, що для зручності редагування HTML-коду Web-документу ім'я початкового дескриптора записується великими буквами, а кінцевого – малими. Згадане зауваження носить рекомендаційний характер.

- Гіперпосилання – фрагмент тексту, який є вказівником на інший файл або об'єкт. Гіперпосилання потрібні для того, щоб забезпечити можливість переходу від одного Web-документу до іншого.

- Фрейм (frame) – цей термін має два значення. Перше – область документу зі своїми смугами прокручування. Друге значення – одинарне зображення у складному (анімаційному) об'єкті (за аналогією з кадром кінофільму).

7.2. Структура Web-сторінки

У подальшому для створення Web-документів будемо використовувати засоби HTML. Загальну структуру Web-сторінки продемонструємо таким типовим прикладом:

```
<HTML>
<HEAD>
<TITLE> назва вікна Web-сторінки </title>
</head>
```

РОЗДІЛ 7. HTML

HTML – Hyper Text Markup Language, мова розмічування гіпертексту. Це скорочена назва мови, за допомогою засобів якої створюються Web-документи.

Термін "програмування мовою HTML", який часто вживається у спеціалізованих літературних джерелах є не зовсім вдалим. HTML – це не мова програмування.

Розмічування гіпертексту дає можливість користувачеві визначити зовнішній вигляд документу. Для цього у текст документу вносять спеціальні коди або дескриптори (tags). До речі, здійснити це можна використовуючи програму Блокнот або довільний текстовий процесор. Створюючи Web-сторінку користувачеві доводиться виконувати багато операцій, які він звик виконувати працюючи з текстовим процесором. Наприклад, відзначити фрагмент тексту курсивом, перетворювати частину тексту у заголовки і багато іншого.

Отже, HTML використовується для подання Web-документу у певному вигляді, а мови програмування – для написання кодів програм, призначених розв'язувати певний клас задач.

Файл, який створений засобами текстового процесора (наприклад, Microsoft Word), можна прочитати далеко не на всіх комп'ютерних платформах. У той же час, створення HTML-файлів та їх публікування у WWW власне і означає, що інформація стає доступною довольному користувачеві довольного комп'ютера, який під'єднаний до Internet.

Варто звернути увагу на ряд характеристик, завдяки яким HTML стала такою популярною:

- HTML розроблена спеціально для WWW;
- HTML – це відкритий стандарт, який не є Власністю якоїсь фірми. HTML може використовувати довольний користувач і до того ж безкоштовно;
- HTML є відносно простою, тому що початково розроблялась орієнтуючись на пересічного користувача. В HTML-кодї можна зорієнтуватися настільки просто, як і у текстовому файлі;
- до HTML включено гіпертекст. Імовірно, це найбільш важлива можливість HTML. Для навігації Web-сторінками використовуються гіперпосилання, які прив'язані до тексту або графічного зображення. HTML розроблена таким чином, щоб розробники Web-документів могли простими і доступними засобами реалізовувати гіперпосилання у своїх документах;
- HTML підтримує мультимедіа. WWW відрізняється від інших засобів телекомунікацій тим, що підтримує різні форми передавання інформації, включаючи звук, текст, графічні зображення, відео тощо.

7.1. Основні терміни

Англомовне програмне забезпечення потрапляло до України різними шляхами, як офіційними, так і неофіційними. Як наслідок, у галузі інформаційних технологій сама по собі склалась унікальна лінгвістична ситуація.

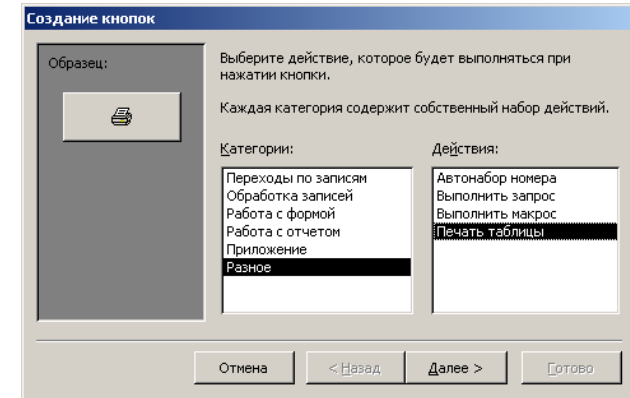


Рис. 5.85. Видяк вікна створення кнопки *Друк таблиці*.

Повторюємо операцію для встановлення функційних кнопок *Знайти запис* та *Додати запис*. Результат виконання подано на рис. 5.86.

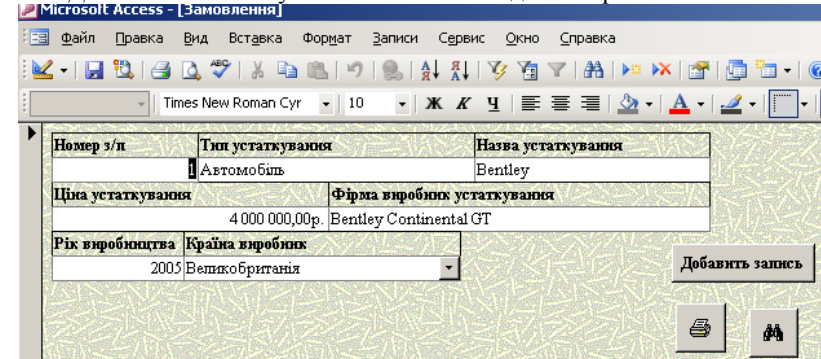


Рис. 5.86. Заповнена форма *Замовлення* зі створеними кнопками керування записами.

4. Налаштуємо параметри запуску бази даних, *Сервіс – Параметри запуску* – у полі *Вивід форми* вибираємо за основу форму *Замовлення* – *ОК* (рис. 5.87).

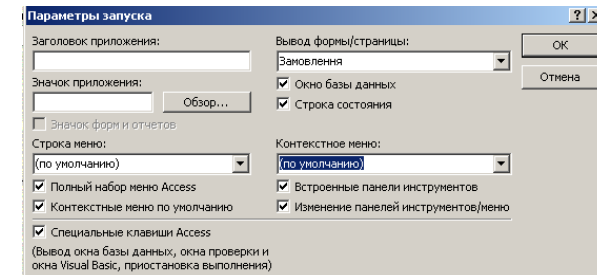


Рис. 5.87. Вікно налагодження параметрів запуску бази даних.

Закриваємо вікно програми та перезавантажуємо базу даних **Замовлення**. В результаті під час завантаження бази відкривається форма **Замовлення** для роботи з даною базою.

5. Створюємо запити для вибірки інформації упорядкованої за Назвою устаткування. Для цього переходимо у поле *Запити, Створення запиту за допомогою Майстра*. В якості вихідного елемента вибираємо таблицю **Замовлення**, додаємо всі доступні поля, *Далі – Далі*, вибираємо *Змінити макет запиту – Готово* (рис. 5.88).

Поле:	Номер з/п	Тип устаткування	Назва устаткування	Ціна устаткування
Імя таблиці:	Замовлення	Замовлення	Замовлення	Замовлення
Сортировка:			по возрастанию	
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Условие отбора:			Like "Ferrari"	
или:				

Рис. 5.88. Фрагмент заповненої таблиці Конструктора запиту для вибору за назвою устаткування.

Для стовпця Назва устаткування у полі *Сортування* вибираємо умову *За зростанням*. Закриваємо запит із збереженням змін. У результаті перегляду отримуємо (рис. 5.89).

Замовлення Запрос : запрос на выборку				
Номер з/п	Тип устаткування	Назва устаткування	Ціна устаткування	Фірма виробник устатку
8	Автомобіль	Ferrari 360	2 000 000,00р.	Ferrari
9	Автомобіль	Ferrari Dino 206 GT	6 100 000,00р.	Ferrari
7	Автомобіль	Ferrari F430	10 400 000,00р.	Ferrari
(Счетчик)			0,00р.	

Рис. 5.89. Фрагмент результату виконання запиту поданого на рис. 5.88.

6. Створюємо запит для вибірки даних за країною виробником. Для цього переходимо у поле *Запити, Створення запиту за допомогою Майстра*. В якості вихідного елемента вибираємо таблицю **Замовлення**, додаємо всі доступні поля, *Далі – Далі*, вибираємо *Змінити макет запиту – Готово*. Для стовпця *Країна виробник*, як приклад, у полі *Умови вибору* вводимо умову – *Італія* (рис. 5.90).

Назва устаткувані	Ціна устаткування	Фірма виробник ус	Рік виробництва	Країна виробник
Замовлення	Замовлення	Замовлення	Замовлення	Замовлення
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
				"Італія"

Рис. 5.90. Фрагмент заповненої таблиці запиту для вибірки за країною виробником – Італія.

Закриваємо запит із збереженням змін. У режимі таблиці отримуємо (рис. 5.91).

Назва устаткування	Ціна устаткування	Фірма виробник устатку	Рік виробництв	Країна виробник
Lamborghini Murcielago	13 234 655,00р.	Lamborghini	2009	Італія
Lamborghini Diablo	56 719 952,00р.	Lamborghini	2009	Італія
Ferrari F430	10 400 000,00р.	Ferrari	2009	Італія
Ferrari 360	2 000 000,00р.	Ferrari	2000	Італія
Ferrari Dino 206 GT	6 100 000,00р.	Ferrari	1968	Італія
	0,00р.		0	

Рис. 5.91. Фрагмент результатів виконання запиту поданого на рис. 5.80.

про те, як пов'язані між собою позиції на екрані і елементи HTML-документа, які відносяться до кожного анкера. При виборі користувачем довільного елемента з використанням "мишки" браузер порівнює інформацію про поточну позицію курсора і інформацію, яка зберігається, про положення анкерів, визначаючи, чи був користувачем вибраний анкер. У даному випадку механізм HTML-позначення гіпертекстового посилання називається **анкером**.

нем локалізування зв'язків. Користувач може звернутися до Web-документу на одному комп'ютері, а потім негайно перейти за посиланням до Web-документу на іншому комп'ютері. Однак, надто швидке припинення з'єднання може призвести до непотрібних втрат у тих випадках, якщо браузер повинен неодноразово звертатися до одного і того ж сервера для отримання великої кількості документів. Припустимо, що документ містить посилання на декілька зображень, які розміщені на тому ж комп'ютері, що і документ. Після вибору користувачем документу браузер відкриває з'єднання, отримує документ і закриває з'єднання. Віднайшовши необхідність вивести на екран зображення, браузер повинен відкрити нове з'єднання з тим же Web-сервером для отримання копії зображення.

6.7.5 Архітектура програмного забезпечення браузера

Програми Web-браузерів мають більш складну структуру у відношенні до Web-серверів. Сервер постійно виконує одну і ту ж нескладну функцію: очікує, поки довільний браузер не відкриє з'єднання і не здійснить запит документу. Потім Web-сервер передає копію вказаного у запиті елементу даних, закриває з'єднання і очікує наступного запиту на встановлення з'єднання. Основну частину операції з забезпечення доступу до Web-документу і його подальшого відтворення виконує браузер. З огляду на це браузер містить декілька великих програмних компонентів, які тісно взаємодіють один з одним, створюючи враження безперерійно функціонуючої служби. Зокрема, браузер включає цілий ряд клієнтів, широкий перелік інтерпретаторів, а також контролер, який керує цими програмними компонентами. Контролер утворює центральну частину браузера. Він інтерпретує і клацання "мишкою", і введення з клавіатури, а також викликає інші компоненти для виконання операцій, вказаних користувачем. Наприклад, при введенні користувачем URL або клацанні "мишкою" на гіпертекстовому посиланні контролер викликає клієнтську програму для відбору вказаного документу з віддаленого сервера, на якому він поміщений, а потім викликає інтерпретатор для відтворення документу на екрані користувача.

Браузер повинен бути оснащений інтерпретатором HTML для відтворення документів. Інші інтерпретатори є необов'язковими. На вхід інтерпретатора HTML надходить документ, який відповідає синтаксичним правилам HTML; на вхід інтерпретатора надходить відформатована версія документу, призначена для відтворення на екрані користувача. Інтерпретатор виконує компонування, перетворюючи специфікацію HTML у команди, які підходять для апаратного забезпечення дисплея, за котрим працює користувач. Наприклад, зустрівши у документі елемент заголовку розділу, інтерпретатор змінює розмір тексту, який служить для відтворення заголовку розділу. Зустрівши елемент розриву стрічки, інтерпретатор відкриває нову сторінку виводу.

Одна з найбільш важливих функцій інтерпретатора HTML відноситься до елементів, які вибираються. Інтерпретатор повинен зберігати інформацію

7. Створюємо запит для вибірки даних за найнижчою вартістю серед автомобілів фірми Bentley. Для цього переходимо у поле *Запити, Створення запиту за допомогою Майстра*. В якості вихідного елементу вибираємо таблицю *Замовлення*, додаємо всі доступні поля, *Далі – Далі*, вибираємо *Змінити макет запиту – Готово*. Заповними таблицю запиту так як подано на рис. 5.92.

Поле:	Фірма виробник устаткування	Ціна устаткування
Имя таблицы:	Замовлення	Замовлення
Сортировка:		по возрастанию
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Условие отбора: или:	"Bentley"	

Рис. 5.92. Фрагмент заповненої таблиці запиту для знаходження мінімальної ціни виробника.

На панелі інструментів виберемо піктограму *Набір значень* і надамо їй значення 1. Закриваємо запит із збереженням змін і виконаємо запит. У результаті отримуємо (рис. 5.93).

Замовлення мінімум : запит на вибірку	
Фірма виробник устаткування	Ціна устаткування
Bentley	4 000 000,00 грн.

Рис. 5.93. Фрагмент результатів виконання запиту поданого на рис. 5.92.

7. Створюємо звіт на основі побудованої бази. Для цього вибираємо об'єкт *Звіти – Створення звіту за допомогою Майстра*, додаємо всі доступні поля, *Далі – Готово*. В результаті отримуємо (рис. 5.94).

Замовлення						
мер з/п	Тип устаткув	Назва устатку	Ціна уст.	Фірма вироб	Рік вироб.	Країна
1	Автомобіль	Bentley	4 000 000,00р.	Bentley Continental G	2005	Велика
2	Автомобіль	Jaguar SS	225 000,00р.	Jaguar	1992	Велика
3	Автомобіль	Land Rover Freela	320 000,00р.	Land Rover	2009	Велика
4	Автомобіль	Rolls-Royce Corni	6 217 905,00р.	Rolls-Royce	2003	Велика
5	Автомобіль	Lamborghini Muro	13 234 655,00р.	Lamborghini	2009	Італія
6	Автомобіль	Lamborghini Diabl	56 719 952,00р.	Lamborghini	2009	Італія
7	Автомобіль	Ferrari F430	10 400 000,00р.	Ferrari	2009	Італія
8	Автомобіль	Ferrari 360	2 000 000,00р.	Ferrari	2000	Італія
9	Автомобіль	Ferrari Dino 206	6 100 000,00р.	Ferrari	1968	Італія
10	Автомобіль	Bentley Continent	5 900 000,00р.	Bentley	2006	Велика
11	Автомобіль	Land Rover Defen	1 450 000,00р.		0	Велика

Рис. 5.94. Фрагмент звіту на основі створеної бази даних **Замовлення**.

Завдання 5.

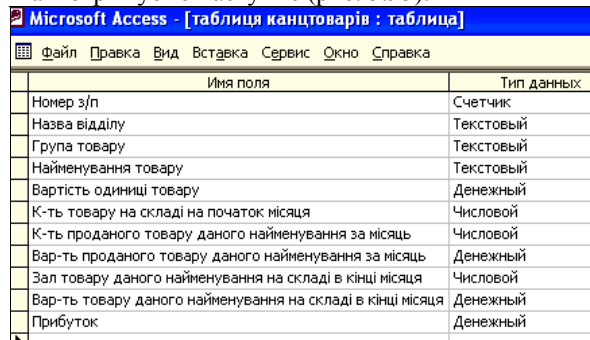
Створити базу даних **Канцтовари**, яка буде містити таблицю *Канцтовари* з такими полями: Номер з/п, Група товару, Найменування товару, Вартість одиниці товару, Кількість товару на складі на початок місяця, Кількість проданого товару даного найменування за місяць, Вартість проданого товару даного найменування за місяць, Залишок товару даного найменування на складі в кінці місяця. На основі таблиці *Канцтовари* створити форму *Канцтовари* для вводу даних. На формі створити елемент керування для додавання, друку та збереження записів. Створити запит на вибірку за назвою товару та відсортувати записи за зростанням. Створити у формі обчислювальне поле *Прибуток*. Створити звіт на основі побудованої бази.

Хід роботи:

1. Створюємо базу даних **Канцтовари**, яка буде містити таблицю *Канцтовари*. Для цього використовуємо пункт *Створення таблиці в режимі Конструктора* на закладці *Таблиця*. Обов'язковими полями є:

- Номер з/п – тип даних Лічильник;
- Група товару – тип даних Текстовий;
- Найменування товару – тип даних Текстовий;
- Вартість одиниці товару – тип даних Грошовий;
- Кількість товару на складі на початок місяця – тип даних Числовий;
- Кількість проданого товару даного найменування за місяць – тип даних Числовий;
- Вартість проданого товару даного найменування за місяць – тип даних Грошовий;
- Залишок товару даного найменування на складі в кінці місяця – тип даних Числовий;
- Вартість товару даного найменування на складі в кінці місяця – тип даних Грошовий;
- Прибуток – тип даних Грошовий.

У результаті отримуємо наступне (рис. 5.95).



Имя поля	Тип данных
Номер з/п	Счетчик
Назва відділу	Текстовый
Група товару	Текстовый
Найменування товару	Текстовый
Вартість одиниці товару	Денежный
К-ть товару на складі на початок місяця	Числовой
К-ть проданого товару даного найменування за місяць	Числовой
Вар-ть проданого товару даного найменування за місяць	Денежный
Зал товару даного найменування на складі в кінці місяця	Числовой
Вар-ть товару даного найменування на складі в кінці місяця	Денежный
Прибуток	Денежный

Рис. 5.95. Таблиця *Канцтовари* в режимі *Конструктора*

дати, видалити, змінити або переіменувати документ на своєму вузлі не повідомляючи про це адміністраторів інших вузлів. Як наслідок, посилання з одного Web-документу на інший не завжди є дійсним. Допустимо, що документ D_1 на комп'ютері K_1 містить посилання на документ D_2 на комп'ютері K_2 . Якщо адміністратор, який відповідає за роботу комп'ютера K_2 вирішить видалити документ D_2 , тоді посилання на нього у документі D_1 стає недійсним. Оскільки адміністратори, які забезпечують роботу комп'ютерів, на яких зберігаються Web-документи, не завжди координують свої дії, посилання з одного документу на інший бувають недійсними.

6.7.3. Подання документів

Оскільки Web-документ може містити багато інформаційних елементів його формат повинен добиратися дуже ретельно, щоб вміст Web-документу міг інтерпретуватися довільним браузером. Зокрема, браузер повинен розрізняти довільний текст, графіку, посилання на інші документи. До того у розпорядженні автора повинні бути засоби опису загального компонування документу (наприклад, порядку, у якому повинні бути подані елементи).

Для кожного Web-документу, який містить гіпермедійний документ, використовується стандартне подання. Цей стандарт, відомий під назвою мови розмічування гіпертексту (Hyper Text Markup Language – HTML), дозволяє автору документу реалізувати свої задуми з відтворення інформації на Web-сторінці і вказати, що на ній знаходиться.

HTML називають мовою розмічування, оскільки вона не передбачає детального опису формату сторінки. HTML дозволяє включати у документ загальні вказівки з його відтворення і вибирати конкретні значення параметрів у браузері. Звідси, двома різними браузерами один і той же документ може бути відтворений у різному вигляді.

6.7.4. Взаємодія типу "клієнт-сервер"

Як і у інших мережевих додатках, у програмах перегляду Web-документів застосовується модель взаємодії типу "клієнт-сервер". Після отримання URL документу браузер стає клієнтом, який звертається з запитом на отримання документу до сервера, який працює на комп'ютері, вказаному в URL. Пізніше браузер відтворює документ для користувача. На відміну від мережевих додатків, у яких передбачене постійне з'єднання між клієнтом і сервером, для взаємодії між Web-браузером і сервером з'єднання створюється тільки протягом короткого часового відтинку. Браузер встановлює з'єднання, відсилає запит і отримує або необхідний елемент даних, або повідомлення, що такого елемента не існує. Одразу ж після пересилання документу або зображення з'єднання закривається; клієнт від'єднується від Web-сервера.

Швидке припинення з'єднань цілком себе виправдовує у більшості випадків, оскільки перегляд Web-документу не характеризується високим ступе-

Transfer Protocol – протокол пересилання гіпертексту). Після імені протоколу ставлять двокрапку (:) та два знаки (/) (похила риска): http://...

2. Указання **доменного імені** комп'ютера (серверу), на якому зберігається цей ресурс: http://www.abcede.com...

3. Указання повного шляху доступу до файлу:

http://www.abcede.com/File/New\abcdefq.zip...

Записуючи URL-адресу, важливо точно дотримуватися реєстру символів. На відміну від правил роботи у Windows, в Internet великі та малі літери у іменах каталогів і файлів вважаються різними.

Саме у форматі URL і зв'язують адресу ресурсу з гіпертекстовими посиланнями у Web-документах. Після активування гіперпосилання оглядач надсилає запит в Internet для пошуку та досилання ресурсу, відзначеного у посиланні. У разі, коли з якихось причин його не знайдено, подається повідомлення, що ресурс є недоступним (можливо, що сервер тимчасово недоступний або змінилася адреса ресурсу).

6.7.2. Інфраструктура World Wide Web та інтерактивний перегляд

World Wide Web – це великомасштабний, оперативний репозитарій інформації, у якому користувачі можуть виконувати цілеспрямований пошук з використанням інтерактивної прикладної програми, яка називається **браузером** (оглядачем). Основне призначення браузера – перегляд інформації у World Wide Web.

За своєю суттю World Wide Web є розподіленою системою, яка забезпечує інтерактивний доступ. Гіпермедійна система виникла на основі гіпертекстової системи. В обох системах інформація зберігається у вигляді наборів документів. Крім самої інформації, документ може містити вказівники на інші документи у наборі. Кожен вказівник пов'язаний з елементом, доступним для вибору, що дозволяє вибрати елемент і переміститися за вказівником до документу, який пов'язаний з поточним документом. Різниця між гіпертекстовим і гіпермедійним середовищем визначається вмістом документів: гіпертекстові документи містять тільки текстову інформацію, а гіпермедійні документи можуть містити інформацію, подану в інших форматах, включаючи оцифровані фотографії та графічні об'єкти, супроводжувані звуком, відеозображеннями тощо.

Різниця між розподіленою і нерозподіленою гіпермедійними системами є значною. У нерозподіленій системі інформація розміщена на одному комп'ютері, часто навіть на одному диску. Оскільки доступ до всього набору документів може бути здійсненим на місці, вказівники, які проводять від одного документу до іншого (посилання), можна легко перевірити, чи є вони реальними (дійсними). Це означає, що у нерозподіленій гіпермедійній системі можна добитися того, що усі посилання будуть дійсними і вказуватимуть на існуючі ресурси.

У World Wide Web документи розподілені на великій кількості комп'ютерів. Окрім того, системний адміністратор довільного комп'ютера може до-

Закриваємо режим *Конструктора* та відкриваємо таблицю в режимі вводу даних. Вводимо 10 довільних записів (рис. 5.96).

Номер з/п	Назва відділу	Група товару	Найменування товару	Вартість одиниці товару	Кількість то
1	Роздрібно́ї торгівлі	Ручки	Ручка чорнильна	30,00р.	
2	Роздрібно́ї торгівлі	Бланки	Будинкова книга формату А4 на 16	2,79р.	
3	Оптово́ї торгівлі	Бланки	Довіреність (офсетна) А5, 70г/м	5,69р.	
4	Оптово́ї торгівлі	Бланки	Дорожній лист вантажного автомобіля	5,38р.	
5	Роздрібно́ї торгівлі	Бланки	Журнал обліку відомостей по ТМЦ	5,35р.	
6	Роздрібно́ї торгівлі	Бланки	Журнал реєстрації довіреностей	6,31р.	
7	Оптово́ї торгівлі	Зошити	Альбом для малювання 30 аркушів	8,76р.	
8	Оптово́ї торгівлі	Зошити	Зошит загальний формату А4	4,00р.	
9	Роздрібно́ї торгівлі	Зошити	Зошити 12літ. ,	0,85р.	
10	Роздрібно́ї торгівлі	Зошити	Книга канцелярська	21,20р.	
11	Оптово́ї торгівлі	Папір	Блок Мікс 85х85х400л кл 1361	4,95р.	
12	Оптово́ї торгівлі	Папір	Ватман 170г/м 62*86, А1	2,29р.	
13	Роздрібно́ї торгівлі	Папір	Папір в рулоні ЛУ45-210 (газетний)	7,63р.	
14	Роздрібно́ї торгівлі	Ручки	Лайнер uni Pin 0.05мм fine line чорний	15,21р.	

Рис. 5.96. Фрагмент вікна таблиці *Канцтовари*.

2. На основі таблиці *Канцтовари* створюємо за допомогою *Майстра* форму *Канцтовари* для вводу даних, яка буде містити наступні поля:

- Номер з/п;
- Група товару;
- Найменування товару;
- Вартість одиниці товару;
- Кількість товару на складі на початок місяця;
- Кількість проданого товару даного найменування за місяць;
- Вартість проданого товару даного найменування за місяць;
- Залишок товару даного найменування на складі в кінці місяця;
- Вартість товару даного найменування на складі в кінці місяця;
- Прибуток.

У результаті, після виходу з режиму *Конструктора*, отримуємо наступну форму з записами (рис. 5.97).

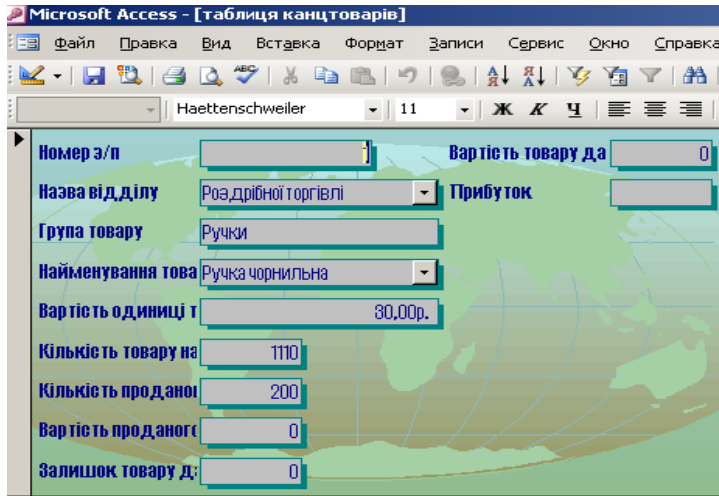


Рис. 5.97. Фрагмент вікна форми *Канцтовари* з першим записом.

3. Створюємо елементи керування формою. Для цього переходимо в режим *Конструктора* форми, за допомогою мишки змінюємо розміри *Області даних*, на панелі елементів вибираємо функційний елемент *Кнопка* та вставляємо його у область даних форми. Відкриється вікно створення кнопок.

У стовпці *Категорії* вибираємо пункт *Обробка записів*, у стовпці *Дії* – *Додати запис* – *Готово* (рис. 5.98).

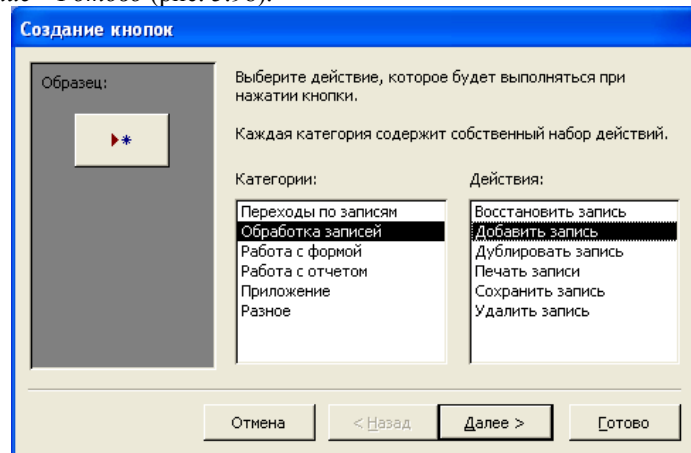


Рис. 5.98. Вигляд вікна створення кнопки *Додати запис*.

Повторюємо операцію для встановлення функційних кнопок *Друк запису* та *Зберегти запис*. Результат виконання подано на рис. 5.99.

вікні броузера гіпертекст (або гіперпосилання) звичайно підкреслюється і відзначається іншим кольором на відміну від решти тексту. Якщо клацнути "мишкою" на гіпертексті, тоді його колір змінюється. Клацання "мишкою" на гіпертексті призводить до того, що Web-броузер робить запит документу, на який вказує посилання, а потім завантажує його у вікно броузера.

Таким чином, завдяки гіпертексту Web-сторінка набуває властивість інтерактивності. Власне гіпертекст закладено в основу структури WWW. Дуже часто гіпертекстове посилання називають "гарячим" або "активним" текстом.

У гіпертексту немає ні початку, ні кінця. Наприклад, більшість друкованих видань призначені для прочитування від першої сторінки до останньої. У цьому сенсі їм притаманна лінійна структура, а гіпертекст більше нагадує павутину з невизначеним центром.

В основі гіпермедіа лежить така ж ідея, але після клацання "мишкою" на посиланні завантажуються не нова Web-сторінка, а мультимедійний документ, наприклад, звуковий файл або оцифрований відеофільм. Якщо клацнути "мишкою" на такому посиланні, тоді Web-броузер завантажить відповідний мультимедійний файл і запустить на комп'ютері користувача допоміжний програмний додаток.

Більшість сучасних Web-броузерів підтримує модульну технологію (plug-ins), котра дозволяє відтворювати відео- і аудіозаписи безпосередньо при відтворенні Web-сторінки, тобто створюється враження, що мультимедійний файл є частиною Web-сторінки.

Таким чином, сукупність величезної кількості гіпертекстових електронних документів, які зберігаються на серверах WWW, утворює своєрідний **гіперпростір документів**, між якими можливе переміщення. Довільне переміщення між документами у Web-просторі називають **Web-серфінгом** (виконують для перегляду з метою ознайомлення). Цілеспрямоване переміщення між Web-документами називають **Web-навігацією** (виконують з метою пошуку потрібної інформації).

Гіпертекстовий зв'язок між сотнями мільйонів документів, які зберігаються на фізичних серверах Internet, є основою існування **логічного простору World Wide Web**. Однак такий зв'язок не був би можливим, якби кожен документ у цьому просторі не мав своєї унікальної адреси. Кожен файл одного локального комп'ютера має **унікальне повне ім'я**, до якого входить ім'я файлу (включаючи розширення імені) та шлях доступу до файлу, починаючи з імені пристрою, на якому він зберігається. Тепер ми можемо розширити уявлення про унікальне ім'я файлу та розвинути його до WWW. Адреса довільного файлу у WWW визначається **уніфікованим покажчиком ресурсу** – URL.

Адреса URL складається з трьох частин:

1. Указання служби, яка здійснює доступ до цього ресурсу (звичайно вказують ім'я прикладного протоколу, який відповідає необхідній службі). Так, наприклад, для служби WWW прикладним є протокол HTTP (Hyper Text

World Wide Web – це єдиний інформаційний простір, який складається з сотень мільйонів взаємопов'язаних електронних документів, які зберігаються на Web-серверах. Окремі документи, які становлять простір Web, називають Web-документами. Групи тематичних об'єднань Web-документів називають Web-вузлами (неусталений термін Web-сайт або просто сайт). Один фізичний Web-сервер може містити досить багато Web-вузлів, для кожного з яких відводиться окрема тека на жорсткому диску сервера.

Від звичайних текстових документів Web-документи відрізняються тим, що вони оформлені безвідносно до конкретного носія. Наприклад, оформлення документа, надрукованого на папері, прив'язане до параметрів друкованого аркуша, який має певну висоту, ширину та розмір полів. Електронні Web-документи призначені для перегляду на екрані комп'ютера, до того ж заздалегідь не відомо, на якому саме. Невідомі ні розміри екрану, ні параметри кольорового та графічного розширення, невідома навіть операційна система, з якою працює комп'ютер клієнта. Тому Web-документи не можуть мати жорсткого форматування. Оформлення виконується безпосередньо під час їхнього відтворення на комп'ютері клієнта й провадиться відповідно до налаштування програми, яка здійснює перегляд.

Програми для перегляду Web-документів називають **оглядачами**. У літературі також можна зустріти неусталений термін **броузер**. Оглядач здійснює відтворення документа на екрані, керуючись командами, які автор документу впровадив у його текст. Такі команди називають **тегами** або **дескрипторами**. Існують спеціальні дескриптори для впровадження графічних і мультимедійних об'єктів (звук, музика, відеокліпи). Натрапивши на такий дескриптор, оглядач посилає запит до сервера стосовно досилання файлу, пов'язаного з дескриптором, та відтворює його відповідно до заданих атрибутів і параметрів дескриптора – ми бачимо ілюстрацію й чуємо звук.

Останнім часом у Web-документах широко застосовують **активні компоненти**. Це також об'єкти, але вони містять не лише текстові, графічні та мультимедійні дані, але й програмний код, тобто можуть не просто відтворюватися на комп'ютері клієнта, але й виконувати роботу за вкладеною до них програмою.

Можливість впровадження у текст графічних об'єктів є однією з найефективніших з огляду на оформлення Web-документів, але не найголовнішою з огляду самої ідеї WWW. Найважливішою особливістю Web-документів є можливість реалізування **гіпертекстових посилань**. З довільним фрагментом тексту або рисунком можна пов'язати інший Web-документ, тобто встановити **гіперпосилання**. У цьому випадку після клацання лівою клавішею "мишки" на тексті або рисунку, які є гіперпосиланням, у мережу надходить запит стосовно досилання нового документу. Цей документ, у свою чергу, також може містити гіперпосилання на інші Web-документи.

Гіпертекст (hypertext) – це особливий текст, який містить посилання на іншу Web-сторінку або документ. У даному випадку **гіпер** (hyper) означає більш як або понад, так як Ви бачите перед собою більше, ніж просто текст. У

Рис. 5.99. Форма *Канцтовари* з створеними функційними кнопками.

5. Створюємо запити для вибірки інформації, упорядкованої за Назвою устаткування. Для цього переходимо у поле *Запити, Створення запити за допомогою Майстра*. В якості вихідного елемента вибираємо таблицю *Канцтовари*, додаємо всі доступні поля, *Далі – Далі*, вибираємо *Змінити макет запити – Готово* (рис. 5.100).

Рис. 5.100. Створення запити за допомогою Майстра.

Для стовпця Найменування товару – у полі *Сортування* вибираємо – *За зростанням*. Закриваємо запит із збереженням змін. У результаті отримуємо (рис. 5.101).

Поле: Имя таблицы: Сортировка: Вывод на экран: Условие отбора: или:	Номер з/п	Назва відділу	Група товару	Найменування тов	Вартість одиниці т
	таблица канцтова	таблица канцтова	таблица канцтова	таблица канцтова	таблица канцтова
	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Рис. 5.101. Фрагмент заповненої таблиці *Конструктора* запиту для вибору та сортуванню за полем Найменування товару.

Закриваємо запит із збереженням змін. У результаті перегляду отримуємо (рис. 5.102).

Номер з.	Назва відділу	Група товару	Найменування товару
16			
7	Оптові торгівлі	Зошити	Альбом для малювання 30 аркушів
11	Оптові торгівлі	Папір	Блок Мікс 85x85x400л кл 1361
2	Роздрібної торгівлі	Бланки	Будинкова книга формату А4 на 16
12	Оптові торгівлі	Папір	Ватман 170г/м 62*86, А1
3	Оптові торгівлі	Бланки	Довіреність (офсетна) А5, 70г/м
4	Оптові торгівлі	Бланки	Дорожній лист вантажного автомобіля
5	Роздрібної торгівлі	Бланки	Журнал обліку відомостей по ТМЦ
6	Роздрібної торгівлі	Бланки	Журнал реєстрації довіреностей
8	Оптові торгівлі	Зошити	Зошит загальний формату А4
9	Роздрібної торгівлі	Зошити	Зошити 12лін.
10	Роздрібної торгівлі	Зошити	Книга канцелярська
14	Роздрібної торгівлі	Ручки	Лайнер uni Pin 0.05мм fine line, чорний
15	Оптові торгівлі	Ручки	Лайнер uni Pin 0.1мм fine line, чорний
13	Роздрібної торгівлі	Папір	Папір в рулоні ЛУ45-210 (газетний)
1	Роздрібної торгівлі	Ручки	Ручка чорнильна

Рис. 5.102. Фрагмент результату запиту, отриманого за допомогою QBE-таблиці, зображеної на рис. 5.101.

5. Відкриваємо форму для вводу обчислювального поля Прибуток. Додаємо поле і назвемо його Прибуток. Відкриваємо властивість даного поля та за допомогою побудовувача виразів вводимо умову =[Вартість одиниці товару]*[Кількість проданого товару за місяць].

Закриваємо форму із збереженням. У результаті отримуємо (рис. 5.103).

женого доступу, які надають інформацію лише зареєстрованим клієнтам – вони висилають запит на введення імені користувача та зв'язаного з ним пароля. Однак існують FTP-сервери з **анонімним доступом** для всіх бажаючих.

Служба IRC. Служба IRC (Internet Relay Chat) призначена для прямого спілкування кількох людей у режимі реального часу. Іноді службу IRC називають **чат-конференціями** або просто **чатом**. На відміну від системи телеконференцій, у якій спілкування між учасниками відкрите, у системі IRC спілкування відбувається у межах лише одного каналу, у роботі якого беруть участь кілька осіб. Кожен користувач може створити власний канал та запросити до нього учасників або ж приєднатися до одного з відкритих у цей момент каналів.

Служба ICQ. Ця служба призначена для пошуку мережевої IP-адреси людини, яка на той час під'єднана до Internet. Потреба в такій послугі пов'язана з тим, що більшість користувачів не мають постійної IP-адреси. Назва служби є акронімом виразу I seek you – я шукаю тебе. Щоб скористатися цією службою, слід зареєструватися на її центральному сервері (<http://www.icq.com>) та отримати персональний ідентифікаційний номер UIN (Universal Internet Number). Його можна розцінювати як аналог персонального "номера" та повідомити своїм партнерам.

Кожен комп'ютер, під'єднаний до Internet, повинен мати чотиризначну IP-адресу. IP-адреса може бути **постійною** й **тимчасовою (динамічною)**. Ті комп'ютери, які під'єднані до Internet на постійній основі, мають постійні IP-адреси. Більшість же користувачів під'єднуються до Internet лише на час сеансу. Їм надають динамічну IP-адресу, яка діє лише протягом одного сеансу. Цю адресу надає той сервер, через який відбулося з'єднання. У різних сеансах динамічна IP-адреса може бути різною, до того ж наперед не відомою. Під час кожного під'єднання до Internet програма ICQ, яка встановлена на Вашому комп'ютері, визначає діючу IP-адресу й повідомляє її центральної службі, яка у свою чергу сповіщає Ваших партнерів з контактів про те, що:

- Ваш комп'ютер у цей час під'єднаний до Internet;
- готовий до приймання виклику;
- має таку IP-адресу.

Далі Ваші партнери (якщо вони теж є клієнтами цієї служби) можуть встановити з Вами прямий зв'язок. Програма надає можливість вибору режиму зв'язку ("готовий до контакту"; "прошу не турбувати, але готовий отримати термінове повідомлення"; "закритий для контакту" тощо). Після встановлення контакту зв'язок проходить у режимі, аналогічному до сервісу IRC.

6.7. Основи функціонування WWW

6.7.1. Основні поняття

Безумовно, WWW – це найпопулярніша служба сучасного Internet. Багато хто навіть гадає, що WWW – це і є Internet, хоч, звичайно, це лише одна з його численних служб.

ключовими словами. В останньому випадку говорять про **автоматичне модерування**.

Для роботи зі службою телеконференцій існують спеціальні клієнтські програми. Так, наприклад, додаток Microsoft Outlook Express, згадуваний вище як поштовий клієнт, придатний і для роботи зі службою телеконференцій. Перед початком роботи треба налаштувати програму на роботу з сервером груп новин, оформити "передплату" на певні групи й періодично, як і електронну пошту, отримувати всі повідомлення, які надходять за темою цієї групи. У цьому випадку слово "передплата" не передбачає з боку клієнта жодних зобов'язань або платежів – це просто вказівка серверу про те, що повідомлення з певних тем треба висилати, а з інших – ні. Скасувати передплату або змінити її склад можна у довільний зручний момент.

Служба імен доменів (DNS). Коли мова йшла стосовно протоколів Internet, ми згадували, що адреса довільного комп'ютера або довільної локальної мережі в Internet може бути подана чотирма байтами, наприклад, так **195.24.132.97**.

А зараз ми заявили, що кожен комп'ютер має унікальне доменне ім'я, наприклад, таке:

<http://www.abcde.com>

Це просто дві різні форми запису одного й того ж мережевого комп'ютера. Людині незручно працювати з числовим поданням IP-адреси, зате доменне ім'я запам'ятати легко, особливо, якщо врахувати, що, як правило, таке ім'я має певний зміст. Наприклад, Web-сервер компанії Microsoft має ім'я www.microsoft.com. Неважко "реконструювати" імена й інших компаній.

З іншого боку, автоматична робота серверів мережі організована з використанням чотиризначної числової адреси. Завдяки цьому проміжні сервери можуть здійснювати пересилання запитів та відповідей у потрібному напрямку, не знаючи, де конкретно перебуває відправник та адресат. Саме тому необхідним є переклад доменних імен у пов'язані з ними IP-адреси. Цей переклад забезпечують сервери служби імен доменів DNS. Наш запит на досилання однієї зі сторінок серверу www.abcd.com спочатку опрацьовує DNS-сервер, далі його спрямовують уже не за доменним іменем, а за IP-адресою.

Служба пересилання файлів (FTP). Приймання та висилання файлів становлять значний відсоток від інших Internet-послуг. Необхідність у пересиланні файлів виникає, наприклад, під час приймання файлів програм або пересиланні великих документів (книг), а також під час пересилання архівних файлів, у яких "запаковані" великі обсяги інформації. Служба FTP має свої сервери у мережі, на яких зберігаються великі архіви даних. З боку клієнта для роботи з серверами FTP може бути встановлене спеціальне програмне забезпечення, хоч у більшості випадків програми-оглядачі WWW мають вмонтовані можливості для роботи і за протоколом FTP.

Протокол FTP надає серверу засоби для ідентифікування клієнта, який до нього звернувся. Цим часто користуються комерційні сервери та сервери обме-

Рис. 5.103. Форма *Канцтовари* з добавленим полем Прибуток.

6. Створюємо звіт на основі побудованої бази. Для цього вибираємо об'єкт *Звіти – Створення звіту за допомогою Майстра*, додаємо всі необхідні поля, *Далі – Готово*. У результаті отримуємо (рис. 5.104).

таблиця канцтоварів			
Номер з/п	Назва відділу	Група товару	Найменування товару
1	Роздрібноі торгівлі	Ручки	Ручка чорнильна
2	Роздрібноі торгівлі	Бланки	Будинков а книга формат
3	Оптові торгівлі	Бланки	Довіреність (офсетна) А5
4	Оптові торгівлі	Бланки	Дорожній лист вангажної
5	Роздрібноі торгівлі	Бланки	Журнал обліку відомосте
6	Роздрібноі торгівлі	Бланки	Журнал реєстрації довіре
7	Оптові торгівлі	Зошити	Альбом для малювання 3
8	Оптові торгівлі	Зошити	Зошит загальний формат
9	Роздрібноі торгівлі	Зошити	Зошити 12лн.,
10	Роздрібноі торгівлі	Зошити	Книга канцелярська
11	Оптові торгівлі	Папір	Блок Мікс 85x85x400л кл
12	Оптові торгівлі	Папір	Ватман 170г/м 62*86, А1
13	Роздрібноі торгівлі	Папір	Папір в рулоні ЛУ45-210 (

Рис. 5.104. Фрагмент звіту на основі створеної бази даних *Канцтовари*.

Завдання 6.

Створити базу даних **Музичні записи**, яка буде містити таблицю *Музичні записи* з такими полями: Номер з/п, Автор слів, Автор музики, Виконавець твору, Час звучання, Назва твору. На основі таблиці *Музичні записи* створити форму *Музичні записи* для вводу даних та встановити на формі елементи керування Знайти запис, та *Надрукувати запис*. Налаштувати параметри запуску бази даних таким чином, щоб під час відкритті бази **Музичні записи** зразу відкривалася форма *Музичні записи*. Створити запит на вибірку за автором твору. На основі бази даних **Музичні записи** створити звіт з групуванням за автором слів.

Хід роботи:

1. Створюємо базу даних **Музичні записи**, яка буде містити таблицю *Музичні записи*. Для цього використовуємо пункт *Створення таблиці в режимі Конструктора* на закладці *Таблиця*. Обов'язковими полями є:

- Номер з/п – тип даних Лічильник;
- Автор слів – тип даних Текстовий;
- Автор музики – тип даних Текстовий;
- Виконавець твору – тип даних Текстовий;
- Час звучання – тип даних Числовий;
- Назва твору – тип даних Текстовий.

У результаті отримуємо наступне (рис. 5.105).

Рис. 5.105. Таблиця *Музичні записи*, створена у режимі Конструктора.

Закриваємо режим *Конструктора* та відкриваємо таблицю в режимі вводу даних. Вводимо 10 довільних записів (рис. 5.106).

Номер з/п	Автор слів	Автор музики	Виконавець твору	Час звучання	Назва твору
1	Висоцький	Висоцький	Висоцький	5	Идет охота на волков
2	Івасюк	Івасюк	Івасюк	4	Червона рута
3	Івасюк	Івасюк	Івасюк	5	Водограй
4	Вакарчук	Вакарчук	Океан Ельзи	5	Лукава кішка
5	Вакарчук	Вакарчук	Океан Ельзи	6	Я так хочу до тебе
6	Чубинський	Вербицький	Пономарьов	4	Гімн України
7	Пономарьов	Пономарьов	Пономарьов	5	Зранку до ночі
8	Вакарчук	Вакарчук	Океан Ельзи	4	Автомобіль
9	Пономарьов	Пономарьов	Пономарьов	5	Молитва
10	Пономарьов	Пономарьов	Пономарьов	6	Любов-війна
11	Ані Лорак	Ані Лорак	Ані Лорак	5	Мій герой
12	Ані Лорак	Ані Лорак	Ані Лорак	0	Мій ангел

Рис. 5.106. Фрагмент вікна таблиці *Музичні записи*.

великий потік поштової інформації на свою адресу можна отримати, підписавшись на списки розсилок. Це спеціальні тематичні сервери, які збирають інформацію за певними темами та переспрямовують її дописувачам у вигляді повідомлень електронної пошти. Темою списків розсилок може бути довільна інформація, наприклад, питання, пов'язані з вивченням іноземних мов, огляди літературних новинок, презентування нових програмних та апаратних засобів. Більшість телекомпаній мають списки розсилок з оглядами телепрограм на найближчі дні. Коли глядач має доступ до багатьох телеканалів, виникає проблема вибору потрібного пересилання. Списки розсилок дозволяють її ефективно розв'язати.

Послуги більшості списків розсилок безоплатні. Опосередкованою платною за користування ними є певний відсоток реклами, яка надходить до електронної поштової скриньки. Оскільки служби списків розсилок засновані на електронній пошті, тому спеціальних програмних засобів, окрім звичайного поштового клієнта, для роботи з ними не потрібно.

Служба телеконференцій (Usenet). Служба телеконференцій схожа на електронну пошту, але не просту, а циркулярну, коли одне повідомлення надсилається не одному кореспондентові, а великій групі (такі групи називають **телеконференціями** або **групами новин**).

Звичайно повідомлення електронної пошти пересилають вузьким ланцюжком серверів від адресанта до адресата. При цьому не передбачається його зберігання на проміжних серверах. Повідомлення, спрямовані на сервер групи новин, висилають з нього на всі пов'язані з ним сервери, якщо на них цього повідомлення ще немає. Далі процес повторюється.

На кожному із серверів повідомлення, яке надійшло, зберігається досить тривалий час (звичайно тиждень), і всі бажаючі можуть протягом цього часу ознайомитися з ним. На сервери, де вже є це повідомлення, повторного пересилання не здійснюють. Уся система телеконференцій поділена на тематичні групи. Вони охоплюють більшість тем, які цікавлять користувачів. Особливу популярність мають групи, присвячені ІТ.

Основний зміст використання груп новин полягає у тому, щоб поставити запитання і отримати відповідь на нього від тих, хто його вже розв'язав. Однак, важливо стежити, щоб зміст запитання відповідав темі обраної телеконференції. Багато кваліфікованих фахівців регулярно переглядають повідомлення телеконференцій, які стосуються сфери їхньої діяльності. Такий перегляд називають **моніторингом інформації**. Регулярне монітування дозволяє фахівцям точно знати, що нового відбувається у світі з їхньої спеціальності, які питання турбують фахівців і на що варто звернути особливу увагу у своїй роботі.

Величезна чисельність повідомлень у групах новин значно ускладнюють монітування, тому в деяких групах здійснюють попереднє "відсіювання" непотрібної інформації (зокрема, рекламної, яка не стосується теми конференції). Такі конференції називають **модерованими**. **Модератором** може бути не лише людина, але й програма, яка фільтрує повідомлення за певними

- встановити зв'язок із сервером, який надає послуги FTP (FTP-сервером).

Для того, щоб скористатися послугами електронної пошти, ми повинні дотримуватися протоколу висилання та отримання повідомлень. Для цього потрібно мати програму (поштовий клієнт) та встановити зв'язок з поштовим сервером. Аналогічно треба діяти й під час роботи з іншими службами.

Термінальний режим (Telnet). В історичному плані однією з перших є служба віддаленого керування комп'ютером Telnet. Під'єднавшись до віддаленого комп'ютера за протоколом цієї служби, можна керувати його роботою. Таке керування ще називають **консольним** або **термінальним**. У минулому цю службу широко використовували для проведення складних математичних розрахунків у віддалених обчислювальних центрах.

У наш час, з огляду на швидке збільшення потужності персональних комп'ютерів, необхідність у такій послугі зменшилася, але разом з тим служби Telnet в Internet продовжують існувати. Часто протоколи Telnet використовують для дистанційного керування технічними об'єктами, наприклад, телескопами, які перебувають на інших континентах, або відеокамерою, яка встановлена у морі. Ми не вказуємо назв основних Telnet-клієнтів, оскільки кожен сервер, який надає Telnet-послуги, пропонує свій клієнтський додаток. Його треба отримати мережею (наприклад за протоколом FTP), встановити на своєму комп'ютері, під'єднатися до сервера й працювати.

Електронна пошта (E-Mail). Ця служба також є однією з найстаріших. Її обслуговують в Internet спеціальні поштові сервери. Звернемо увагу на те, що коли ми говоримо про довільний сервер, тоді не маємо на увазі, що це спеціально виділений комп'ютер. Тут, і далі, сервером можуть називатися програми. Тобто один вузловий комп'ютер Internet може виконувати функцію кількох серверів та забезпечувати роботу різних служб, залишаючись універсальним комп'ютером, на якому можна виконувати й інші завдання.

Поштові сервери отримують повідомлення від клієнтів і пересилають їх ланцюжком до поштових серверів адресатів, де ці повідомлення нагромаджуються. Під час встановлення з'єднання між адресатом і його поштовим сервером відбувається автоматичне висилання повідомлень, які надійшли на комп'ютер адресата.

Поштова служба базується на двох прикладних протоколах: SMTP та POP3. За першим відбувається висилання кореспонденції з комп'ютера на сервер, а за другим – приймання повідомлень, які надійшли. Існує багато різноманітних клієнтських програм. До них зараховують програму Microsoft Outlook Express, яка входить до складу операційної системи Windows як стандартна. Потужніша програма, яка інтегрує у собі, окрім підтримки електронної пошти, інші засоби діловодства – Microsoft Outlook (яка входить до складу пакету Microsoft Office). Зі спеціалізованих поштових програм добре зарекомендували себе програми Gmail, The Bat! та Eudora Pro.

Списки розсилання (Mail List). Звичайна електронна пошта передбачає наявність двох партнерів з листування. Якщо ж партнерів немає, тоді досить

2. На основі таблиці *Музичні записи* створюємо форму *Музичні записи* для вводу даних, яка буде містити наступні поля:

- Номер з/п ;
- Автор слів;
- Автор музики;
- Виконавець твору;
- Час звучання;
- Назва твору.

У результаті отримуємо (рис. 5.107).



Рис. 5.107. Фрагмент вікна форми *Музичні записи*.

3. Створюємо елементи керування формою. Для цього переходимо в режим *Конструктора* форми, за допомогою мишки змінюємо розміри *Області даних*, на панелі елементів вибираємо функційний елемент *Кнопка* та вставляємо його у область даних форми. Відкриється вікно створення кнопок.

У стовпці *Категорії* вибираємо пункт *Перехід по записах*, у стовпці *Дії* – *Знайти запис* – *Готово*. Повторюємо операцію для встановлення функційної кнопки *Друк запису*. Результат виконання наступний (рис. 5.108).

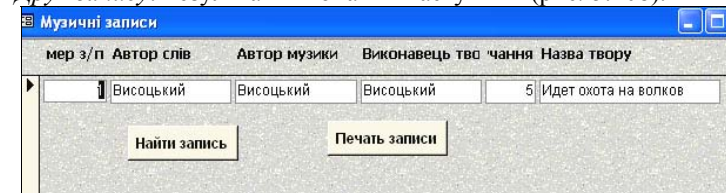


Рис. 5.108. Фрагмент вікна форми *Музичні записи* зі створеними функціональними кнопками.

4. Налаштовуємо параметри запуску бази даних наступним чином: пункт меню *Сервіс – Параметри запуску*, у полі Вивід форми вибираємо за основу форму *Музичні записи – ОК* (рис. 5.109).

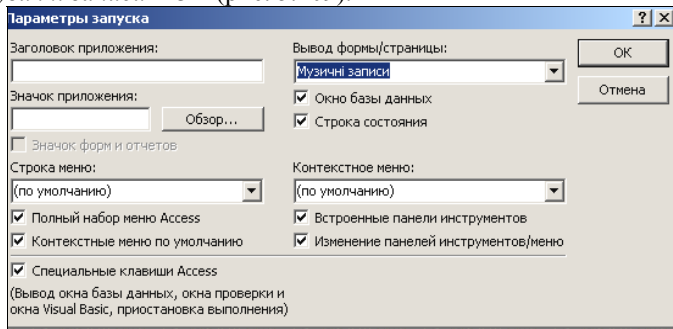


Рис. 5.109. Вікно налагодження параметрів запуску бази даних **Музичні записи**.

Закриваємо вікно програми та перезавантажуємо базу даних **Музичні записи**. У результаті під час завантаження бази відкривається уже відома форма *Музичні записи* для роботи з даною базою (див. рис. 5.107).

5. Створюємо запити для вибірки інформації за виконавцем. Для цього переходимо у поле *Запити, Створення запиту за допомогою Майстра*. В якості вихідного елемента вибираємо таблицю *Музичні записи*, додаємо всі доступні поля, *Далі – Далі*, вибираємо *Змінити макет запиту – Готово*.

Для стовпця Виконавець твору у полі – *Умова вибору* вводимо умову – "Вис*". Закриваємо запит із збереженням змін. У результаті отримуємо (рис. 5.110).

Поле:	Номер з/п	Автор слів	Автор музики	Виконавець твору	Час звучання
Имя таблицы:	Музичні записи	Музичні записи	Музичні записи	Музичні записи	Музичні записи
Сортировка:					
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Условие отбора:				Like "Вис**"	
или:					

Рис. 5.110. Фрагмент заповненої таблиці *Конструктора* запиту для вибору за полем Виконавець твору.

Закриваємо запит із збереженням змін. У результаті перегляду отримуємо (рис. 5.111).

Запрос_виконавец : запрос на выборку					
Номер з/п	Автор слів	Автор музики	Виконавець твору	Час звучання	Назва твору
1	Висоцький	Висоцький	Висоцький	5	Идет охота на волков
(Счетчик)				0	

Рис. 5.111. Фрагмент результату запиту, отриманого за допомогою QBE-таблиці, зображеної на рис. 5.65.

локальні мережі, а для поодиноких користувачів – персональні комп'ютери довільних моделей. Відомо, що різні типи процесорів мають різні системи команд. Відповідно, неможливо переносити програми з одного комп'ютера на інший (без спеціальних умов), а у рамках мережі Internet всі комп'ютери з успіхом "спілкуються" один з одним. Така сумісність пояснюється тим, що спілкування відбувається на рівні даних, тобто комп'ютери фактично обмінюються даними.

Формати даних добре стандартизовані. Так, наприклад, система кодування ASCII, діє на абсолютній більшості комп'ютерів і це забезпечує простоту пересилання та приймання текстів англійською мовою. Деяко гірша ситуація щодо кодування символів національних абеток та кодування графіки. У цьому випадку сумісності досягають шляхом перетворення даних за допомогою спеціальних програм.

Коли йдеться про роботу в Internet або про використання Internet, насправді йдеться не про Internet у цілому, а лише про одну або кілька з його численних служб. Залежно від конкретної мети і завдань користувачі використовують відповідні служби.

Якщо порівнювати з Internet залізничну мережу країни, тоді у ній можна відзначити службу пасажирських сполучень, поштово-багажну службу, службу доставляння промислових вантажів та численні регіональні служби руху приміських електропоїздів. Хоч ці служби використовують ті самі рейки, добре зрозуміла різниця у функціональному призначенні служб і не станемо користуватися ні службами приміського сполучення, ні службою доставляння вантажів, щоб проїхати з Києва до Львова. Теоретично це можливо, але практично дуже незручно.

Таким чином, можна сказати, що різні служби залізничного відомства керуються різними протоколами. Протокол визначає термін і порядок доставляння вантажів, пошти, людей. Протоколи різних служб визначають і різний характер стосунків між службою та клієнтом. Так, наприклад, протокол служби приміських пасажирських перевезень не передбачає надання спальних місць та забезпечення постільною білизною на час поїздки, навіть коли вона є тривалою.

Аналогічно працює й Internet. Різні служби мають різні протоколи (це не суперечить тому, що ми сказали про стек протоколів TCP/IP, який лежить в основі всього функціонування Internet). Маємо на увазі протоколи вищого рівня – прикладні протоколи. Дотримання їх забезпечує та підтримує роботу спеціальних програм. Таким чином, якщо ми хочемо скористатися довільною службою Internet, тоді повинні встановити на комп'ютері програму, здатну працювати за протоколами цієї служби. Такі програми називають клієнтськими, або просто – клієнтами.

Так, наприклад, для пересилання файлів у Internet використовують спеціальний протокол FTP (File Transfer Protocol). Якщо ми хочемо отримати з Internet файл, тоді ми повинні:

- мати на комп'ютері програму, яка є клієнтом FTP (FTP-клієнт);

Коміркові/персональні комунікаційні послуги (PCS). Доступ до Internet може бути забезпечений через чинні коміркові системи з використанням модемів VBD. Оскільки коміркові канали вузькосмугові, швидкості доступу обмежені до 9.6 кб/с для систем AMPS і TDMA і до 14.4 кб/с для систем CDMA. Коміркові цифрові пакети даних (Cellular Digital Packet Data – CDPD) – це технологія, яка дозволяє розширення швидкостей пересилання даних для AMPS до 19.2 кб/с. CDPD забезпечує вищі швидкості, вставляючи IP-пакети безпосередньо у комірковий канал, припиняючи пересилання голосового трафіку, внаслідок чого канал тимчасово вільний. Вивчаються технології, які забезпечують доступ до Internet та до інших послуг даних через PCS. Стандарти даних PCS досліджуються за підтримки технічного комітету ANSI T-1, TIA, ITU-T.

Наземне ширококомовлення. Багатоканальний багатоточковий розподільний сервіс (Multichannel Multipoint Distribution Service – MMDS), який також називають "безпроводними кабелями", може забезпечити доступ до Internet у напрямку "вниз" на відстанях до 50 км від центральної точки пересилання. Шлях "вниз" MMDS разом із шляхом "вгору" телефонним каналом забезпечує повну організацію доступу до Internet. MMDS оперує у частотному діапазоні 2 ГГц і має 33 канали, кожний з яких здатний до підтримання швидкостей даних "вниз" до 10 Мб/с. Технологічні дослідження скеровані на збільшення швидкості пересилання даних у майбутньому до 27 Мб/с.

Локальний багатоточковий розподільчий сервіс (Local Multipoint Distribution Service – LMDS) простіший від MMDS. LDMS використовує передавачі у діапазоні 28 ГГц, при чому кожен передавач охоплює відстань до 5 км. Відносно менша відстань пересилання пов'язана з фактом, що LDMS має у 4 рази більшу смугу від MMDS, а це дозволяє формувати істотно щільнішу мережу користувачів від MMDS.

Сателітне ширококомовлення. Окремі дослідження пропонують використання сателітів для забезпечення доступу від Internet "вниз". Ряд пропозицій базуються на використанні стаціонарних сателітів, тоді як інші використовують кластери (групи) сателітів. Пропоновані швидкості пересилання даних змінюються від низькошвидкісних каналів для окремих користувачів до спільних каналів із швидкостями понад 1 Мб/с. Перші широкодоступні системи працюють у діапазоні 12 ГГц і використовують швидкості 400 кб/с. Обладнання у розташуванні користувача складається з параболічної антени діаметром приблизно 52 см, мікрохвильового приймача і цифрової карти-декодера, яка монтується безпосередньо у комп'ютері. Сателітні системи також використовують телефонні канали для доступу "вгору".

6.6.4. Служби Internet. Сумісність на рівні даних

У фізичному розумінні Internet – це величезна сукупність комп'ютерів, локальних і глобальних мереж. Звичайно, говорити про якість стандартування обладнання недоцільно – кожен учасник мережі користується доступним для нього обладнанням. Для великих організацій це можуть бути

6. Створюємо звіт на основі побудованої бази. Для цього вибираємо об'єкт *Звіти – Створення звіту за допомогою Майстра*, додаємо всі необхідні поля, *Далі – Готово*. У результаті отримуємо (рис. 5.112).

Музичні записи				
Автор слів	Номер з/п	Автор музики	Виконавець творч. звучання	Назва твору
Ані Лорак	12	Ані Лорак	Ані Лорак	0 Мій ангел
	11	Ані Лорак	Ані Лорак	5 Мій герой
Вакарчук	8	Вакарчук	Океан Ельзи	4 Автомобіль
	5	Вакарчук	Океан Ельзи	6 Я так хочу до тебе
	4	Вакарчук	Океан Ельзи	5 Лукава кішка
Висоцький	1	Висоцький	Висоцький	5 Йде т охота на волк
Івасюк	3	Івасюк	Івасюк	5 Водограй
	2	Івасюк	Івасюк	4 Червона рута
Пономарьов	10	Пономарьов	Пономарьов	6 Любов-війна
	9	Пономарьов	Пономарьов	5 Милитва
	7	Пономарьов	Пономарьов	5 Зранку до ночі
Чубинський	6	Вербийський	Пономарьов	4 Гіми України

Рис. 5.112. Фрагмент звіту, створеного на основі бази **Музичні записи**

Завдання 7.

Створити базу даних **Фірма_кондитерських_виробів**, яка буде містити таблицю *Фірма* з такими полями: Порядковий номер, Найменування фірми, Адреса фірми, Найменування товару, Ціна товару, Наявність_товару, Продано_товару, Залишок_товару. На основі таблиці *Фірма* створити форму *Фірма* для вводу даних та встановити на формі елементи керування *Наступний запис*, *Попередній запис*, *Знищити запис*. Створити умову обчислення поля Залишок_товару у формі *Фірма*. Створити запит для вибірки товару за ціною в межах від 4 грн. до 7 грн. Створити запит для пошуку товару за найнижчою ціною.

Хід роботи:

1. Створюємо базу даних **Фірма_кондитерських_виробів**, яка буде містити таблицю *Фірма*. Для цього використовуємо пункт *Створення таблиці в режимі Конструктора* на закладці *Таблиця*. Обов'язковими полями є:

- Порядковий номер – тип даних Лічильник;
- Найменування фірми – тип даних Текстовий;
- Адреса фірми - тип даних Текстовий;
- Найменування товару - тип даних Текстовий;
- Ціна товару - тип даних Числовий;
- Наявність_товару - тип даних Числовий;
- Продано_товару - тип даних Числовий;
- Залишок_товару - тип даних Числовий;

Для поля Найменування фірми у полі властивостей *Загальні – Значення за замовчуванням* – вводимо назву фірми *Світоч* (рис. 5.113).

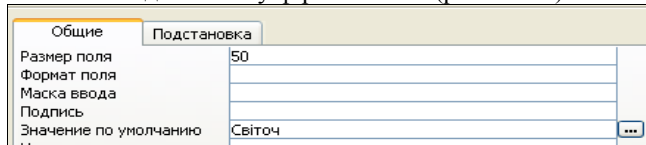


Рис. 5.113. Вікно встановлення значення Найменування фірми за замовчуванням.

Аналогічним чином вводимо для поля *Адреса фірми* значення за замовчуванням – *Кульпарківська, 15*.

У результаті отримуємо в режимі *Конструктора* готову таблицю *Фірма* (рис. 5.114).

Имя поля	Тип данных
№	Счетчик
Найменування_фірми	Текстовый
Адреса_фірми	Текстовый
Найменування_товарів	Текстовый
Ціна товару	Числовой
Наявність_товару	Числовой
Продано_товару	Числовой
Залишок_товару	Числовой

Рис. 5.114. Фрагмент вікна таблиці *Фірма* у режимі Конструктора.

Кабельні модеми можуть бути встановлені додатково до системи HFC або бути інтегрованою частиною HFC.

Комутована цифрова широкосмугова система (SDB). SDB класифікується як цифрова система в основній смузі частот із номінальною швидкістю потоку "вниз" до 50 Мб/с у режимі "точка-точка", що може бути розподілений між цифровим відео та даними. Для пересилання даних "вгору" швидкість становить 1.5 Мб/с. Не дивлячись на те, що система базується на змаганні за доступ, завжди наявна мінімальна гарантована швидкість для даних "вниз". Зазвичай, вона становить приблизно 16 кб/с. Всі три описані архітектури мають спроможність до пересилання як аналогового, так і цифрового відео в широко-мовному режимі.

Обидві системи, двоспрямована HFC і SDB є широкосмуговими і придатні до телефонії, відео, Internet-даних і PCS. Відзначимо, що розглянуті архітектури мають ряд окремих характеристик і компонент. Двоспрямована система HFC забезпечує розподіл у оптоволоконному вузлі, де сигнали збираються і розподіляються багатьма коаксіальними лініями, які охоплюють житловий район. Оптоволоконні вузли призначені для обслуговування від 500 до 2000 квартир.

Системи SDB допроваджують оптоволокну аж до користувача. У типових системах оптоволоконний кабель може бути оптично розгалужений. Оптичні мережеві пристрої (Optical Network Unit – ONU) є закінченнями оптоволоконного кабелю і забезпечують індивідуальні коаксіальні або типу "скручена пара" розгалуження до абонентів. Типовий ONU може обслуговувати від 4 до 60 квартир. Такі SDB забезпечують підведення оптоволоконна безпосередньо до користувача.

Безпроводні системи. Як подано на рис. 6.38, для забезпечення доступу до Internet із використанням безпроводних технологій застосовують три способи: сателітне ширококомовлення, наземне ширококомовлення і коміркові/персональні комунікаційні послуги (Personal Communications Services – PCS).

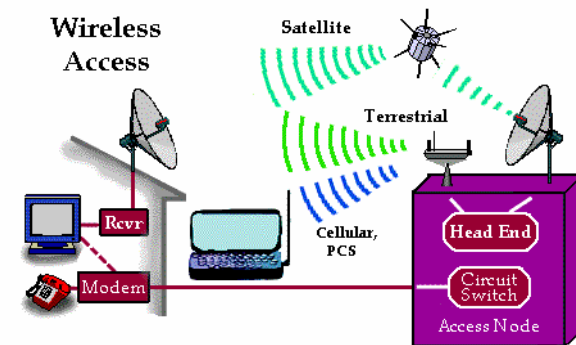


Рис. 6.38. Безпроводні технології доступу до Internet.

Механізми віддаленого доступу, такі як абонентська петля – транспортна система, створюють можливості для міграції оптоволоконних технологій. Послуги для кінцевих користувачів остаточно забезпечуються через "скручену пару" від віддалених терміналів, які в свою чергу, з'єднані через оптоволоконне обладнання з вузлом послуг. Віддалена система може бути під'єднана двома способами. Перший полягає у безпосередньому під'єднанні до комутатора – це називають **інтегрованим методом доступу**. Другий спосіб використовує під'єднання до головного цифрового терміналу у вузлі послуг – це називають **універсальним методом доступу**. Обидва вказані методи використовуються для забезпечення доступу до Internet. Вибір методу залежить від вмонтованої мережі та від можливостей, які повинні бути забезпечені для кінцевих користувачів додатково для доступу до Internet.

Існують три основні способи поєднання провідних кабельних та оптоволоконних систем, які використовуються для доступу до Internet (рис. 6.37):

- система кабельного телебачення та гібридна оптоволоконно-коаксіальна система (HFC);
- двоспрямована HFC;
- комотована цифрова широкополосна система (SDB).

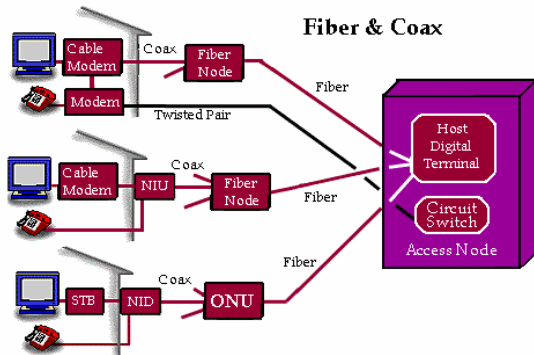


Рис. 6.37. Способи поєднання провідних кабельних та оптоволоконних систем.

Система кабельного телебачення та гібридна оптоволоконно-коаксіальна система (HFC). Традиційні системи мають широкополосні властивості лише у напрямку "вниз" і передають потік у широкополосному режимі в діапазоні від 50 МГц до 550/750 МГц з каналами 6 МГц. Кабельні модеми, які використовуються, дозволяють пересилання даних у системах HFC у напрямку "вниз". Можна реалізувати швидкість пересилання даних до 30 Мб/с у відеоканалі частотою 6 МГц. Сигнали "вгору" передають через чинний телефонний канал, використовуючи VBD або ISDN.

Закриваємо режим Конструктора та відкриваємо таблицю в режимі вводу даних. Вводимо 10 довільних записів (рис. 5.115).

№	Найменування_фірми	Адреса_фірми	Найменування_товарів	Ціна_товару	Наявність_товару	Продано_товару	Залишок_товару
1	Світоч	Кульпарківська 15	Шоколад з родзинками	7,5	100	40	60
2	Світоч	Кульпарківська 15	Молочний шоколад	7	125	10	115
3	Світоч	Кульпарківська 15	Шоколад з кокосовою ст.	7,65	245	56	189
4	Світоч	Кульпарківська 15	Горішки	3	50	10	40
5	Світоч	Кульпарківська 15	Чіпси	4	250	189	61
* (Счетчик)	Світоч	Кульпарківська 15		0	0	0	0

Рис. 5.115. Фрагмент вікна таблиці *Фірма*.

2. На основі таблиці *Фірма* створюємо форму *Фірма* для вводу даних, яка може містити наступні поля:

- Порядковий номер;
- Найменування фірми;
- Адреса фірми ;
- Найменування товару;
- Ціна товару;
- Наявність_товару;
- Продано_товару ;
- Залишок_товару.

У результаті отримуємо (рис. 5.116).

Рис. 5.116. Фрагмент вікна форми *Фірма* з першим записом.

3. Створюємо елементи керування формою. Для цього переходимо в режим *Конструктора* форми, за допомогою мишки змінюємо розміри *Області даних*, на панелі елементів вибираємо функційний елемент *Кнопка* та вставляємо його в область даних форми. Відкривається вікно створення кнопок.

У стовпці *Категорії* вибираємо пункт *Перехід по записам*, у стовпці *Дії – Наступний запис* – *Готово*. Повторюємо операцію для встановлення функційної кнопки *Попередній запис та Видалити запис*. Результат виконання наступний (рис. 5.117).

№	
Найменування_фірм	Світоч
Адреса_фірми	Купьпарківська 15
Найменування_товар	Шоколад з родзинками
Ціна товару	7,5
Наявність_товару	100
Продано_товару	40
Залишок_товару	0

Рис. 5.117. Фрагмент вікна форми *Фірма* зі вставленими елементами керування.

4. Створюємо умову обчислення поля Залишок_товару у формі *Фірма*. Для цього: відкриваємо форму в режимі *Конструктора*, для поля Залишок_товару через контекстне меню відкриваємо вікно *Властивості – Дані – Дані*, та за допомогою *Побудовувача* виразів описуємо наступну умову (рис. 5.118). Натискаємо ОК.

Построитель выражений

=[Наявність_товару]-[Продано_товару]

OK Отмена Назад

+ - / * & = > < <> And Or Not Like () Вставить Справка

Фірма	<Форма>	<Значение>
Таблицы	<Список полей>	AfterDelConfirm
Запросы	№_Надпись	AfterFinalRender
Forms	№	AfterInsert
Reports	Найменування_фірми_Надпи	AfterLayout
Функции	Найменування_фірми	AfterRender
Константы	Адреса_фірми_Надпись	AfterUpdate
Операторы	Адреса_фірми	AllowAdditions
Общие выражения	Найменування_товарів_Надг	AllowDatasheetView
	Найменування_товарів	AllowDeletions

Рис. 5.118. Вікно побудовувача виразів для обчислення поля Залишок_товару у формі *Фірма*.

Результат виконання подано на рис. 5.119.

№	
Найменування_фірм	Світоч
Адреса_фірми	Купьпарківська 15
Найменування_товар	Шоколад з родзинками
Ціна товару	7,5
Наявність_товару	100
Продано_товару	40
Залишок_товару	60

Рис. 5.119. Обчислене поле Залишок_товару у формі *Фірма*.

не у порівнянні з модемами у телефонному діапазоні. Модеми VBD мусять бути сумісними від одного кінцевого користувача до іншого, а модеми ADSL оперують тільки у проміжку від кінцевого користувача до мережевого провайдеру.

Надвисокошвидкісна технологія DSL (Very high-speed DSL – VDSL) передбачає значно вищі швидкості у порівнянні з ADSL, однак на значно меншій відстані. Стандарти опрацьовують чотири різні організації: ANSI, ADSL Forum, ATM Forum і DAVIC (Digital Audio-Visual Council). Запропоновано чотири різні технології (CAP, DMT, DWMT і SLC) з метою зменшення потужності і вартості у порівнянні з ADSL.

R-ADSL. Як видно з назви, модеми технології цифрових абонованих ліній з адаптуванням швидкості (Rate-Adaptive DSL – R-ADSL) підлаштовують швидкість пересилання даних до якості з'єднання через "скручену пару". Пропоноване програмне забезпечення здійснюватиме це автоматично, з мінімальним втручанням адміністратора.

HDSL і SDSL. Модеми високошвидкісної DSL (High data-rate DSL – HDSL) передають із швидкістю 1.544 Мб/с в обох напрямках. Якщо використовують дві "скручені пари", тоді половина трафіку пересилається кожною парою. Застосовуючи три "скручені пари", можна пересилати зі швидкістю 2.048 Мб/с по 1/3 трафіку через кожену пару. Обмеження на відстань становить 4 км при діаметрі провідників 24 AWG (0.511 мм) і 3 км для провідників діаметром 26 AWG (0.404 мм).

SDSL відноситься до DSL через одну пару (single-pair DSL) або симетричної DSL (symmetric DSL). Діапазон швидкостей SDSL становить від $n \times 64$ кб/с до 2.048 Мб/с в обох напрямках.

HDSL і SDSL розглядаються скоріше як заміна виділених ліній T-1 з малими витратами, ніж як засіб доступу за місцем проживання.

Всі три основні технології доступу, описані вище, можуть бути застосовані для віддаленого доступу, як це подано на рис. 6.36.

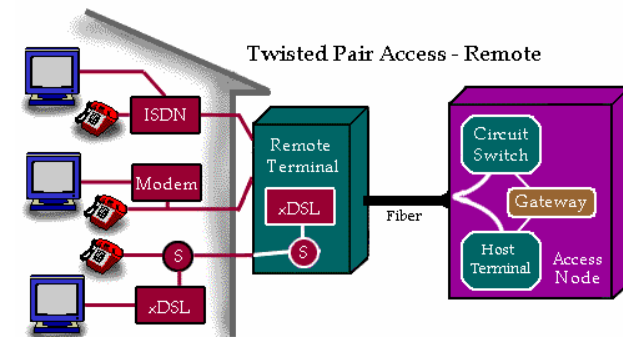


Рис. 6.36. Віддалений доступ при застосуванні кабелів типу "скручена пара".

Модеми VBD добре відомі і використовуються індивідуальними користувачами та користувачами зі сфери малого бізнесу. Вони функціонують у діапазоні частот телефонії через кабелі "скручена пара", використовуючи фазову або квадратурну модуляцію для пересилання даних.

ISDN – це цифрова широкосмугова технологія, яка оперує з двосторонньою швидкістю 144 кб/с, використовуючи схему кодування 2B1Q, і містить два канали В по 64 кб/с і один канал D (16 кб/с). Канали В можна використати для двох телефонних викликів, двох каналів пересилання даних, окремого пересилання голосу і даних, або для поєданого пересилання даних. Для стандартного кабелю довжина абонентської петлі обмежена до 6 км.

Ряд технологій xDSL стандартизовані або знаходяться у процесі стандартизування в ANSI та ADSL Forum. Як подано на рис. 6.35, найвищі швидкості для замовників досягаються на малих відстанях від xDSL-модему провайдера. Такі модеми повинні бути розміщені у центральному офісі провайдера або у віддалених точках доступу поблизу більшості кінцевих користувачів.

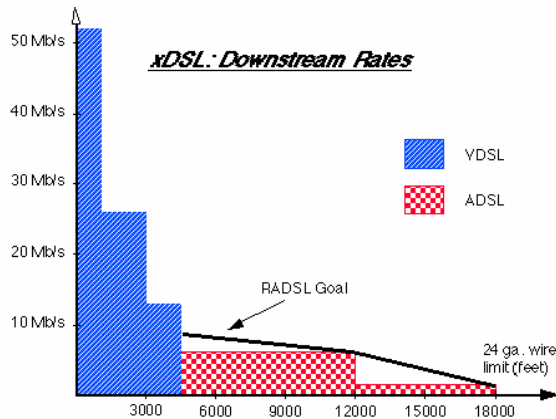


Рис. 6.35. Швидкості потоку "вниз" для різних технологій xDSL.

ADSL (асиметрична DSL) є однією з багатьох типів технологій xDSL. Головними стандартами ADSL є ADSL-1, який визначає швидкість потоку "вниз" 1.5...2 Мб/с і швидкість потоку "вгору" 16...64 кб/с; ADSL-3 із швидкістю "вниз" до 6.144 Мб/с і з двосторонніми каналами - до 640 кб/с. Якісні лінії типу "скручена пара" без перемичок можуть підтримувати швидкості ADSL-1 на відстанях до 6 км (з провідниками AWG-24, тобто діаметром 0.511 мм) і швидкості ADSL-3 – до 4 км.

ANSI та ADSL Forum впровадили технології дискретної багатотонавої модуляції. Однак технологія амплітудно-фазової модуляції без носія (Carrierless Amplitude and Phase – CAP) поширена у лініях ADSL приблизно у 30 разів більше, ніж CAP. Модеми DMT і CAP несумісні, але це питання мало актуаль-

5. Створюємо запити для вибірки інформації за полем Найменування_товару. Для цього переходимо у поле *Запити, Створення запиту за допомогою Майстра*. В якості вихідного елемента вибираємо таблицю *Фірма*, додаємо всі доступні поля, *Далі – Далі*, вибираємо *Змінити макет запиту – Готово*.

Для стовпця Найменування_товарів у полі *Умова вибору* вводимо умову – “Шоколад*” (як приклад). Закриваємо запит із збереженням змін. У результаті отримуємо такий запит (рис. 5.120).

Поле:	№	Найменування_фірм	Адреса_фірми	Найменування_тов	Ціна_товару
Имя таблицы:	Фірма	Фірма	Фірма	Фірма	Фірма
Сортировка:					
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Условие отбора:				Like "шоколад*"	

Рис. 5.120. Фрагмент таблиці QBE-запиту для вибірки за назвою товару.

Після запуску даного запиту отримаємо наступний результат (рис. 5.121).


№	Найменування_	Адреса_фірми	Найменування_товарів	Ціна_товару	Наявність_топ
1	Світоч	Кульпарківська 15	Шоколад з родзинками	7,5	100
3	Світоч	Кульпарківська 15	Шоколад з кокосовою стружк	7,65	245
тчик)	Світоч	Кульпарківська 15		0	0

Рис. 5.121. Фрагмент вікна з результатами вибірки за QBE-запитом, поданим на рис. 5.120.

6. Створюємо запит для вибірки даних за Ціною товару у межах від 4 грн. до 7 грн. Для цього переходимо у поле *Запити, Створення запиту за допомогою Майстра*. В якості вихідного елемента вибираємо таблицю *Фірма*, додаємо всі доступні поля, *Далі – Далі*, вибираємо *Змінити макет запиту – Готово*. Для стовпця Ціна товару у полі *Умова вибору* вводимо умову – *Between 4 And 7* (як приклад). Закриваємо вікно *Майстра* запитів із збереженням змін. Запускаємо запит на виконання і в результаті отримуємо (рис. 5.122).

№	Найменування_фірми	Адреса_фірми	Найменування_товарів	Ціна_товару
2	Світоч	Кульпарківська 15	Молочний шоколад	7
5	Світоч	Кульпарківська 15	Чіпси	4
(Счетчик)	Світоч	Кульпарківська 15		0

Рис. 5.122. Фрагмент вікна з результатами запиту за Ціною товару в межах від 4 грн. до 7 грн.

7. Створюємо запит для вибірки товару за найнижчою вартістю. Для цього переходимо у поле *Запити, Створення запиту за допомогою Майстра*. В якості вихідного елемента вибираємо таблицю *Фірма*, додаємо всі доступні поля, *Далі – Далі*, вибираємо *Змінити макет запиту – Готово*. На панелі інструментів активуємо кнопку *Підсумки* , в результаті додається ще один рядок QBE-таблиці, що має назву *Групова операція*. У всіх полях QBE-таблиці

вибираємо функцію Min (як приклад). Закриваємо запит із збереженням змін. У результаті отримуємо (рис. 5.123).

Фірма Запрос3 : запит на виборку						
Min-№	Min-Найменує	Min-Адреса_ф	Min-Найменує	Min-Ціна това	Min-Наявність	Min-Продано
1	Світоч	Кульпарківська	Горішки	3	50	10

Рис. 5.123. Фрагмент результату виконання запиту на пошук товару за найнижчою ціною.

8. Створюємо звіт на основі побудованої бази. Для цього вибираємо об'єкт *Звіти – Створення звіту за допомогою Майстра*, додаємо всі необхідні поля, *Далі – Готово*. В результаті отримуємо (рис. 5.124).

Фірма							
№	Найменування_ф	Адреса_фір	Найменуван	Ціна товару	Наявність	Продано	Залишок
1	Світоч	Кульпарківська	Шоколад з родз	7,5	100	40	0
2	Світоч	Кульпарківська	Молочний шоко	7	125	10	0
3	Світоч	Кульпарківська	Шоколад з коко	7,85	245	56	0
4	Світоч	Кульпарківська	Горішки	3	50	10	0
5	Світоч	Кульпарківська	Чіпси	4	250	189	0

Рис. 5.124. Фрагмент звіту, створений на основі бази *Фірма_кондитерських_виробів*.

Коаксіальні та оптоволоконні системи початково були застосовані для широкомовного відео. Оскільки цим системам притаманна широкосмуговість, тому були опрацьовані технології, які використовують цю перевагу для швидкісного пересилання даних, особливо для доступу до Internet у домашніх умовах.

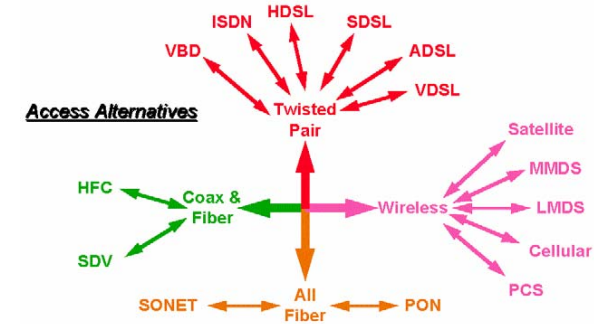


Рис. 6.33. Альтернативи для доступу до Internet.

Безпроводний доступ до Internet має два види:

- сателітні системи, створені для широкомовного відео, які мають спроможність пересилати дані з високою швидкістю;
- коміркові/персональні комунікаційні послуги (Personal Communications Services – PCS) – системи, проєктовані для обслуговування мобільних користувачів.

Переважаюча кількість систем доступу для користувачів з сфери бізнесу використовують оптоволоконні системи SONET у США і SDH у Європі. У майбутньому очікується застосування пасивних оптичних мережевих систем для отримання повністю оптоволоконного доступу користувачами у домашніх умовах.

Існують три головні категорії застосувань для кабелів типу "скручена пара" при їх застосуванні для доступу до Internet (рис. 6.34):

- модеми в смузді частот телефонії (голосу) (Voice band data – VBD);
- цифрові абоновані лінії (Digital Subscriber Line – DSL) ISDN;
- інші методи DSL (xDSL).

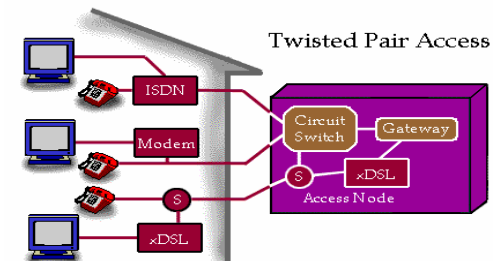


Рис. 6.34. Дані у діапазоні телефонії та ISDN.

Зауважимо, що дві останні групи можуть поєднуватися, наприклад, ряд компаній надають послуги як телефонії, так і доступу до Internet.

Повертаючись до питання про те, як отримати доступ до Internet, відзначимо, що для провайдерів телефонії, послуг безпроводного та кабельного зв'язку доступом є мережеве з'єднання від розташування користувача до зовнішньої точки під'єднання у вузлі послуг. У традиційній телефонній архітектурі це частіше називають точкою під'єднання провідного кабелю до кросу, що відноситься до головної розподільчої системи (Main Distribution Frame – MDF). Решта мережі провайдера відноситься до мережі комутування та транспортування.

На відміну від цього, з точки зору Internet-провайдера (Internet Service Provider – ISP), доступ розглядається як з'єднання від замовника до мережі провайдера, точніше, від місця локації замовника до вузла доступу до шлюзу, який обслуговує ISP. Для ISP телефонний доступ, комутування і транспортна мережа є частинами "доступу".

Позиція провайдерів телефонії, послуг безпроводного та кабельного зв'язку є центральною при розгляді новітніх та пропонованих IT.

6.6.3. Розвиток засобів доступу

Розвиток засобів доступу відбувається, передовсім, під впливом вимог щодо збільшення ширини смуги для підтримки зростаючої кількості користувачів. Перед 1994 роком трафік, який пересилався через Internet, складався, переважно, з текстової інформації, а найбільш поширеними видами послуг були пересилання файлів та електронна пошта.

Зростання обсягів інформації, яка передається Internet від 1995 року, частково обумовлене графічною природою World Wide Web (WWW), що у свою чергу пов'язане з тим, що графічні зображення у цифрових форматах складаються з великої кількості бітів. Пересилання таких файлів вимагає значно більших швидкостей, ніж це необхідно для порівняно малих текстових файлів. WWW також створив базу для нарощування інших застосувань, таких як анімована графіка, аудіо- та відео. Це, у свою чергу, створює потребу у збільшенні можливостей високошвидкісного доступу.

Технології доступу до Internet можна охопити чотирма великими категоріями:

- з використанням провідних кабелів типу "скручена пара";
- з використанням оптоволоконних і коаксіальних кабелів;
- з використанням безпроводних систем;
- з використанням виключно оптоволоконних систем.

Як подано на рис. 6.33, окремі технології та впровадження містяться всередині цих широких категорій.

Телефонні лінії типу "скручена пара" є середовищем доступу, яке сьогодні застосовується більшістю користувачів. Незабаром будуть впроваджені інші IT, які забезпечать більшу швидкість пересилання даних цим середовищем.

РОЗДІЛ 6. ПРОГРАММИ, ТОПОЛОГІЇ ТА ТЕХНОЛОГІЇ

Концепція комп'ютерних мереж (КМ) є логічним результатом еволюції інформаційних технологій (IT). Перші комп'ютери 50-х років минулого століття – великі, громіздкі і дорогі – призначалися для дуже невеликого кола обраних користувачів. Такі комп'ютери не були призначені для інтерактивної роботи, а використовувалися у режимі пакетного оброблення даних.

Система пакетного оброблення, як правило, формувалася на базі мейнфрейму – потужного і надійного комп'ютера універсального призначення. Користувачі готували перфокарти, які містили дані і команди програм, передавали їх до обчислювального центру. Оператори вводили ці перфокарти у спеціальний пристрій зчитування, після чого дані вводилися у оперативну пам'ять комп'ютера, а роздруковані результати користувачі, як правило, отримували наступного дня. Звичайно, для користувачів інтерактивний режим роботи, при якому можна з терміналу оперативно керувати процесом оброблення своїх даних, був би набагато зручнішим. Але інтересами користувачів на перших етапах розвитку обчислювальних систем значною мірою нехтували, оскільки пакетний режим – це найефективніший режим використання обчислювальних потужностей, так як він дозволяє виконати за одиницю часу більше задач, ніж довільні інші режими. Основним показником вважалася ефективність роботи найдорожчого пристрою обчислювальної системи – процесора, а інтереси користувачів вважались другорядними.

У міру зниження вартості процесорів на початку 60-х років ХХ століття появилися нові способи організації обчислювальних процесів, які надавали можливість врахувати інтереси користувачів. Почали розвиватися інтерактивні багатотермінальні системи розподілу часу. У таких системах комп'ютер надавався у розпорядження відразу кільком користувачам. Кожний з користувачів отримував у своє розпорядження термінал, з допомогою якого він міг працювати у діалоговому режимі. Хоча обчислювальні потужності комп'ютерів залишалися повністю централізованими, деякі функції – такі як ввід і вивід даних – стали розподіленими. Такі багатотермінальні централізовані системи ззовні уже нагадували локальні обчислювальні мережі. Дійсно, пересічний користувач роботу за терміналом мейнфрейму сприймав приблизно так само, як сьогодні він сприймає роботу за під'єднаним до мережі персональним комп'ютером. Користувачі отримали можливість доступу до спільних файлів і периферійних пристроїв.

У середині 80-х років стан справ у локальних мережах почав радикально змінюватися. Утвердилися стандартні технології об'єднання комп'ютерів у мережу – Ethernet, Arcnet, Token Ring. Потужним стимулом для їх розвитку послужили перші персональні комп'ютери, які стали ідеальними елементами для формування мереж – з одного боку вони були достатньо потужними для роботи мережевого програмного забезпечення, а з другого – явно потребували об'єднання своєї обчислювальної потужності для розв'язання складних задач, а також розподілу дорогих периферійних пристроїв і дискових масивів. Стан-

дартні мережеві технології перетворили процес побудови локальних мереж з мистецтва у творчу працю.

Комп'ютерні мережі (КМ) відносяться до розподілених (децентралізованих) обчислювальних систем. Оскільки основною ознакою розподіленої обчислювальної системи є наявність кількох центрів оброблення даних, тому поряд з комп'ютерними мережами до розподілених систем відносять також мультипроцесорні комп'ютери і багатомашинні обчислювальні комплекси.

У комп'ютерних мережах зв'язок між комп'ютерами здійснюється за допомогою спеціальних периферійних пристроїв – мережевих адаптерів, з'єднаних каналами зв'язку значної довжини. Взаємодія між комп'ютерами мережі здійснюється за рахунок пересилання повідомлень через мережеві адаптери і канали зв'язку. За допомогою цих повідомлень один комп'ютер здійснює запит доступу до локальних ресурсів іншого комп'ютера. Такими ресурсами можуть бути як дані на магнітних носіях, так і різноманітні периферійні пристрої – принтери, модеми, факс-апарати та ін. Розподіл локальних ресурсів кожного комп'ютера між усіма користувачами мережі – основна мета створення мережі.

Яким же чином відбивається на користувачах той факт, що їх комп'ютери під'єднані до мережі? Перш за все, користувач може користуватися не тільки файлами, дисковим простором, принтерами та іншими ресурсами свого комп'ютера, але і аналогічними ресурсами інших комп'ютерів, під'єднаних до цієї ж мережі. Для цього недостатньо забезпечити комп'ютери мережевими адаптерами і з'єднати їх кабельною системою. Необхідні ще деякі програмні додатки до операційних систем (OS) цих комп'ютерів. На тих комп'ютерах, ресурси яких повинні бути доступними усім користувачам мережі, необхідно додати модулі, які постійно будуть знаходитися у режимі очікування запитів, які поступають мережею від інших комп'ютерів. Такі модулі прийнято називати програмними **серверами** (server), оскільки головна їхня задача – обслуговувати (serve) запити на доступ до ресурсів свого комп'ютера. На комп'ютерах, користувачі яких хочуть отримувати доступ до ресурсів інших комп'ютерів, також необхідно додати до OS спеціальні програмні модулі, які повинні генерувати спеціальні запити на доступ до віддалених ресурсів і пересилати їх мережею на потрібний комп'ютер. Такі модулі прийнято називати програмними **клієнтами** (client). Власне мережеві адаптери та канали зв'язку розв'язують у мережі достатньо просту задачу – вони передають повідомлення із запитами і відповідями від одного комп'ютера до іншого, а основну роботу з організації сумісного використання ресурсів виконують клієнтські і серверні частини OS.

Пара модулів "клієнт-сервер" забезпечують сумісний доступ користувачів до певного типу ресурсів, наприклад, до файлів. У цьому випадку говорять, що користувач має справу з файловою **службою** (service). Як правило, мережева ОС підтримує декілька видів мережевих служб для своїх користувачів – файлову службу, службу друку, службу електронної пошти, службу віддаленого доступу та ін.

зв'язала кілька великих наукових, дослідних та освітніх центрів. Її основною метою було координування зусиль груп колективів, які працювали над єдиними науково-технічними проектами, а основним призначенням став обмін електронною поштою та файлами з науковою та проектно-конструкторською документацією.

Мережа ARPANET почала працювати в 1969 році. Нечисленні вузли, які входили до неї на той час, були пов'язані "виділеними лініями". Приймання та висилання інформації забезпечувалася програмами, які працювали на "вузлових" комп'ютерах. Мережа постійно розширювалася за рахунок під'єднання нових вузлів, на початок 80-х років на базі найбільших вузлів були створені регіональні мережі, які у цілому відтворювали загальну архітектуру ARPANET на нижчому рівні (у регіональному або локальному масштабі).

Щоразу, коли ми говоримо про інформаційні технології, треба мати на увазі принцип єдності апаратного та програмного забезпечення. Доки глобальне розширення ARPANET відбувалося за рахунок механічного під'єднання все нових і нових апаратних засобів (вузлів мережі), до Internet, у сучасному розумінні цього слова, було ще дуже далеко. Справжнє народження Internet відбулося у 1983 році, коли сталися революційні зміни у програмному забезпеченні комп'ютерного зв'язку. Днем народження Internet, як ми його розуміємо зараз, стала дата стандартизування протоколу TCP/IP, який лежить в основі Всесвітньої мережі й нині.

6.6.2. Доступ до Internet

Кількість користувачів Internet швидко зростає. Вимоги до мереж загального користування (публічних мереж), особливо до мереж доступу, достатньо високі. Однак технологічні перспективи швидко розширюються, створюючи публічним мережам можливість відігравати головну роль у наданні нових видів телекомунікаційних послуг замовникам. На даний час, доступ через провідні кабелі типу "скручена пара" за допомогою модемів, які працюють у частотному діапазоні телефонії, є нормою для користувачів. З використанням новітніх ІТ для користувачів стануть доступними ще більш швидкості доступу та обміну.

Розглянемо чинні та перспективні технології доступу до Internet.

Доступ до Internet здійснюється різними способами для різних груп користувачів. Ці групи розрізняють за їх приналежністю до послуг Internet:

- кінцеві користувачі, які можуть мати доступ до Internet, а також до інших видів послуг, таких як телефонія або кабельне телебачення;
- провайдери послуг даних, які надають доступ до Internet, послуги, подібні до AOL, або інші послуги пересилання даних, такі як віртуальна приватна мережева взаємодія;
- провайдери телефонії, послуг безпроводного та кабельного зв'язку, які можуть забезпечити з'єднання між користувачами і провайдерами послуг даних.

METRIC - вказує метрику, тобто вартість витрат для досягнення відзначеної адреси отримувача. Вартість витрат вимірюється у транзитних ділянках (hop). Кількість транзитних ділянок, яка дорівнює 1, вказує на те, що отримувач знаходиться в LAN. Додаткова транзитна ділянка вводиться в якості додаткових витрат з висилання пакету кожен раз, коли пакет проходить через маршрутизатор.

Наприклад, додавання нового маршруту:

```
route ADD 194.124.12.0 MASK 255.255.255.0 150.50.70.1 METRIC 3 IF 1,  
де IF 1 – перший інтерфейс.
```

Якщо параметр IF не задається, тоді хост намагається знайти інтерфейс самостійно для вказаного шлюзу за замовчуванням.

6.6. Мережа INTERNET

Дослівно слово Internet означає міжмережа, тобто у вузькому розумінні Internet – це об'єднання мереж. Однак останнім часом у цього слова з'явилося й ширше значення. Internet – це всесвітня комп'ютерна мережа. Internet можна розглядати, з фізичного боку, як кілька мільйонів комп'ютерів, пов'язаних між собою найрізноманітнішими лініями зв'язку, однак такий підхід є неточним. Доцільніше говорити про Internet як абстрактне поняття.

Internet, також, не прийнято розглядати як сукупність прямих з'єднань між комп'ютерами. Так, наприклад, якщо два комп'ютери, які є на різних континентах, обмінюються інформацією в Internet – це зовсім не означає, що між ними постійно діє одне пряме з'єднання. Дані, які вони посилають один одному, розбивають на пакети, і навіть в одному сеансі зв'язку різні пакети одного повідомлення можуть пройти різними маршрутами. Яким би маршрутом не рухалися пакети даних, вони все одно досягнуть пункту призначення і будуть зібрані у цілий документ. До того ж, пізніше відправлені дані можуть надійти раніше, але це не завадить правильно зібрати документ, оскільки кожен пакет має своє маркування.

Таким чином, Internet є немовби "простором", усередині якого здійснюється безперервна циркуляція даних. У цьому значенні його можна порівняти з теле- та радіоэфіром, хоч є й очевидна відмінність, тому що у ефірі ніяка інформація зберігатися не може, а в Internet вона переміщується між комп'ютерами, які становлять вузли мережі, і деякий час зберігається.

6.6.1. Основи Internet

Ранні експерименти щодо приймання та пересилання інформації за допомогою комп'ютера розпочалися ще в 50-ті роки XX ст. і мали лабораторний характер. Лише наприкінці 60-х років коштами Агентства Перспективних Розробок Міністерства оборони США (DARPA – Defense Advanced Research Project Agency) було створено першу мережу національного масштабу. За назвою агентства вона отримала назву ARPANET. Ця мережа

У технічній літературі англomовний термін service звичайно перекладається як "служба", "сервіс" або "послуга". Доволі часто ці терміни використовують як синоніми. У той же час окремі фахівці розрізняють термін "служба" і терміни "сервіс" або "послуга". Під "службою" розуміється мережевий компонент, який реалізує деякий набір послуг, а "сервісом" називають опис цього набору послуг, який надається даною службою. Таким чином, сервіс – це інтерфейс між споживачем послуг і постачальником послуг (службою). Надалі будемо послуговуватися терміном "служба" в усіх випадках, коли різниця у значенні цих термінів не носить принципового характеру.

Терміни "клієнт" і "сервер" використовуються не тільки для визначення програмних модулів, але і комп'ютерів, з'єднаних у мережу. Якщо комп'ютер надає свої ресурси іншим комп'ютерам мережі, тоді він називається **сервером**, а якщо він їх споживає – **клієнтом**. Деколи один і той же комп'ютер може одночасно відігравати роль і сервера, і клієнта.

6.1. Розподілені програми

Мережеві служби завжди є розподіленими програмами. Розподілена програма – це програма, яка складається з декількох взаємопов'язаних частин (на рис. 6.1. – з двох), де кожна частина, як правило, виконується на окремому комп'ютері мережі.



Рис. 6.1. Взаємодія частин розподіленого додатку.

До цього часу мова йшла про системні розподілені програми. Однак, у мережі можуть виконуватися і розподілені програми користувачів – додатки. Розподілений додаток також складається з декількох частин, кожна з яких виконує визначену завершену роботу з розв'язання прикладної задачі. Розподілені додатки повною мірою використовують потенційні можливості розподіленої оброблення, які надаються обчислювальною мережею, тому доволі часто називаються мережевими додатками.

Зауважимо, що не кожен додаток, який виконується у мережі, є мережевим. Існує велика кількість популярних додатків, які не є розподіленими і цілком виконуються на одному комп'ютері мережі. Тим не менше і такі додатки можуть використовувати переваги мережі за рахунок вмонтованих в OS мережевих служб. Власне, розвиток локальних мереж пов'язаний з використанням таких нерозподілених додатків.

6.1.1. Основні програмні і апаратні компоненти мережі

Навіть у результаті поверхневого аналізу роботи у КМ стає зрозумілим, що комп'ютерна мережа – це складний комплекс взаємопов'язаних і узгоджено працюючих програмних і апаратних компонент. Вивчення мережі у цілому передбачає знання принципів роботи її окремих елементів, зокрема:

- комп'ютерів;
- комунікаційного обладнання;
- операційних систем;
- мережевих додатків.

Весь комплекс програмних і апаратних засобів мережі можна описати багаторівневою моделлю. В основу функціонування довільної мережі закладено стандартизовані комп'ютерні платформи, які формують перший рівень. Часто у мережах з успіхом використовуються комп'ютери різних класів, набір яких відповідає класам задач, які розв'язуються мережею.

Другий рівень – це комунікаційне обладнання. Хоча комп'ютери і є центральними елементами оброблення даних у мережах, останнім часом менш важливу роль стали відігравати комунікаційні пристрої. Кабельні системи, повторювачі, мости, комутатори, маршрутизатори з допоміжних компонент мережі перетворилися в основні поряд з комп'ютерами і системним програмним забезпеченням, як за впливом на характеристики мережі, так і у вартісному відношенні. Комунікаційним пристроєм може бути складний спеціалізований мультипроцесор, який необхідно конфігурувати, оптимізувати і адмініструвати. Вивчення принципів функціонування комунікаційного обладнання вимагає ґрунтовних знань великої кількості протоколів, які використовуються у локальних і глобальних мережах.

Третім рівнем, який формує програмну платформу мережі є OS. Від того, які концепції управління локальними і розподіленими ресурсами закладені в основу мережевої OS, залежить ефективність функціонування усієї КМ. На стадії проектування мережі важливо врахувати наскільки просто дана OS може взаємодіяти з іншими OS мережі, наскільки вона забезпечує безпеку і захищеність даних, до яких границь вона дозволяє нарощувати кількість користувачів (масштабування КМ) та багато інших міркувань.

Найважливішим рівнем мережевих засобів є мережеві додатки, такі як мережеві бази даних, поштові системи, засоби архівування і захисту даних, системи автоматизування колективного доступу до даних.

6.1.2. Механізм взаємодії комп'ютера з периферійними пристроями

У механізмах взаємодії комп'ютерів у КМ багато чого запозичено від схеми взаємодії комп'ютера з периферійними пристроями. Зважаючи на цю обставину, здійснимо аналіз принципів функціонування мережі з цієї схеми.

Для обміну даними між комп'ютером і периферійним пристроєм (ПП) у комп'ютері передбачено зовнішній **інтерфейс**, тобто набір провідників, які

tracert.exe - програма для визначення маршруту до вказаного хосту.

Формат:

tracert [-d] [-h maximum_hops] [-j host-list] [-w timeout] target_name

Опції:

-d - не здійснювати вирішення адрес (не задіювати DNS протокол);

-h maximum_hops - максимальна кількість транзитних ділянок при пошуку цільового хосту;

-w timeout - таймаут в мілісекундах для очікування на кожну відповідь.

ipconfig.exe - програма для встановлення TCP/IP конфігурування на хості під управлінням Windows NT.

Формат:

ipconfig

/? - показати допоміжну інформацію про синтаксис команди;

/all - показати повну інформацію про конфігурування TCP/IP на хості;

/release - звільнити оренду IP-адреси для вказаного адаптера (протокол DHCP);

/renew - поновити оренду IP-адреси для вказаного адаптера.

winipcfg.exe - програма для встановлення TCP/IP конфігурування на хості під управлінням Windows (має графічний інтерфейс).

route.exe - програма для встановлення таблиці маршрутизації на хості.

Формат:

route [-f] [-p] [command [destination] [MASK netmask] [gateway] [METRIC metric] [IF interface]

-f - очищення усіх записів таблиці маршрутизування. Якщо використовується разом з однією з команд, тоді таблиця очищується перед тим, як виконується команда;

-p - коли використовується з командою ADD, тоді записується маршрут, який зберігається після перезавантаження комп'ютера. За замовчуванням нові маршрути після перезавантаження не зберігаються. Коли використовується з командою PRINT, тоді показує список зареєстрованих постійних (persistent) маршрутів. Ігнорується для інших команд, бо вони мають відношення лише до постійних маршрутів. Ця опція не працює на Windows'95 хості;

command - одна з наступних:

PRINT - друкує маршрут;

ADD - додає маршрут;

DELETE - ліквідує маршрут;

CHANGE - модифікує існуючий маршрут;

destination - вказує хост;

MASK - вказує, що наступним параметром є значення netmask;

netmask - вказує значення маски підмережі для вказаного маршруту. Якщо маска не вказана, тоді значення за замовчуванням дорівнює 255.255.255.255;

gateway - вказує шлюз за замовчуванням;

interface - номер інтерфейсу для вказаного маршруту;

6.5.5. Протокол IPv6, формат IP-адреси

Існуючому протоколу IPv4 притаманний ряд недоліків, серед яких відзначимо брак вільного адресного простору. У новій версії протоколу IP (IPv6) втілено ряд ідей з поновлення IP.

IPv6 створювався спеціально для розв'язання таких проблем, як розширення адресного простору, забезпечення достовірності та конфіденційності інформації, яка передається, підтримки пересилання трафіку реального часу, механізму додавання нових можливостей. Ми розглянемо лише адресування IPv6.

У новій версії протоколу розрядність адрес відправників та отримувачів збільшена з 32 розрядів до 128 розрядів. Формат запису IP-адреси виглядає наступним чином

443F:FFGA:1236:E09I:3409:1403:56C4:700A

Окрім існування звичайних (unicast) IP-адрес, призначених для зв'язку точка – точка, та групових (multicast) IP-адрес, появляється новий тип – невизначені (anycast) IP-адреси.

Хости, які використовують виключно протокол IPv4, не зможуть взаємодіяти з хостами, які використовують протокол IPv6, оскільки підтримується лише зворотня сумісність.

6.5.6. Основні утиліти для конфігурування та діагностування TCP/IP

ping.exe - програма для тестування зв'язку із визначеним хостом.

Формат:

ping [-t] [-a] [-n count] [-l size] [-f] [-i TTL] [-v TOS] [-r count] [-s count] [[-j host-list] | [-k host-list]] [-w timeout] destination-list

Опції:

- t - "пінгувати" відзначений хост до моменту переривання;
- a - привести відповідну текстовій числову або відповідну числовій текстову адреси (здійяти DNS протокол для визначення адрес);
- n count - кількість ехо-запитів, які посилаються;
- l size - розмір буфера посилання;
- f - встановити прапорець "не робити фрагментування" у пакеті;
- i TTL - час життя;
- v TOS - тип сервісу;
- r count - записати маршрут для кількості count транзитних ділянок;
- s count - позначка часу для count транзитних ділянок;
- j host-list - зробити довільним маршрут від джерела вздовж списку хостів;
- k host-list - зробити "жорстким" маршрут від джерела вздовж списку хостів;
- w timeout - таймаут у мілісекундах для очікування на кожну відповідь.

з'єднують комп'ютер і периферійний пристрій, а також набір правил та угод для обміну інформацією цими провідниками. Прикладами інтерфейсів, які використовуються у комп'ютерах, є паралельний інтерфейс Centronics, який призначений для під'єднання принтерів, та послідовний інтерфейс RS-232C, через який під'єднується "мишка", модем та ін. Інтерфейс реалізується зі сторони комп'ютера сукупністю апаратних і програмних засобів: контролером ПП та спеціальною програмою, яка керує цим контролером і яку прийнято називати **драйвером** відповідного ПП.

Зі сторони ПП інтерфейс, найчастіше, реалізується апаратним пристроєм управління, хоча зустрічаються і програмно-керовані периферійні пристрої.

Програма, яка виконується процесором, може обмінюватися даними за допомогою введення/виведення з довільними модулями, під'єднаними до внутрішньої шини комп'ютера, у тому числі і з контролерами ПП.

Периферійні пристрої можуть приймати від комп'ютера як дані, наприклад, байти інформації, які необхідно роздрукувати на папері, так і команди управління, у відповідь на які ПП може виконати спеціальні дії, скажімо, перевести головку диска на необхідну доріжку або виштовхнути аркуш паперу з принтера. Периферійний пристрій використовує зовнішній інтерфейс комп'ютера не тільки для приймання інформації, але і для пересилання інформації, тобто, обмін даними зовнішнім інтерфейсом, як правило, є двоспрямованим.

Контролери ПП приймають команди і дані від процесора у свій внутрішній буфер, який прийнято називати **регістром** або **портом**, пізніше виконують необхідні перетворення цих даних і команд у відповідності до форматів, які зрозумілі ПП, і передають їх на зовнішній інтерфейс.

Розподіл обов'язків між контролером і драйвером ПП може бути різним, але, як правило, контролер виконує набір простих команд з управління ПП, а драйвер використовує ці команди, щоб примусити пристрій здійснювати більш складні дії за деяким алгоритмом (рис. 6.2).

6.1.3. Найпростіший випадок взаємодії двох комп'ютерів

У найпростішому випадку взаємодія комп'ютерів може реалізовуватися за допомогою тих самих засобів, які використовувалися для взаємодії комп'ютера і ПП, наприклад через послідовний інтерфейс RS-232C. На відміну від взаємодії комп'ютера і ПП, коли програма працює тільки з однієї сторони – зі сторони комп'ютера, у цьому випадку відбувається взаємодія двох програм, які працюють на кожному з комп'ютерів.

Програма, яка працює на одному комп'ютері, не може отримувати безпосередній доступ до ресурсів другого комп'ютера – його дисків, файлів, принтера. Вона може тільки "попросити" про це програму, яка працює на комп'ютері, якому належить цей ресурс. Такі "прохання" реалізуються у вигляді **повідомлень**, які передаються каналами зв'язку між комп'ютерами. Повідомлення можуть містити не тільки команди на виконання певних дій, але й інформаційні дані.

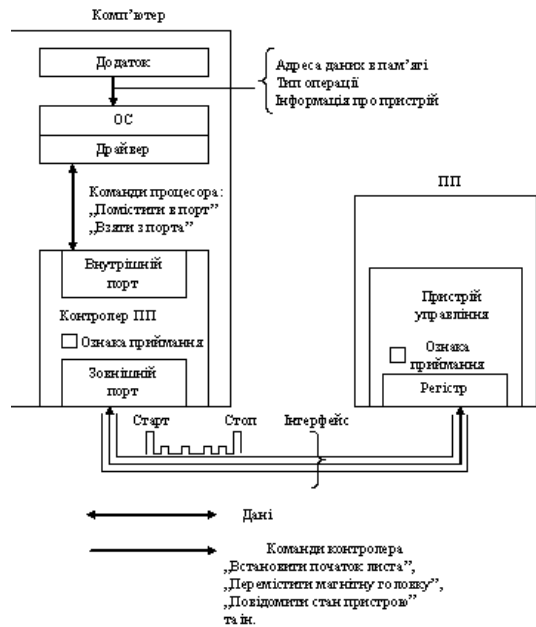


Рис. 6.2. Структурна схема та послідовність зв'язку комп'ютера з периферійним пристроєм.

Розглянемо випадок, коли користувачеві, який працює з текстово-графічним процесором на комп'ютері А, необхідно прочитати частину файла на диску комп'ютера В (рис. 6.3). Припустимо, що ми з'єднали ці комп'ютери кабелем зв'язку через СОМ-порти, які реалізують інтерфейс RS-232С. Для визначеності комп'ютери працюють під управлінням операційної системи MS-DOS, хоча принципного значення, у даному випадку, це не має.

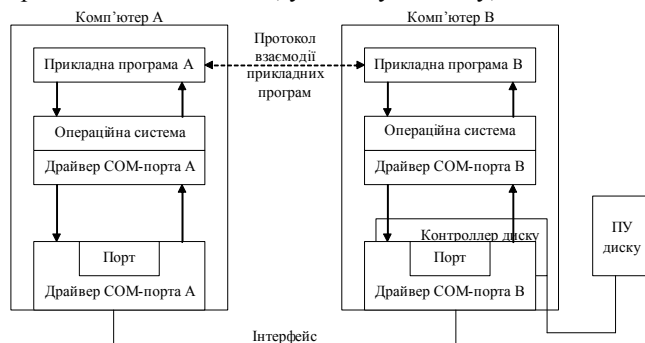


Рис. 6.3. Структурна схема процесу взаємодії двох комп'ютерів.

хостів цієї мережі. Наприклад, адреса мережі 193.194.5.0, маска цієї підмережі 255.255.255.0, ширококомвна адреса усіх хостів мережі 193.194.5.255, адреса маршрутизатора може бути 193.194.5.1.

6.5.4. Встановлення і налаштування стеку протоколів TCP/IP на хості

Встановлення та налаштування стеку протоколів TCP/IP на хості може здійснюватись у двох режимах – ручному (звичайний режим, який, проте, вимагає знання усіх параметрів TCP/IP для даного хосту, – їх можна довідатися у системного адміністратора КМ) та автоматичному (з використанням протоколу динамічного конфігурування хостів DHCP (Dynamic Host Configuration Protocol)). Для встановлення та налаштування стеку TCP/IP на хості у ручному режимі необхідно задати наступні параметри:

- IP-адресу хосту (яка повинна бути унікальною);
- маску підмережі;
- IP-адресу шлюзу за замовчуванням.

Якщо останній параметр не задати, тоді зв'язок буде обмежений локальною мережею.

Кожна плата мережевого адаптера повинна мати свої індивідуальні налаштування TCP/IP, причому одному адаптеру може відповідати декілька IP-адрес.

Встановлення TCP/IP у ручному режимі для хостів з операційними системами від фірми Microsoft здійснюється через вкладку Панель управління (Control Panel) ⇒ Мережа (Network). Для операційної системи Windows NT необхідно мати права адміністратора.

Встановлення TCP/IP з використанням протоколу DHCP, який автоматизує процес призначення IP-адрес, суттєво спрощує адміністрування КМ, особливо великих, а також у випадках, коли хост перемикається з однієї мережі на іншу. Протокол DHCP працює за клієнт-сервальною моделлю, де клієнт – хост, під час старту операційної системи здійснює запит до сервера на отримання IP-адреси. За динамічного розподілу IP-адрес сервер висилає адресу клієнту на обмежений час – час оренди (lease duration). Це дає можливість потім повторно використовувати цю IP-адресу для призначення її іншому хосту. Можлива навіть ситуація, коли хостів є більше, ніж зареєстрованих IP-адрес (при цьому максимальна кількість включених хостів буде дорівнювати кількості зареєстрованих IP-адрес).

DHCP-сервер окрім надання хосту відзначених вище параметрів TCP/IP може надавати додаткову інформацію про IP-адреси DNS-, WINS-серверів, тип хосту у мережі Microsoft – наприклад, гібридне або ширококомвне, доменне ім'я хосту тощо.

Для здійснення поновлення оренди параметрів TCP/IP на конкретному хості можна скористатись утилітами: для хосту під управлінням Windows NT – ipconfig.exe, для хосту під управлінням Windows 9x – winipcfg.exe.

У масці підмережі біти, які відповідають ідентифікатору мережі, встановлюються 1. Таким чином, значення кожного октету буде дорівнювати 255. Усі біти, які відповідають ідентифікатору хоста, встановлюються 0.

Наприклад, для хосту 129.84.3.24 значення маски за замовчуванням дорівнює 255.255.0.0 – це означає, що ідентифікатор мережі дорівнює 129.84.0.0, а ідентифікатор хосту в цій мережі дорівнює 0.0.3.24.

6.5.3. Визначення адреси призначення пакету. Шлюз за замовчуванням

Протокол IP використовує операцію логічного "І" для визначення того, якому хосту призначений пакет або який хост розташований у локальній мережі, або у віддаленій. При ініціалізуванні підтримки TCP/IP IP-адреса хосту порівнюється з його маскою підмережі за допомогою операції логічного "І". Перед висиланням кожного IP-пакету IP-адреса призначення порівнюється з тією ж маскою підмережі. Якщо результати двох перелічених вище операцій співпадають, тоді це означає, що отримувач пакету знаходиться у локальній мережі. В іншому випадку (якщо відправнику повідомлення не відомий маршрут до отримувача повідомлення) пакет відправляється на шлюз за замовчуванням (default gateway), який є шлюзом у іншій мережі. Шлюзом за замовчуванням є маршрутизатор.

Для виконання операції логічного "І" TCP/IP порівнює попарно відповідні біти IP-адреси і маски. Якщо обидва біти дорівнюють 1, тоді результат також дорівнює 1, а в інших випадках результуючий біт дорівнює 0.

Подамо приклад виконання набором TCP/IP логічної операції "І" для згаданого вище випадку адреси 129.84.3.24:

```
IP-адреса  10000001 01010100 00000011 00011000
маска підмережі 11111111 11111111 00000000 00000000
результат 10000001 01010100 00000000 00000000
```

Якщо у Вас найпростіша маленька мережа, яка складається усього лише з однієї підмережі, тоді усі хости у ній можуть пересилати дані безпосередньо один одному – немає необхідності у маршрутизації (насправді у такій мережі TCP/IP протокол не потрібен, і можна цілком обійтися, наприклад, протоколом NetBEUI – немаршрутизованим протоколом для хостів з операційними системами від Microsoft, базованими на ширококомовних повідомленнях). З іншої сторони, Ваша мережа може бути настільки велика, що застосування ширококомовного адресування зробить її неприцездатною. Виникне необхідність розбити її на менші підмережі.

Кожна підмережа має, як мінімум, один маршрутизатор. Якщо маршрутизатора немає, тоді спілкування з іншими мережами є неможливим і ніякої об'єднаної мережі не буде. Прийнято, що перша IP-адреса після номера мережі – це IP-адреса шлюзу за замовчуванням, тобто маршрутизатора і це є просто настанова, а не правило, як адреса мережі або ширококомовна адреса

Драйвер COM-порта разом з контролером COM-порта працюють так само, як і у випадку взаємодії ПП з комп'ютером. Однак, роль пристрою управління ПП виконує контролер і драйвер COM-порта іншого комп'ютера. Разом вони забезпечують пересилання кабелем між комп'ютерами одного байта інформації.

Драйвер комп'ютера В періодично опитує ознаку завершення приймання, встановлену контролером за правильно виконаного пересилання даних і поява цієї ознаки надає можливість зчитування прийнятого байта з буфера контролера в оперативну пам'ять, роблячи його доступним для програм комп'ютера В. У деяких випадках драйвер викликається асинхронно за перериваннями контролера.

Таким чином, у розпорядженні комп'ютерів А і В наявний засіб для пересилання одного байта інформації. Але розглянута задача є набагато складнішою, так як необхідно передати не один байт інформації, а певну частину заданого файлу. Усі пов'язані з цим додаткові проблеми повинні розв'язувати програми більш високого рівня, ніж драйвери COM-портів. Назвемо такі програми додатком А і додатком В. Отже, додаток А повинен сформувати повідомлення-запит для додатку В. У повідомленні необхідно вказати ім'я файлу, тип операції (у даному випадку – зчитування), зміщення і розмір області файлу, який містить необхідні дані.

Щоб передати це повідомлення комп'ютеру В, додаток А звертається до драйвера COM-порту, повідомляючи йому адресу в оперативній пам'яті, за якою драйвер знаходить повідомлення, а потім передає його байт за байтом додаткові В. Додаток В, отримавши запит, виконує його, тобто зчитує необхідну область файлу з дискового простору, використовуючи засоби локальної OS у буферну область своєї оперативної пам'яті, а далі за допомогою драйвера COM-порту передає зчитані дані каналом зв'язку у комп'ютер А, де вони і потрапляють у додаток А.

Описані функції додатку А могла би виконати сама програма текстово-графічного процесора, але вмонтовувати ці функції до складу кожного додатку – текстово-графічних процесорів, графічних редакторів, систем управління базами даних, котрим потрібен доступ до файлів – є нерациональним. Вигідніше створити спеціальний програмний модуль, який буде виконувати функції формування повідомлень-запитів і приймання результатів для усіх додатків комп'ютера. Такий службовий модуль називається **клієнтом**. На стороні комп'ютера В повинен працювати інший модуль – **сервер**, який постійно очікує надходження запиту на віддалений доступ до файлів, розміщених на диску цього комп'ютера. Сервер, отримавши запит з мережі, звертається до локального файлу і виконує з ним задані дії за участі локальної OS.

Програмні клієнт і сервер виконують системні функції з обслуговування запитів додатків комп'ютера А на віддалений доступ до файлів комп'ютера В.

Схема взаємодії клієнта і сервера з додатками і операційною системою подана на рис. 6.4.

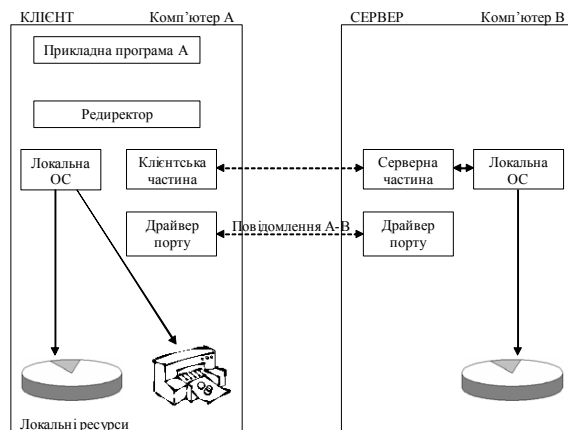


Рис. 6.4. Схема взаємодії програмних компонент при з'єднанні двох комп'ютерів.

Зручною і корисною функцією клієнтської програми є спроможність розпізнавати запит до віддаленого файлу від запиту до локального файлу. Якщо клієнтська програма уміє це робити, тоді додатки не повинні "турбуватися" про те, з яким файлом вони працюють (локальним або віддаленим), клієнтська програма сама розпізнає і переспрямовує (redirect) запит до віддаленого комп'ютера. Звідси і походить назва для клієнтської частини мережевої ОС – **редиректор**. Інколи функції розпізнавання виокремлюються в окремий програмний модуль. У цьому випадку редиректором називають не всю клієнтську частину, а тільки цей програмний модуль.

6.2. Еталонна модель взаємодії відкритих систем

Прагнення до максимального упорядкування і спрощення процесів розроблення, модернізування і розширення КМ визначило необхідність прийняття стандартів, які регламентують принципи і процедури організування взаємодії абонентів комп'ютерних мереж. Інтенсивні роботи у цьому напрямку ведуться міжнародними організаціями, такими як Міжнародна організація стандартизування (ISO), Європейська асоціація виробників комп'ютерів (European Computer Manufactur Association – ECMA) та ін.

Міжнародною організацією стандартизування був спеціально створений Технічний комітет ТС 97, один із підкомітетів якого (підкомітет SC 16), займається розробленням стандартів для комп'ютерних мереж. Першим завданням, розв'язаним у рамках стандартизування КМ, було визначення структури стандартів і принципів організації робіт з їх створення. Основним результатом

Класи D та E мають специфічне призначення. Адреси класу D призначені для групових повідомлень. Чотири старші біти в цих адресах завжди дорівнюють двійковому значенню 1110. Решта бітів означають конкретну групу отримувачів і не діляться на частини. Пакети з такими адресами розсилаються вибраній групі хостів у мережі.

Таблиця 6.2

Показники мереж класів А, В та С

Клас	Кількість мереж	Кількість хостів у мережі	Діапазон значень ідентифікаторів мережі
Клас А	126	16777214	1 – 126
Клас В	16384	65534	128 – 191
Клас С	2097152	254	192 - 223

Клас Е – експериментальний, зарезервований для майбутнього використання і, наразі, не використовується. Чотири старші біти у цих адресах завжди дорівнюють двійковому значенню 1111.

При визначенні максимальної кількості хостів m у мережі використовується формула $m = 2^n - 2$, де: n – кількість бінарних розрядів, відведених під ідентифікатор хосту, а віднімання числа 2 від загальної кількості пояснюється наявністю у кожній мережі адреси самої мережі та адреси усіх хостів у цій мережі.

6.5.2. Маска під мережі

Маска підмережі є 32-розрядним бінарним числом, яке використовується для виокремлення (маскування) з IP-адреси її частин: ідентифікаторів мережі та хосту. Така процедура необхідна для того, щоб з'ясувати, відноситься та, або інша IP-адреса до локальної, чи до віддаленої мережі.

Кожен хост TCP/IP повинен мати маску підмережі або таку, яка задається за замовчуванням (у тому випадку, коли мережа не ділиться на підмережі), або спеціальну (якщо мережа розбита на декілька підмереж). Значення маски підмережі за замовчуванням залежить від використовуваного у даній мережі класу IP-адрес (табл. 6.3).

Таблиця 6.3

Значення масок за замовчуванням для IP-адрес класів А, В, С

Клас адрес	Біти, які використовуються для маски підмережі	Десятковий запис з точковими розділювачами
Клас А	11111111 00000000 00000000 00000000	255.0.0.0
Клас В	11111111 11111111 00000000 00000000	255.255.0.0
Клас С	11111111 11111111 11111111 00000000	255.255.255.0

розділювачами (dotted decimal notation). Остання форма використовується як зручніша для сприйняття в порівнянні з бінарною формою. Кожна IP-адреса має довжину 32 біти і для зручності її поділяють на чотири октети, які відділяються один від одного крапками. Кожен октет подається десятковим числом у діапазоні від 0 до 255. Ці 32 розряди IP-адреси містять ідентифікатор мережі (network ID) та ідентифікатор хосту (host ID)

Ідентифікатор мережі визначає фізичну мережу. Він є однаковим для усіх вузлів у одній КМ і унікальний для кожної з мереж, включених у об'єднану мережу. Ідентифікатором вузла є адреса конкретного вузла у цій мережі.

Кожен хост повинен отримати унікальну адресу в Internet у вигляді IP-адреси. Для кінцевого користувача або організації IP-адресу, або, відповідно, діапазон IP-адрес може надавати регіональний Internet сервіс-провайдер (ISP) (фірма, яка надає послуги Internet і є точкою входу в Internet (POP)). Організації відзначається блок IP-адрес, який відповідає розміру її КМ.

Варто відзначити, що хоча інформація, отримана з Internet, є безкоштовною, проте послуги за користування Internet є платними. Причиною цього може служити хоча би той факт, що ISP є суб'єктом підприємницької діяльності. Тому, природно, що за кожен зареєстрований IP-адресу слід платити і ця оплата буде пропорційною до кількості зареєстрованих IP-адрес.

ISP, у свою чергу, повинен отримати діапазон адрес для надання їх клієнтам у організації IANA (Internet Assigned Numbers Authority), яка координує розподіл IP-адрес у мережі Internet, або ж у Мережевого інформаційного центру Internet (InterNIC).

У протоколі IPv4 існує п'ять класів IP-адрес у відповідності з різними розмірами комп'ютерних мереж. Клас адреси визначає, які біти відносяться до ідентифікатора мережі, а які – до ідентифікатора вузла. Також, клас визначає максимально можливу кількість вузлів у мережі. Класи IP-адрес ідентифікують за значенням першого октету адреси.

Адреси класу А призначаються хостам дуже великих мереж. Старший біт у цих адресах завжди дорівнює нулю. Перший октет присвоюється організацією InterNIC і не підлягає модифікуванню. Решта три октети містять ідентифікатор вузла.

Адреси класу В призначаються хостам великих та середніх за розмірами мереж. Два старші біти в цих адресах завжди дорівнюють двійковому значенню 10. Два перші октети присвоюються організацією InterNIC і не підлягають модифікуванню. Решта два октети містять ідентифікатор вузла.

Адреси класу С застосовуються у невеликих мережах. Три старші біти в цих адресах завжди дорівнюють двійковому значенню 110. Три перші октети присвоюються організацією InterNIC і не підлягають модифікуванню, а останній четвертий октет є ідентифікатором вузла.

Кількість мереж, кількість хостів у цих мережах та діапазон значень ідентифікаторів мереж для класів А, В та С подані у табл. 6.2.

роботи у цьому напрямку стало створення стандарту 7498, який визначає базову еталонну модель взаємодії відкритих систем, так звану "Еталонну модель OSI". OSI – скорочення від Open System Interconnection (взаємодія відкритих систем). Цей стандарт був прийнятий за основу всіма організаціями, які займаються розробленням стандартів у галузі КМ.

Цей стандарт визначає:

- поняття й основні терміни, використовувані для побудови відкритих систем;
- можливості і конкретні послуги, які повинна надавати відкрита система;
- логічну структуру відкритих систем, протоколи, які забезпечують функціонування відкритих систем.

Розглядаючи взаємодію структурних елементів КМ, вводиться поняття **система**, під яким мається на увазі сервер, абонентська або довільна інша система, яка надає або споживає мережеві ресурси. Згідно зі стандартом, відкритою вважається система, яка відповідає вимогам еталонної моделі OSI, реалізує стандартний набір послуг та підтримується стандартними протоколами. Дотримання цих вимог забезпечує можливість взаємодії відкритих систем між собою, незважаючи на їх технічні й логічні розбіжності у реалізуванні, що є істотним у побудові КМ. У рамках мережевих технологій "відкритість" систем використовується з метою забезпечення можливості під'єднання до КМ обладнання різних фірм без додаткового дороблення мережевого програмного і апаратного забезпечення. При цьому основною і, мабуть єдиною, умовою є те, щоб засоби під'єднання також відповідали вимогам моделі взаємодії відкритих систем.

Відкриті системи об'єднуються за допомогою мережі пересилання даних у відкриту КМ. Варто зауважити, що модель OSI не розглядає структуру і характеристики фізичних засобів з'єднання, а тільки визначає основні вимоги до них. Основним завданням моделі OSI є опис численних функцій, які визначають правила взаємодії відкритих систем. При цьому широко використовується поняття **процес**, визначений як динамічний об'єкт, який реалізує цілеспрямований акт оброблення інформації. Таке формалізування дозволяє відзначити характерні риси процесу взаємодії систем, незалежно від засобів його реалізування. Справа у тому, що за інтенсивного режиму роботи, який є характерним для сучасних комп'ютерів, виконання тієї самої програми у різні моменти часу може здійснюватися по-різному. Це залежить від деяких чинників і, у першу чергу, – від числа задач у системі, порядку їх виконання і ресурсів системи, які їм надаються. Таким чином, програма не може однозначно визначати функціонування систем і порядок їх взаємодії, з цією метою і вводиться поняття процесу.

Прийнято поділяти процеси на прикладні і системні. Прикладний процес ототожнюється з реалізуванням певних процедур, пов'язаних з опрацюванням інформації при розв'язанні завдань користувачами. Системні ж процеси визначають виконання допоміжних функцій, пов'язаних із забезпеченням

прикладних процесів. До системних процесів відносяться: організація зв'язку між прикладними процесами; керування каналами пересилання даних; активування терміналів тощо. Процес, як довільний динамічний об'єкт, триває у часі і складається з етапів ініціалізування, виконання і завершення. При цьому процес може породжуватись користувачем, системою або іншим процесом. Введення даних, необхідних для виконання процесу і виведення даних здійснюється у формі повідомлень через логічні (програмно-організовані) точки, названі **портами**. Розрізняють вхідні і вихідні порти. Через вхідні порти вводяться дані для конкретного процесу, відповідно, через вихідні порти поточний процес видає результати опрацювання даних. Взаємодія процесів (рис. 6.5) здійснюється шляхом обміну повідомленнями, які є блоками даних певної структури. Проміжок часу, протягом якого взаємодіють процеси, прийнято називати **сеансом обміну** або **сесією**. Протягом сеансу обміну процес формує повідомлення і необхідну для його пересилання супровідну інформацію. Залежно від завдання, яке розв'язується, взаємодіючі процеси можуть генеруватися в одній або суміжних системах.



Рис.6.5. Схема взаємодії процесів.

Розглянемо модель взаємодії відкритих систем. В основу цієї моделі покладено концепцію багаторівневої організації протоколів. Істотною особливістю моделі взаємодії відкритих систем є розроблення і використання єдиного підходу до організації протоколів і інтерфейсів різних рівнів. Згідно з цією концепцією кожному рівню ставиться у відповідність набір певних функцій, пов'язаних з розв'язанням конкретної задачі з організації взаємодії відкритих систем. Нумерація рівнів здійснюється відносно до фізичних засобів з'єднання, тобто перший номер присвоюється фізичному рівню, а найбільший номер – прикладному (користувацькому) рівню. Кожний рівень з меншим номером вважається допоміжним для суміжного з ним вищого рівня і надає йому певний набір послуг, названий **сервісом**. Слід зауважити, що еталонна модель OSI не визначає засобів реалізування протоколів, а тільки специфікує їх. Таким чином, функції кожного рівня можуть бути реалізовані різними апаратними і програмними засобами. Основною умовою при цьому є те, що взаємодія між довільними суміжними рівнями повинна чітко визначатися, тобто здійснюватись через точки доступу за допомогою стандартного міжрівневого інтерфейсу (рис. 6.6).

Точка доступу є портом, у якому об'єкт N-го рівня надає послуги (N+1)-му рівню. Ця достатньо важлива умова визначає можливість зміни протоколів окремих рівнів без зміни системи у цілому, що у свою чергу є однією з основ-

окрім цього, виконує функції ресинхронізування та підсилення сигналів на відміну від пасивного концентратора.

У випадку використання пасивного MSAU роль підсилення сигналів бере на себе кожний мережевий адаптер, а роль ресинхронізуючого блоку виконує мережевий адаптер активного монітора кільця.

Кінцеві станції під'єднуються до MSAU за топологією "зірка", а самі MSAU об'єднуються через спеціальні порти Ring In та Ring Out для створення магістрального фізичного кільця. Кабелі, які з'єднують станцію з концентратором, називають **відгалуженими** (lobe cable), а кабелі, які з'єднують концентратори – **магістральними** (trunk cable).

Технологія Token Ring дозволяє використовувати для з'єднання кінцевих станцій і концентраторів такі види кабелів як: STP Type 1; STP Type 3; STP Type 6, а також волоконно-оптичний кабель.

6.5. Протоколи TCP/IP та IP-адресування

Набір протоколів TCP/IP (Transmission Control Protocol/Internet Protocol) – це стандартний промисловий набір протоколів, розроблений для глобальних мереж (WAN), який був створений у результаті досліджень мереж з комутацією пакетів агенством DARPA. TCP/IP є найбільш уживаним сімейством мережевих протоколів, оскільки:

- є єдиним незалежним від операційних платформ набором протоколів;
- це єдиний набір протоколів з відкритим процесом визначення стандартів та відсутністю власника.

У 4-рівневій моделі TCP/IP протокол IP належить до мережевого рівня, функцією якого є забезпечення пересилання інформації в системі, яка об'єднує довільну кількість мереж, оскільки ці мережі можуть використовувати різноманітні принципи пересилання повідомлень між кінцевими вузлами і володіти довільною структурою зв'язків. Загалом, протокол IP не орієнтований на з'єднання, не гарантує доставлення повідомлення і тому вважається ненадійним протоколом. Він призначений для маршрутизування та висилання пакетів між мережами та вузлами.

У термінології Internet комп'ютер, на якому працює мережевий протокол, наприклад, протокол IP з набору TCP/IP, називається **хостом** (host). Хости обмінюються даними між собою і значна частка діяльності в Internet обумовлена управлінням інформаційними потоками між комп'ютерами-хостами. Терміном **вузол** (node), як правило, коротко називають такі пристрої як міст, маршрутизатор, комутатор, шлюз або хост.

Існує дві версії протоколу IP: IPv4 та IPv6. На даний момент використовується протокол IPv4, який описаний у RFC 791.

6.5.1 Протокол IPv4, формат IP-адреси. Класи IP-адрес. Зарезервовані IP-адреси

Розглянемо IP-адресування протоколу IPv4. IP-адреса може бути записана у двох форматах – двійковому (binary) та десятковому з точковими

станція зобов'язана припинити пересилання власних даних (поточний кадр дозволяється завершити) і передати маркер далі кільцем. Час утримання маркера за замовчуванням становить 10 мс, протягом яких станція може встигнути передати один або декілька кадрів у залежності від їх розміру.

Для контролю мережі одна із станцій виконує роль **активного монітора**. Активний монітор вибирається під час ініціалізування кільця як станція з максимальним значенням MAC-адреси (номер мережевого адаптера станції). При виході активного монітора з ладу, процедура ініціалізування кільця повторюється і вибирається новий активний монітор. Активний монітор відповідає за наявність у мережі єдиної копії маркера. Якщо активний монітор не одержує маркер протягом певного проміжку часу (наприклад, 2.6 с), тоді він створює новий маркер.

Кожній робочій станції у кільці присвоєно певний пріоритет у керуванні маркером. Перші три біти маркера є бітами пріоритету. Станція, одержавши маркер, порівнює його пріоритет зі своїм і, у випадку збігу, має право на пересилання. Якщо ж станція має нижчий пріоритет, ніж маркер, тоді вона просто ретранслює його. Останні три біти маркера станція використовує для оголошення про свій пріоритет. Одержавши кадр, станція порівнює пріоритет, записаний у полі оголошення пріоритету, зі своїм власним. Якщо вона має інформацію для пересилання і її пріоритет вищий від оголошеного, тоді станція пропоставляє у полі оголошення свій пріоритет. Отже, коли кадр повністю обійде кільце у ньому буде записаний максимальний пріоритет станції, яка потребує пересилання.

Стандарт Token Ring передбачає формування зв'язків у мережі за допомогою концентраторів, які називаються MAU (Multistation Access Unit) або MSAU (Multi-Station Access Unit), тобто пристроями багатостанційного доступу (рис. 6.22). У даній мережі може бути під'єднано до 260 станцій.

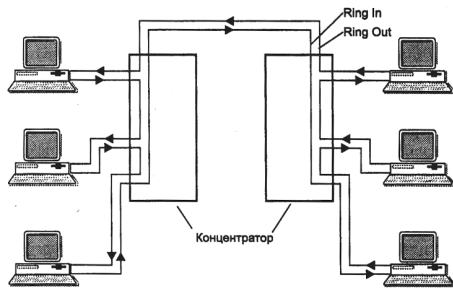


Рис. 6.22. Логічна топологія технології Token Ring.

Концентратор Token Ring може бути пасивним і активним. Пасивний концентратор просто з'єднує порти внутрішніми зв'язками так, щоб станції, які під'єднуються до цих портів утворювали кільце. MSAU забезпечує обхід порту, до якого приєднаний вимкнений комп'ютер. Активний концентратор,

них умов побудови відкритих систем. Зауважимо, що у випадку програмного реалізування міжрівневого інтерфейсу, портами є адреси, за якими записуються міжрівневі повідомлення.

У свою чергу, взаємодія об'єктів (як правило, програм) однойменних рівнів різних систем визначається за допомогою протоколів відповідного рівня, проте, і у цьому випадку, обмін даними здійснюється через міжрівневі інтерфейси всередині кожної з систем, а між ними – каналами пересилання даних.

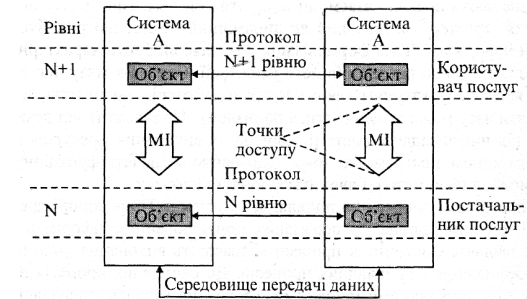


Рис. 6.6. Схема взаємодії об'єктів та послуг.

Структурною одиницею інформації, яка передається між рівнями, є так званий **протокольний блок даних** (рис. 6.7), який складається з керуючого поля, названого заголовком, і поля даних. Заголовок N-го блоку містить керуючу інформацію, формовану (рис. 6.8) на N-ому рівні. Вміст поля даних N-го рівня є блоком даних (N+1)-го рівня. Таким чином формується вкладена структура, протокольних блоків даних, які починаючи з верхнього рівня, вкладаються один у одного. При передаванні інформації у зворотному напрямку відбувається зворотна процедура "розпакування" блоків.

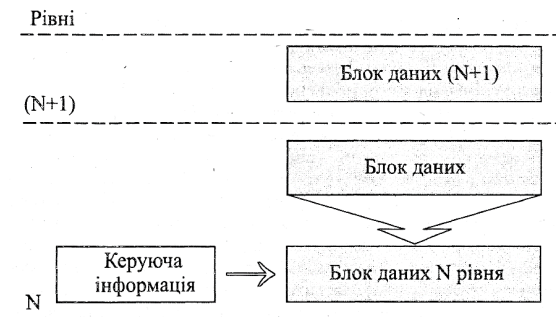


Рис. 6.7. Структура протокольного блоку даних.



Рис. 6.8. Схема формування протокольного блоку даних.

У процесі побудови довільної багаторівневої структури виникає необхідність визначення оптимального числа її рівнів. Так, при розробленні еталонної моделі OSI число її рівнів визначалося з урахуванням таких міркувань:

- розбивання на рівні повинно максимально відтворювати логічну структуру КМ;
- міжрівневі межі повинні визначатися таким чином, щоб забезпечити мінімальне число і простоту міжрівневих зв'язків;
- вважається, що велика кількість рівнів, з одного боку спрощує внесення змін у систему, а з іншого – збільшує кількість міжрівневих протоколів і ускладнює опис моделі у цілому.

З урахуванням цього, міжнародною організацією стандартів для КМ була запропонована семирівнева (рис. 6.9) модель OSI.

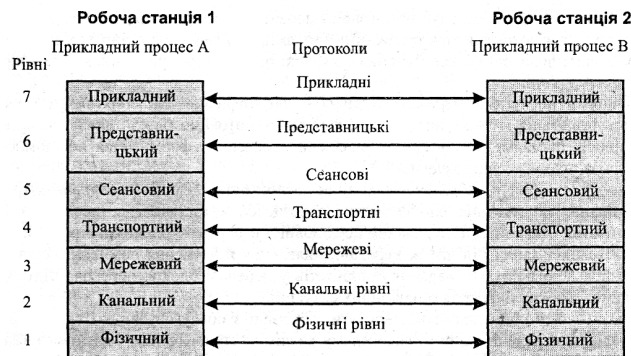


Рис. 6.9. Еталонна модель OSI.

Основним, з точки зору користувача, є **прикладний рівень**. Цей рівень забезпечує виконання прикладних процесів користувачів і визначає семантику, тобто змістовну складову інформації, якою обмінюються відкриті системи у процесі взаємодії. З цією метою прикладний рівень, крім протоколів взаємодії прикладних процесів, містить протоколи пересилання файлів, віртуального терміналу, електронної пошти тощо.

Шостий рівень називається **представницьким** (рівень подання даних);

женими (правило 5-4-3), тобто такими, до яких під'єднуються кінцеві мережеві пристрої.

У стандарті 10Base-T дозволяється використовувати не більше 4 повторювачів. Обмежене число повторювачів пояснюється додатковими затримками розповсюдження сигналу, яке вони вносять. Застосування повторювачів збільшує час подвійного розповсюдження сигналу, який для надійного розпізнавання колізій не повинен перевищувати час пересилання кадру мінімальної довжини (576 біт).

6.4.2. Технологія Token Ring

Мережеві технології Token Ring притаманний детермінований алгоритм доступу до розподіленого середовища пересилання даних, який базується на передаванні станціям права на використання середовища у певному порядку. Це право передається за допомогою кадру спеціального формату, який називається **маркером** або **токеном**. Логічна топологія даної технології є кільцем, у якому довільна станція завжди безпосередньо одержує дані від тієї станції, яка є попередньою у кільці (рис. 6.21).

Одержавши маркер, станція аналізує його і, за відсутності у неї даних на пересилання, забезпечує його пересилання до наступної станції. Станція, яка має дані для пересилання, при одержанні маркера знімає його з кільця, що дає їй право доступу до фізичного середовища і пересилання своїх даних. Після цього станція висилає у кільце кадр даних встановленого формату послідовно бітами. Дані проходять кільцем до станції призначення завжди в одному напрямку. Кадр містить у собі окрім поля даних поле адреси отримувача і поле адреси відправника. Усі станції кільця ретранслюють кадр побітно, як повторювачі. Якщо кадр проходить через станцію призначення, тоді, розпізнавши свою адресу, ця станція копіює кадр у свій внутрішній буфер і встановлює у кадр ознаку підтвердження приймання. Станція, яка вислала кадр даних у кільце, при його повторному одержанні з підтвердженням приймання, знімає цей кадр з кільця і пересилає у мережу новий маркер для забезпечення можливості іншим станціям мережі пересилати дані.

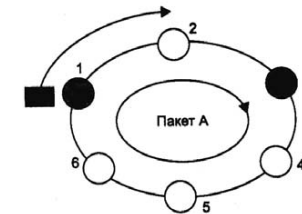


Рис 6.21. Логічна топологія технології Token Ring.

Час володіння станцією розподіленим середовищем у мережі Token Ring обмежується **часом утримання маркера** (token holding time), із плином якого

Важливою частиною технології Ethernet є процедура визначення доступу до середовища пересилання. Щоб отримати можливість передати кадр, станція повинна переконавшись, що розподілене середовище вільне. Це досягається прослуховуванням основної гармоніки сигналу. Ознака незайнятості середовища – відсутність на ній несучої частоти. Якщо середовище вільне, тоді станція має право розпочати пересилання кадру. При попаданні кадру у розподілене середовище пересилання даних усі мережеві адаптери одночасно починають приймати цей кадр. Всі вони аналізують адресу отримувача, і, якщо ця адреса співпадає з їх власною адресою, тоді кадр поміщається у внутрішній буфер відповідного мережевого адаптера.

Коли одночасно дві або більше робочих станцій вирішують, що мережа вільна і починають пересилати дані, тоді зміст цих кадрів "зіштовхується" у спільному кабелі і відбувається спотворення інформації. Така ситуація називається **колізійою**. У стандарті Ethernet передбачений алгоритм виявлення і опрацювання колізії – станції, які пробували передати свої кадри, припиняють пересилання і після паузи випадкової тривалості пробують знову отримати доступ до середовища і передати той кадр, який викликав колізійою. Ймовірність виникнення колізії залежить від інтенсивності мережевого трафіку. Метод доступу CSMA/CD залишається тим самим для довільної специфікації фізичного середовища технології Ethernet. У таблиці 6.1 подані специфікації фізичного середовища Ethernet.

Таблиця 6.1

Специфікації фізичного середовища Ethernet

Кабель	10Base-5	10Base-2	10Base-T	10Base-F
	"Товстий" коаксіальний кабель RG-8 або RG-11	"Тонкий" коаксіальний кабель RG-58	Неекранована скручена пара категорії 3, 4, 5	Волоконнооптичний кабель
Максимальна довжина сегмента, м	500	185	100	2000
Максимальна віддаль між комп'ютерами мережі (при використанні повторювачів), м	2500	925	500	2500 (2740 для 10Base-FB)
Максимальне число станцій в сегменті	100	30	1024	1024
Максимальне число повторювачів між довільними станціями мережі	4	4	4	4 (5 для 10Base-FB)

У стандартах 10Base-5 і 10Base-2 дозволяється використовувати у мережі не більше 5 сегментів кабелю. Тільки 3 сегменти із 5 можуть бути наванта-

він визначає єдиний для всіх відкритих систем синтаксис інформації, яка передається. Необхідність у цьому рівні зумовлена різною формою подання інформації у мережі пересилання даних та комп'ютерах. Цей рівень відіграє важливу роль у забезпеченні відкритості систем, дозволяючи їм спілкуватися між собою незалежно від їхньої внутрішньої мови.

П'ятий рівень називають **сеансовим**, тому що основним його призначенням є організування сеансів зв'язку між прикладними процесами у різних абонентських системах. На цьому рівні створюються порти для приймання і пересилання повідомлень і організуються з'єднання – логічні канали між процесами. Необхідність протоколів цього рівня зумовлюється відносною складністю мережі пересилання даних і прагненням забезпечити високу надійність пересилання інформації.

Четвертий, **транспортний** рівень (рівень наскрізного пересилання), забезпечує пересилання даних між двома взаємодіючими відкритими системами та з'єднання абонентів мережі з системою пересилання даних. На цьому рівні однозначно визначається взаємодія абонентських систем – джерела і адреси даних, організовується і підтримується логічний канал (транспортне з'єднання) між абонентами.

Третій, **мережевий** рівень, забезпечує маршрутизування інформації і керування мережею пересилання даних. На відміну від попередніх, цей рівень більшою мірою орієнтований на мережу пересилання даних. Тут розв'язуються задачі, пов'язані з управлінням мережею пересилання даних, у тому числі маршрутизування і керування інформаційними потоками.

Канальний рівень забезпечує функціональні і процедурні засоби для встановлення, підтримання і розривання з'єднань на рівні каналів пересилання даних. Процедури канального рівня забезпечують виявлення і, можливо, усунення помилок, які виникають на фізичному рівні.

Фізичний рівень забезпечує механічні, електричні, функціональні і процедурні засоби організування фізичних з'єднань при передаванні біта даних між фізичними об'єктами.

Чотири нижні рівні утворюють транспортну службу КМ, яка забезпечує пересилання ("транспортування") інформації між абонентськими системами, звільняючи вищі рівні від розв'язання цих завдань.

У свою чергу, три верхні рівні, які забезпечують логічну взаємодію прикладних процесів, функціонально об'єднуються у абонентську службу.

Простого перерахування рівнів недостатньо для визначення правил взаємодії систем, тому в рамках еталонної моделі також визначаються **послуги**, які повинні забезпечувати її рівні. Послуги – за своєю суттю, функції, які виконуються на заданому рівні.

Зокрема, фізичний рівень повинен забезпечувати такі види послуг: установа й ідентифікування фізичних з'єднань; організація послідовного пересилання інформації; оповіщення про закінчення зв'язку.

Канальний рівень забезпечує організацію необхідної послідовності та пересилання блоків даних; керування потоками між суміжними вузлами; іденти-

фікування кінцевих пунктів каналних з'єднань; виявлення і усунення помилок; оповіщення про помилки, не усунені на каналному рівні.

Основними послугами мережевого рівня є: ідентифікування кінцевих точок мережевих з'єднань; організація мережевих з'єднань; керування потоками блоків даних; забезпечення послідовності доставляння блоків даних; виявлення помилок і формування повідомлень про них; розривання мережевих з'єднань.

Транспортний рівень забезпечує такі види послуг: встановлення і розривання транспортних з'єднань; формування блоків даних; забезпечення взаємодії сеансових з'єднань з транспортними з'єднаннями; керування послідовністю пересилання блоків даних; забезпечення цілісності блоків даних під час їх пересилання; виявлення й усунення помилок, повідомлення про неусунені помилки; надання пріоритетів у передаванні блоків; пересилання підтверджень про приймання блоків; ліквідування безвихідних ситуацій.

На сеансовому рівні надаються послуги, пов'язані з обслуговуванням сеансів обміну і забезпеченням пересилання даних у діалоговому режимі; встановленням сеансового з'єднання; обміном даними; керуванням обміном даними; синхронізуванням сеансового з'єднання; повідомленням про виняткові ситуації; відтворенням на транспортний рівень сеансового з'єднання; завершенням сеансового з'єднання.

Представницький рівень забезпечує такі види послуг: вибір форми подання даних; інтерпретування і перетворення даних до вигляду, зручного для прикладних процесів; перетворення синтаксису даних; формування даних.

Прикладний рівень забезпечує широкий набір послуг: керування терміналами; керування файлами; керування діалогом; керування задачами; керування мережею у цілому; забезпечує цілісність інформації та надає додаткові послуги з організування електронної пошти, пересилання масивів повідомлень тощо. Послуги різних рівнів визначаються за допомогою протоколів еталонної моделі OSI, тобто, правилами взаємодії об'єктів однойменних рівнів відкритих систем. Відповідно до семирівневої моделі взаємодії відкритих систем вводиться сім типів протоколів, які іменуються так само, як рівні. При цьому за функціональним призначенням усі протоколи доцільно поділити на три групи.

Першу групу складають **протоколи абонентської служби**, які відповідають прикладному, представницькому і сеансовому рівням моделі OSI. Протоколи цієї групи є незалежними від мережі, тобто їхні характеристики і структура не залежать від використовуваної мережі пересилання даних. Вони визначаються лише структурою абонентських систем і функціями щодо оброблення інформації. Дві інші групи протоколів описують **транспортну службу** КМ і різняться між собою процедурою доступу до передавального середовища. Одна з цих груп визначає систему пересилання даних з маршрутизування інформації, а інша – з селекцією інформації.

Маршрутизування – це процедура визначення шляху пересилання інформації у мережах пересилання даних і є характерною для глобальних і

Даний метод застосовується виключно у мережах з логічною спільною шиною. Всі комп'ютери такої мережі мають безпосередній доступ до спільної шини, тому вона може бути використана для пересилання даних між довільними двома комп'ютерами мережі.

Основний організаційний принцип, закладений в основу Ethernet, – **випадковий метод доступу** до розподіленого середовища пересилання даних. У якості такого середовища може використовуватись товстий (10Base-5) або тонкий (10Base-2) коаксіальний кабель, скручена пара (10Base-T), оптоволоконні кабелі (10Base-FL, 10Base-FB). У даній технології чітко зафіксована топологія логічного зв'язку ("спільна шина"), хоча WS можуть під'єднуватися до розподіленого середовища у відповідності з типовою структурою електричних зв'язків "спільна шина" (рис. 6.19) або "зірка" (рис. 6.20).

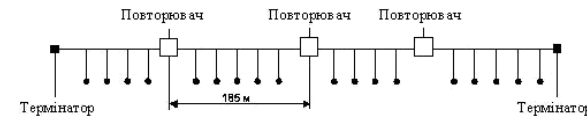


Рис. 6.19. Мережа Ethernet. Фізична топологія "спільна шина".

За допомогою розподілу у часі функціонування шини довільні два комп'ютери можуть обмінюватись даними. Час монопольного використання розподіленого середовища одним комп'ютером обмежується часом пересилання одного кадру. **Кадр** – одиниця даних, якими обмінюються комп'ютери у мережі Ethernet.

Кадр має фіксований формат і містить такі поля як: поле даних, поле адреси відправника, поле адреси отримувача. Управління доступом до лінії зв'язку здійснюється мережевими адаптерами Ethernet, кожен з яких має унікальну адресу. Пересилання даних здійснюється зі швидкістю 10 Мбіт/с. Ця величина і є перепускною здатністю мережі Ethernet.

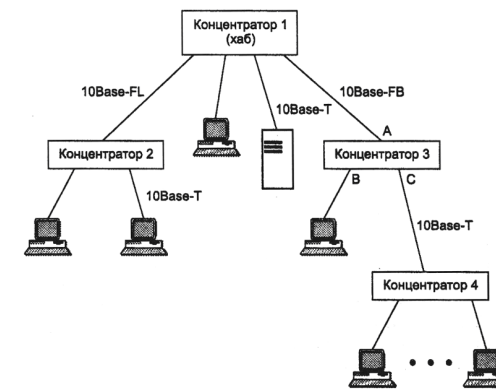


Рис. 6.20. Мережа Ethernet. Фізична топологія "зірка".

Топологія подвійного кільця. Мережами з таким конфігуруванням є мережі FDDI. Вони відрізняються вмонтованою надлишковістю, яка забезпечує захист від системних відмов: основне кільце слугує для пересилання даних, а допоміжне кільце – для пересилання керуючих сигналів (рис. 6.17).

Існує можливість пересилання даних обома кільцями у протилежних напрямках у випадку відсутності обривів кабелю. Якщо ж трапляється обрив кабелю або одна зі станцій виходить з ладу, основне кільце об'єднується з допоміжним, знову утворюючи єдине кільце. Цей режим роботи мережі називається завертанням кілець.

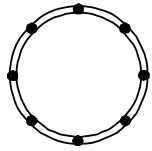


Рис. 6.17. Фізична топологія подвійного кільця.

Лінійна (ланцюгова) топологія. Це топологія, у якій кожна робоча станція з'єднана з попередньою та наступною відносно себе (рис. 6.18). Виникає з кільцевої топології при видаленні однієї гілки. Часом трактується як ідентична до шинної топології.



Рис. 6.18. Лінійна (ланцюгова) фізична топологія.

6.4. Базові технології локальних мереж

Мережева технологія – це узгоджений набір стандартних протоколів і програмно-апаратних засобів, які їх реалізують (мережевих адаптерів, драйверів, кабелів та роз'ємів), що є достатнім для побудови комп'ютерної мережі. Інколи мережеві технології називають **базовими технологіями**, маючи на увазі те, що на їх основі формується базис довільної мережі. Прикладами базових технологій можуть слугувати такі як: Ethernet, Token Ring, FDDI, 100VGAny-LAN. Для побудови довільної з цих мереж достатньо придбати програмні та апаратні засоби, які відносяться до цієї базової технології – мережеві адаптери з драйверами, концентратори, комутатори, кабельну систему тощо – і з'єднати їх у відповідності з вимогами стандарту на дану технологію.

6.4.1. Технологія Ethernet

У мережах Ethernet використовується метод доступу до середовища пересилання даних, відомий під назвою **множинний доступ** з прослуховуванням основної гармоніки сигналу і виявленням колізій (CSMA/CD – carrier-sense-multiply-access-with/collision-detection).

регіональних комп'ютерних мереж, у рамках яких і розглядається відповідна група протоколів.

Селекцією у комп'ютерних мережах називається процес вибору чергової абонентської системи для під'єднання її до мережі пересилання даних з метою обміну інформацією. Селекція інформації, в основному, використовується в системах пересилання даних локальних комп'ютерних мереж, де і розглядається третя група протоколів.

6.3 Топології фізичних зв'язків

Під **топологією фізичних зв'язків комп'ютерної мережі** розуміється конфігурування графа, вершинам якого відповідають робочі станції (WS) мережі (можливо інше обладнання, наприклад: концентратори, комутатори, маршрутизатори), а ребрам – фізичні зв'язки між ними. Зауважимо, що конфігурування **фізичних зв'язків** визначається електричними з'єднаннями комп'ютерів між собою і може відрізнитися від конфігурування **логічних зв'язків** між станціями мережі. Логічними зв'язками є маршрути пересилання даних між станціями мережі, які залежать від типу та відповідного конфігурування мережевого обладнання.

Розглянемо найбільш поширені види топології мереж та основні визначення для її окремих видів.

Топологія повного з'єднання відповідає мережі, у якій кожна WS мережі пов'язана з усіма іншими робочими станціями (рис. 6.10). У цьому випадку для кожної пари WS повинна бути виділена окрема електрична лінія зв'язку. Даний вид топології застосовується досить рідко, оскільки кожна WS у мережі повинна мати таку кількість комунікаційних портів, яка є достатньою для з'єднання з іншою WS мережі.

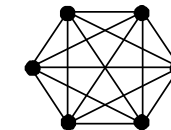


Рис. 6.10. Фізична топологія повного з'єднання.

Топологія шини. У цьому випадку WS з'єднуються одна з одною коаксіальним кабелем за схемою "монтажного АБО" (рис. 6.11). Інформація, яка передається від однієї WS до іншої робочої станції, розповсюджується, як правило, в обидва боки.

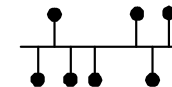


Рис. 6.11. Фізична топологія шини.

Основними перевагами такої схеми є дешевизна і простота розведення кабелю приміщеннями, можливість майже миттєвого широкомовного звертання до всіх станцій мережі. Головний недолік спільної шини полягає у її

низькій надійності: довільний дефект кабелю або довільного із численних роз'ємів повністю паралізує всю мережу. Іншим недоліком спільної шини є її невисока продуктивність, оскільки за такого способу з'єднання у кожний момент часу тільки одна робоча станція може пересилати дані до мережі. Тому, перепускна спроможність каналу зв'язку завжди поділяється тут між усіма станціями мережі.

Зіркова топологія. У цьому випадку кожна WS під'єднується окремим кабелем до спільного пристрою, який знаходиться у центрі мережі, і називається **концентратором** (рис. 6.12).

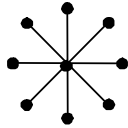


Рис. 6.12. Фізична зіркова топологія.

У функції концентратора входить скерування інформації, яка передається якоюсь WS, однієї або усім іншим робочим станціям мережі. Головна перевага даної топології перед спільною шиною – вища надійність. Пошкодження кабелю стосується лише тієї WS, до якої цей кабель приєднаний і тільки несправність концентратора може вивести з ладу всю мережу.

Крім того, концентратор може відігравати роль інтелектуального фільтра інформації, яка поступає від різних станцій у мережу, і, за необхідністю, блокувати заборонені адміністратором пересилання. До недоліків топології типу "зірка" відноситься вища вартість мережевого обладнання (вартість концентратора). Крім того, можливості з нарощення кількості станцій у мережі обмежуються кількістю портів концентратора.

Кільцева топологія. У мережах з кільцевою конфігурацією дані передаються кільцем від однієї робочої станції до іншої, як правило, в одному напрямку (рис. 6.13).

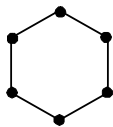


Рис. 6.13. Кільцева фізична топологія.

Це мережева топологія, у якій кожна станція має виключно два зв'язки з іншими станціями. Якщо WS розпізнає дані як "свої", тоді вона копіює їх у свій внутрішній буфер. Оскільки, у випадку виходу з ладу мережевого адаптера довільної станції переривається канал зв'язку між іншими станціями мережі, даний вид топології використовується в якості логічної топології.

Топологія дерева. Таке мережеве з'єднання WS з топологічної точки зору схоже на "зіркове", в якому окремі периферійні мережеві пристрої

можуть пересилати до одного іншого мережевого пристрою, або приймати тільки від нього в напрямку до центрального мережевого пристрою (рис. 6.14).

Як і у класичній топології "зірки", окремі мережеві пристрої можуть бути ізольовані від мережі внаслідок ліквідування одного зв'язку (гілки), наприклад, внаслідок аварії у лінії. У мережі за топологією дерева існує один виділений мережевий пристрій, який є коренем дерева.

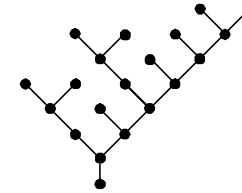


Рис. 6.14. Фізична топологія дерева.

Топологія сітки. Даний вид топології отримують із топології повного з'єднання шляхом видалення окремих можливих зв'язків (рис. 6.15). Це мережева топологія в якій існують щонайменше дві робочі станції з двома або більше шляхами між ними.

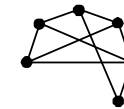


Рис. 6.15. Фізична топологія сітки.

Змішана (гібридна) топологія (рис. 6.16). Це поєднання двох або більшої кількості мережевих топологій. Можна подати приклади, коли дві об'єднані основні мережеві топології не змінюють характеру топології мережі і тому не створюють гібридної мережі. Наприклад, з'єднання мережі з топологією дерева дає мережу з такою ж топологією. Тому, гібридна топологія мережі виникає тільки тоді, коли з'єднані дві мережі з основними топологіями у результаті формують мережу, топологія якої не відповідає жодній з схем основних топологій. Наприклад, дві мережі із зірковою топологією, при об'єднанні, утворюють мережу з гібридною топологією. Гібридна топологія мережі формується, також, при об'єднанні мереж із різними видами топологій.

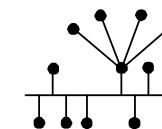


Рис. 6.16. Змішана (гібридна) фізична топологія.