

ЧЕРНІГІВСЬКИЙ НАЦІОНАЛЬНИЙ ПЕДАГОГІЧНИЙ УНІВЕРСИТЕТ
імені Т.Г. ШЕВЧЕНКА

ТЕХНОЛОГІЧНИЙ ФАКУЛЬТЕТ

Кафедра технологічної освіти та інформатики

Г.М.НІТЧЕНКО

**МЕТОДИЧНІ РЕКОМЕНДАЦІЇ
ДО ЛАБОРАТОРНИХ РОБІТ З КУРСУ
ОБ'ЄКТНО-ОРІЄНТОВАНЕ
ПРОГРАМУВАННЯ
ДЛЯ СТУДЕНТІВ НАПРЯМУ ПІДГОТОВКИ
6.010103 ТЕХНОЛОГІЧНА ОСВІТА**

Чернігів 2012

ББК 397р 30 – 252.43
УДК 378.147.091.33-027.22:004(075.8)
Н 69

Рецензенти:

Садовий М.І. доктор педагогічних наук, професор, завідувач кафедри загально-технічних дисциплін та методики трудового навчання Кіровоградського державного педагогічного університету імені В. Винниченка;

Торубара О.М. доктор педагогічних наук, професор, завідувач кафедрою професійної освіти та безпеки життєдіяльності Чернігівського національного педагогічного університету імені Т.Г.Шевченка

Нітченко Г.М.

Н 69 **Методичні рекомендації до лабораторних робіт з курсу "Об'єктно-орієнтоване" для студентів напряму підготовки 6.010103 «Технологічна освіта» / Г.М. Нітченко. – Чернігів: ЧНПУ, 2012. - 52с.**

ББК 397р 30 – 252.43
УДК 378.147.091.33-027.22:004(075.8)

Рекомендовано до друку Вченою радою технологічного факультету
Чернігівського національного педагогічного університету імені Т.Г. Шевченка
(протокол №2 від 21 вересня 2012 р.)

ЛАБОРАТОРНА РОБОТА №1 РОБОТА У ВІЗУАЛЬНОМУ РЕЖИМІ DELPHI

Мета роботи: Ознайомитись з різними видами проектів, типами файлів, пов'язаних з проектами, елементами середовища Delphi. Вивчити правила роботи у візуальному режимі Delphi. Навчитися працювати з формами, компонентами введення, виведення однорядкових даних.

Знати: основні властивості компонентів, що розглядаються.

Вміти: використовувати компоненти Form, Label, Edit, Button в створенні власних програм.

Обладнання: технічне забезпечення: ПЕОМ, програмне забезпечення: інтегроване середовище Delphi.

КОРОТКИ ТЕОРЕТИЧНІ ВІДОМОСТІ

1. Основні поняття. Технологія роботи у середовищі Delphi базується на ідеях об'єктно-орієнтованого та візуального програмування. Ідея об'єктно-орієнтованого програмування полягає в інкапсуляції (об'єднанні) даних і засобів їх опрацювання (методів) у тип, який називається класом. Конкретною змінною певного класу і є об'єкт. Прикладами об'єктів можуть бути елементи керування у вікні: кнопки, списки, текстові поля тощо. Середовище візуального програмування Delphi – це графічна автоматизована оболонка над об'єктно-орієнтованою версією мови Паскаль (Object Pascal). Якщо у мові Паскаль структурними одиницями є дані та команди, то тут такою структурною одиницею є візуальний об'єкт, який називається компонентом. Автоматизація програмування досягається завдяки можливості переносити компонент на форму (у програму) з палітри компонентів і змінювати його властивості, не вносячи вручну змін до програмного коду.

Формою називають компонент, який володіє властивостями вікна Windows і призначений для розташування інших компонентів. Компоненти на формі можуть бути видимими та невидимими. Перші призначені для організації діалогу з користувачем. Це різні кнопки, списки, текстові поля, зображення тощо. Вони відображаються на екрані під час виконання програми. Невидимі компоненти призначені для доступу до системних ресурсів комп'ютера.

Проект – це сукупність файлів, з яких складається Delphi-програма.

2. Інструменти середовища Delphi. Основними інструментами є:

- головне меню;
- панель інструментів;
- палітра компонентів (Component Palette);
- інспектор об'єктів (Object Inspector);
- вікно форми;
- редактор коду програми (Code Editor).

Ці інструменти стають доступними після запуску Delphi: три знаходяться у головному вікні (верхня частина екрана), а решта – в окремих вікнах.

3. Головне меню та панель інструментів. Головне меню складається з таких елементів: File, Edit, Search, View, Project, Run, Component, Database, Tools, Help (рис. 1).

Меню File містить стандартні команди для роботи з файлами проекту. За допомогою цих команд можна створити новий проект (New Application), нову форму або модуль (New Form і New Unit), відкрити чи закрити файл проекту (Open і Close), закрити всі відкриті файли (Close All), зберегти файл, проект або все відразу (Save, Save As, Save Project As, Save All).

За допомогою команд меню Edit можна вирівнювати компоненти відносно сітки та між собою (Align to Grid, Align), задавати порядок відображення компонентів, які перетинаються (Bring to Front, Send to Back), змінювати розмір вибраного компонента (Size), масштабувати візуальні компоненти (Scale) тощо.

Меню Search містить стандартні команди пошуку та заміни фрагмента тексту (Find, Replace, Search Again, Incremental Search) та інші. У меню View знаходяться команди візуалізації елементів середовища. Меню Project містить команди компіляції (Compile, Build All) та перевірки синтаксису програми (Syntax Check). Меню Run містить команди налагодження та запуску програми. Меню Component використовують для створення та інсталяції нових компонентів. Меню Database містить команди виклику інструментів бази даних. У меню Tools знаходяться команди для задання параметрів середовища.

Панель інструментів служить для розташування кнопок інструментів. На ній можуть міститися кнопки всіх згаданих команд.

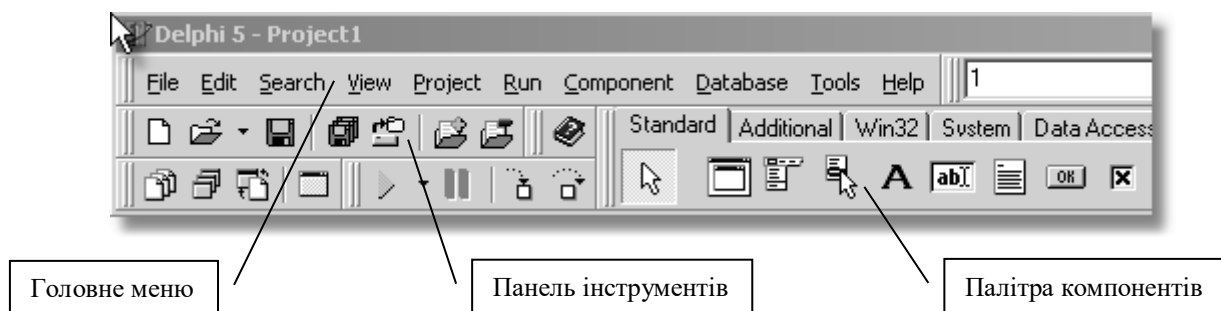


Рис. 1. Головне вікно Delphi

4. Палітра компонентів. Палітра компонентів розташована у головному вікні і має вигляд багатосторінкового блокнота. Кожній сторінці відповідає свій набір компонентів. Щоб помістити компонент у центрі вікна форми, двічі клацають на його піктограмі. Якщо потрібно розташувати компонент десь на формі, клацають один раз на його піктограмі і один раз у потрібному місці форми.

5. Інспектор об'єктів. За допомогою інспектора об'єктів можна задавати

початкові значення властивостей об'єкта та їхню реакцію на стандартні події. Вікно інспектора об'єктів містить список компонентів поточної форми, а також дві закладки: властивостей (Properties) та подій (Events). Розглянемо це вікно. Закладка властивостей складається з двох стовпців: лівий містить назви властивостей компонентів, а правий – їхні значення. Властивості можуть бути простими або комплексними. Комплексні властивості складаються з набору інших властивостей. Такі властивості в інспекторі об'єктів позначені символом "+", наприклад, +Font.

Закладка подій також має два стовпці. У лівому відображаються імена стандартних подій, на які об'єкт може реагувати, а в правому – імена методів-обробників (процедур), які будуть реалізовувати реакцію на подію. Кожній стандартній події відповідає назва методу, яка з'являється після подвійного клацання мишею у правому стовпці. У цей момент у вікно тексту програми додається шаблон базового коду (процедури) для відповідного методу, який треба заповнити.

6. Вікно форми. Форма – це вікно Windows, яке утворюється в одному з можливих для вікон стилів. Увесь внутрішній простір є робочою областю, яка має сітку вирівнювання для зручного розташування компонентів на формі. Для виконання групових операцій декілька компонентів можна об'єднувати. Для цього необхідно натиснути на ліву клавішу миші і переміщенням вказівника охопити всі потрібні компоненти. У групу долучаються компоненти, які хоча б частково попадають в охоплену область. Можна також долучити/вилучити окремий елемент. Для цього необхідно натиснути на клавішу **Shift** та, не відпускаючи її, вибрати мишею потрібний компонент на формі. Вилучення виокремлених компонентів чи групи виконується клавішею **Delete**. Переміщення виокремленого компонента в межах форми здійснюється мишею. Над компонентами та їхніми групами можна виконувати операції вирізання, копіювання в буфер обміну та вставлення з буфера.

7. Структура проекту. Проектом називають сукупність файлів, з яких Delphi створює готову для виконання програму. До складу кожного проекту обов'язково входять наступні файли:

- файл проекту *.dpr. Це невеликий файл з програмним кодом мовою Object Pascal, який містить посилання на всі файли проекту та ініціалізує програму;
- файли опису всіх форм, які входять у проект: файл модуля *.pas і файл форми *.dfm. Будь-якій формі проекту відповідає власний модуль;
- файл ресурсів програми *.res. У ньому описані ресурси, які не входять у форму, наприклад, піктограма програми;
- файл параметрів проекту *.dof;
- файли параметрів середовища *.drf, *.dsk, *.dsm. Ці файли створюються лише після компіляції проекту.

Для збереження Delphi-проекту необхідно задати імена модулів (автоматично пропонуються імена Unit1.pas, Unit2.pas, ...) та ім'я проекту (Project1.dpr). Ці імена можна змінити на власні. Для переміщення Delphi-

проекту на інший комп'ютер необхідно мати лише файли таких типів: *.dpr, *.dfm, *.pas, *.res. Інші файли створюються автоматично.

8. Редактор коду. Редактор коду є в окремому вікні. Це вікно організоване як багатосторінковий блокнот відкритих у даний момент файлів. Під час відкривання нового проекту в модуль Unit1.pas, який відповідає формі Form1, редактор автоматично заносить програмний код опису цієї форми. Під час додавання нових компонентів у вікно форми у програму автоматично заноситься коди з описами параметрів цих компонентів (висота, ширина, розташування, стиль тощо). Долучення певного об'єкта чи застосування методу до нього веде до появи заготовки базового коду відповідної процедури у вікні редактора. Заготовка (шаблон) складається з заголовка процедури та ключових слів **begin** і **end**. Заготовку заповнює користувач. Завершується модуль власною командою **end**.

САМОСТІЙНА ПІДГОТОВКА ДО ВИКОНАННЯ ЛАБОРАТОРНОЇ РОБОТИ

1. Записати в зошит тему, мету, обладнання, практичні завдання свого варіанту (вибирається згідно номеру комп'ютера).

2. По літературі до лабораторної роботи вивчити основи роботи в інтегрованому середовищі розробки Delphi.

3. По інструкції до лабораторної роботи ознайомитись з порядком виконання роботи.

ПОРЯДОК ВИКОНАННЯ РОБОТИ

1. Запустити Delphi.

2. Виконати практичні завдання відповідно до свого варіанту.

3. Оформити звіт (завдання до роботи, тексти програм, висновки).

Варіант №1

1. Написати програму калькулятор, в якій за введеними двома множниками знаходився б їх добуток.

2. Написати програму, яка обчислює повний гідростатичний тиск у точці рідини в стані спокою, якщо відомо атмосферний тиск p_0 (приблизно 100кПа), густина ρ та глибина занурення h . $p = p_0 + g\rho h$.

Варіант №2

1. Написати програму, в якій за введеним значення аргументу обчислюється значення функції $y = \sin(x)$.

2. Написати програму, яка обчислює значення електромагнітного моменту синхронного двигуна. $\varphi = \pi/6$. $k = 5,34$. $M_{em} = k\Phi I \cos\varphi$

Варіант №3

1. Написати програму, яка переводить швидкість км/год в м/с.

2. Написати програму, яка визначає повний опір Z для кола з послідовним з'єднанням активного опору та індуктивності. $\omega = -\pi/2$. $Z = \sqrt{R^2 + \omega^2 L^2}$

Варіант №4

1. Написати програму, яка переводить значення температури по шкалі Цельсія (С) в градуси по шкалі Фаренгейта (F) за формулою: $F=1,8C+32$.
2. Написати програму, яка визначає силу тиску на горизонтальну площадку ω . Відомо ω (m^2), висота h . $P=g\rho h\omega$.

Варіант №5

1. Написати програму калькулятор, яка визначає щільність повітря на заданій висоті ($\rho=\rho_0e^{-hz}$, $\rho_0=1,29$ кг/м³, $z=1,25\cdot 10^{-4}$ 1/м).
2. Написати програму, яка обчислює повний опір Z кола з послідовним з'єднанням активного опору, індуктивності та ємності. $\omega = \pi/3$.

$$Z = \sqrt{R^2 + (\omega L - \frac{1}{\omega C})^2}$$

Варіант №6

1. Написати програму, яка переводить значення відстані в км в дюйми (1 дюйм = 2,54 см).
2. Написати програму, яка обчислює потужність для однофазного змінного струму. $\omega = \pi/6$. $P = UI \cos \omega$

Варіант №7

1. Написати програму, яка переводить значення ваги в кг у фунти (1 фунт=400 г).
2. Написати програму, яка обчислює значення електрорушійної сили. $\omega = -2\pi/3$. $e = E \sin \omega t$

Варіант №8

1. Написати програму, яка підраховує вартість пального у циліндричному бакові, якщо відома ціна одного літру пального.
2. Написати програму, яка визначає значення миттєвої напруги u . $\omega = \pi/6$. $u = U_m \sin \omega t$.

Варіант №9

1. Написати програму, яка підраховує вартість відрізу тканини, якщо відома вартість одного метру тканини та розміри відрізу.
2. Написати програму, яка визначає значення миттєвого струму i . $\omega = \pi/5$. $i = I_m \sin \omega t$

Варіант №10

1. Написати програму, яка за показниками лічильника в два різні моменти часу визначає вартість витраченої електроенергії за відповідний проміжок часу.
2. Написати програму, яка обчислює значення електромагнітного моменту синхронного двигуна. $\varphi = 2\pi/3$. $k=5,34$. $M_{em} = k\Phi I \cos \varphi$

Варіант №11

1. Написати програму, яка підраховує вартість цукру, якщо відома його маса та ціна за кілограм.
2. Написати програму, яка обчислює значення електромагнітного моменту синхронного двигуна. $k=5,34$. $M_{em} = k\Phi I \cos \varphi$

ПИТАННЯ ДЛЯ САМОСТІЙНОЇ РОБОТИ

1. Опишіть призначення компонента Edit?
2. Опишіть призначення компонента Form?
3. Опишіть призначення компонента Label?
4. Опишіть призначення компонента Button?
5. Перечисліть та опишіть будову та призначення панелі інструментів інтегрованого середовища Delphi?
6. Як можна налаштувати інтегроване середовище Delphi?
7. Опишіть призначення інспектора об'єктів (Object Inspector)?
8. Опишіть призначення редактора кода програми (Code Editor)?
9. Опишіть призначення палітри компонентів (Component Palette)?
10. Опишіть, які файли входять в структуру проекту та їх призначення?

ЛІТЕРАТУРА

1. Бадд Т. Объектно-ориентированное программирование в действии / Перев. с англ. – СПб.: Питер, 1997. – 464 с.
2. Бобровский С. Delphi: учебный курс – СПб: Питер, 2001. – 640 с.
3. Буч Г. Объектно-ориентированный анализ и проектирование, 2-е изд./ Пер. с англ. – М.: «Издательство Бином», СПб.: «Невский диалект», 1999. – 560 с.
4. Дантемманн Д., Мишель Д., Тейлор Д. Программирование в среде Delphi. Киев: Диа-Софт Лтд., 1995.
5. Дарахвидзе П., Марков Е. Delphi-среда визуального программирования. СПб.: ВНУ-Санкт-Петербург, 1995.
6. Жуков А. Изучаем Delphi – СПб: Питер, 2000. – 352 с.
7. Конопка Р. Создание оригинальных компонентов в среде Delphi. К.: Диа-Софт Лтд., 1996.
8. Миллер Т., Пауэл Д. Использование Delphi 3. – К.: Диалектика, 1997. – 768 с.
9. Оузьер Д., Гробман С, Батсон С. Delphi 2. Освой самостоятельно: Пер. с англ. – М.: Восточная Книжная Компания, 1997. – 624 с.
10. Соколов В.В. Основы програмування в Delphi // Комп'ютер у школі та сім'ї. – №3, 2001. – С. 21-25.
11. Черняхівський В.В. Delphi-4: сучасна технологія візуального програмування. – Львів: БаК, 1999. – 196 с.

ЗВІТ ДО РОБОТИ

1. Оформити лабораторну роботу в зошиті. Записати тему, мету, відповіді на запитання для самостійної підготовки.
2. Виконати завдання згідно свого варіанту.
3. Зробити висновок.

ЛАБОРАТОРНА РОБОТА №2 ВВЕДЕННЯ ТА ВИВЕДЕННЯ ДАНИХ З ФАЙЛІВ

Мета роботи: Ознайомитись з компонентом Мемо, вікном введення InputBox, роботою процедури ShowMessage та функцією MessageDlg. Навчитися створювати програми, які здійснюють створення, обробку текстових файлів.

Знати основні властивості компонентів, що розглядаються.

Вміти: використовувати компонент Мемо, процедури та функції ShowMessage та MessageDlg в створенні власних програм.

Обладнання: технічне забезпечення: ПЕОМ, програмне забезпечення: інтегроване середовище Delphi.

КОРОТКИ ТЕОРЕТИЧНІ ВІДОМОСТІ

Компонент в палітрі компонентів на закладці Standart (рис. 1). Основні його властивості описані в табл. 1.

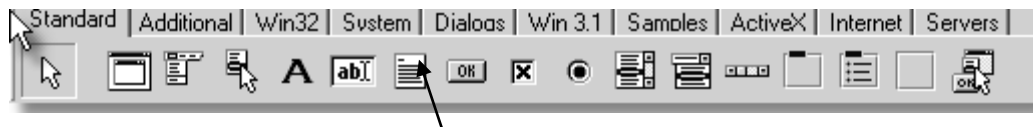


Рис. 1. Компонент Мемо

Таблиця 1.

<i>Властивість</i>	<i>Опис</i>
Name	Ім'я компонента.
Text	Текст, що знаходиться в полі Мемо. Розглядається як єдине ціле
Lines	Масив рядків, що відповідає вмісту поля. Доступ до рядка здійснюється по номеру. Рядки нумеруються з нуля
Lines.Count	Кількість рядків тексту в полі Мемо
Left	Відстань від лівої границі поля до лівої границі форми
Top	Відстань від верхньої границі поля до верхньої границі форми
Height	Висота поля
Width	Ширина поля
Font	Шрифт, використаний для відображення тексту, що вводиться

Для інтерактивного вводу та виводу інформації використовуються вбудовані підпрограми діалогових вікон InputBox() та MessageBox(). Для зчитування даних в рядкову змінну використовується оператор:

```
рядкава_змінна := InputBox(заголовок_вікна, заголовок_рядка,  
                           рядок_за_замовчуванням);
```

Наступний оператор виводить діалогове вікно з надписом *повідомлення* та кнопкою ОК. В заголовку діалогового вікна виводиться ім'я файлу програми:

```
Змінна тип_Word := MessageDlg(повідомлення, тип_діалогу, кнопки,  
контекст_довідки);
```

Тип_діалогу (вигляд відображеного вікна) залежить від вибору константи:

mtWarning	Вікно зауважень, яке містить жовтий знак оклику.
mtError	Вікно помилок, яке містить червоний стоп-сигнал.
mtInformation	Інформаційне вікно, яке містить блакитний символ “i”.
mtConfirmation	Вікно підтвердження, яке містить зелений знак питання
mtCustom	Вікно без малюнка. Заголовок вікна містить ім'я файлу програми.

Виведення кнопок залежить від константи:

mbYes	Кнопка з надписом	'Yes'
mbNo	Кнопка з надписом	'No'
mbOK	Кнопка з надписом	'OK'
mbCancel	Кнопка з надписом	'Cancel'
mbHelp	Кнопка з надписом	'Help'
mbAbort	Кнопка з надписом	'Abort'
mbRetry	Кнопка з надписом	'Retry'
mbIgnore	Кнопка з надписом	'Ignore'
mbAll	Кнопка з надписом	'All'

Процедура ShowMessage(*повідомлення*) виводить діалогове вікно з надписом повідомлення та кнопкою ОК. В заголовку вікна виводиться ім'я файлу програми.

САМОСТІЙНА ПІДГОТОВКА ДО ВИКОНАННЯ ЛАБОРАТОРНОЇ РОБОТИ

1. Записати в зошит тему, мету, обладнання, практичні завдання свого варіанту (вибирається згідно номеру комп'ютера).
2. По літературі до лабораторної роботи вивчити основи роботи з компонентом Memo, вікном введення InputBox, з процедурою ShowMessage та функцією MessageDlg в інтегрованому середовищі розробки Delphi.
3. По інструкції до лабораторної роботи ознайомитись з порядком виконання роботи.

ПОРЯДОК ВИКОНАННЯ РОБОТИ

1. Запустити Delphi.
2. Виконати практичні завдання відповідно до свого варіанту.
3. Оформити звіт (завдання до роботи, тексти програм, висновки).

Варіант №1

1. На формі розмістити два компонента для введення багаторядкового тексту. Один компонент використовувати для введення тексту, другий – для виведення тексту після певного перетворення, яке здійснюється при натисненні кнопки. Кожний рядок тексту переписати в зворотному порядку.
2. Дано текстовий файл. Підрахувати кількість входжень у текст файлу літери ‘а’.

Варіант №2

1. Дано текстовий файл. Створити спрощений словник, в якому на задану літеру з файлу виводяться всі слова, які починаються на цю літеру.
2. На формі розмістити два компонента для введення багаторядкового тексту. Один компонент використовувати для введення тексту, другий – для виведення тексту після певного перетворення, яке здійснюється при натисненні кнопки. З кожного рядка тексту вилучити останні три символи.

Варіант №3

1. На формі розмістити два компонента для введення багаторядкового тексту. Один компонент використовувати для введення тексту, другий – для виведення тексту після певного перетворення, яке здійснюється при натисненні кнопки. Розтягнути текст, поставивши після кожного символу пропуск.
2. Створити спрощений текстовий редактор, в якому передбачити завантаження текстового файлу, можливість виконання редакційних операцій, збереження виправленого тексту.

Варіант №4

1. На формі розмістити два компонента для введення багаторядкового тексту. Один компонент використовувати для введення тексту, другий – для виведення тексту після певного перетворення, яке здійснюється при натисненні кнопки. Кожний рядок тексту розбити на дві частини посередині.
2. Записати у файл вміст компонента Мето у зворотньому порядку.

Варіант №5

1. На формі розмістити два компонента для введення багаторядкового тексту. Один компонент використовувати для введення тексту, другий – для виведення тексту після певного перетворення, яке здійснюється при натисненні кнопки. Стиснути текст, замінюючи символи, що повторюються одним символом.
2. Створити спрощений телефонний довідник, в якому на задану літеру з файлу виводяться прізвища, що починаються на цю літеру, та номери телефонів. Дано текстовий файл. Переписати в інший файл всі рядки з максимальною довжиною та визначити їх кількість.

Варіант №6

1. Створити спрощений текстовий редактор, в якому передбачити завантаження текстового файлу, можливість виконання редакційних операцій, збереження виправленого тексту.
2. Дано текстовий файл. Надрукувати рядки, що мають парну довжину, підрахувавши їх кількість.

Варіант №7

1. На формі розмістити два компонента для введення багаторядкового тексту. Один компонент використовувати для введення тексту, другий – для виведення тексту після певного перетворення, яке здійснюється при натисненні кнопки. В тексті великі літери замінити на маленькі.
2. Створити спрощений словник, в якому на задану літеру з файлу виводяться всі слова, які починаються на цю літеру.

Варіант №8

1. На формі розмістити два компонента для введення багаторядкового тексту. Один компонент використовувати для введення тексту, другий – для виведення тексту після певного перетворення, яке здійснюється при натисненні кнопки. Вилучити з тексту, який міститься в першому компоненті кожний другий рядок.
2. Дано текстовий файл f. Переписати елементи файлу в файл g, при чому перед кожним рядком написати порядковий номер ряду (наприклад: 'перший', 'другий' і т.д.). Порядок рядків повинен бути збережений.

Варіант №9

1. На формі розмістити два компонента для введення багаторядкового тексту. Один компонент використовувати для введення тексту, другий – для виведення тексту після певного перетворення, яке здійснюється при натисненні кнопки. Кожний рядок тексту взяти в квадратні дужки.
2. Дано текстовий файл. Переписати у другий файл рядки у зворотньому порядку.

Варіант №10

1. На формі розмістити два компонента для введення багаторядкового тексту. Один компонент використовувати для введення тексту, другий – для виведення тексту після певного перетворення, яке здійснюється при натисненні кнопки. В тексті маленькі літери замінити на великі.
2. Дано текстовий файл. Надрукувати рядки, що мають непарну довжину, підрахувавши їх кількість.

Варіант №11

1. На формі розмістити два компонента для введення багаторядкового тексту. Один компонент використовувати для введення тексту, другий – для виведення тексту після певного перетворення, яке здійснюється при натисненні кнопки. Із тексту вилучити всі входження заданого символу.
2. Дано текстовий файл. Підрахувати, скільки раз в тексті зустрічаються пропуски.

ПИТАННЯ ДЛЯ САМОСТІЙНОЇ РОБОТИ

1. Поясніть, що таке вбудовані вікна в Delphi?

2. Опишіть призначення та основні властивості компонента Memo?
3. Опишіть призначення та основні властивості компонента Edit?
4. Як можна очистити поле вводу вміст компонентів Memo та Edit?
3. Як програміст може підвищити зрозумілість кода розробляємої програми?
4. Які є способи завершення програми в Delphi?
5. Опишіть призначення та параметри процедури ShowMessage?
6. Опишіть призначення та параметри діалогового вікна введення InputBox?
7. Опишіть призначення та параметри функції MessageDlg?
8. Опишіть доступний синтаксис коментаріїв в Delphi?
9. Поясніть чим змінна відрізняється від константи?
10. Навіщо змінним потрібно присвоювати початкові значення?

ЛІТЕРАТУРА

1. Бадд Т. Объектно-ориентированное программирование в действии / Перев. с англ. – СПб.: Питер, 1997. – 464 с.
2. Бобровский С. Delphi: учебный курс – СПб: Питер, 2001. – 640 с.
3. Буч Г. Объектно-ориентированный анализ и проектирование, 2-е изд./ Пер. с англ. – М.: «Издательство Бином», СПб.: «Невский диалект», 1999. – 560 с.
4. Дантемманн Д., Мишель Д., Тейлор Д. Программирование в среде Delphi. Киев: Диа-Софт Лтд., 1995.
5. Дарахвидзе П., Марков Е. Delphi-среда визуального программирования. СПб.: ВНУ-Санкт-Петербург, 1995.
6. Жуков А. Изучаем Delphi – СПб: Питер, 2000. – 352 с.
7. Конопка Р. Создание оригинальных компонентов в среде Delphi. К.: Диа-Софт Лтд., 1996.
8. Миллер Т., Пауэл Д. Использование Delphi 3. – К.: Диалектика, 1997. – 768 с.
9. Оузьер Д., Гробман С., Батсон С. Delphi 2. Освой самостоятельно: Пер. с англ. – М.: Восточная Книжная Компания, 1997. – 624 с.
10. Соколов В.В. Основы програмування в Delphi // Комп'ютер у школі та сім'ї. – №3, 2001. – С. 21-25.
11. Черняхівський В.В. Delphi-4: сучасна технологія візуального програмування. – Львів: БАК, 1999. – 196 с.

ЗВІТ ДО РОБОТИ

1. Оформити лабораторну роботу в зошиті. Записати тему, мету, відповіді на запитання для самостійної підготовки.
2. Виконати завдання згідно свого варіанту.
3. Зробити висновок.

ЛАБОРАТОРНА РОБОТА №3 ОРГАНІЗАЦІЯ ВИБОРУ

Мета роботи: Ознайомитись з компонентами RadioButton, RadioGroup, CheckBox, MainMenu та їх властивостями. Навчитися створювати програми, які мають головне меню та містять компоненти вибору.

Знати: основні властивості розглядаємих компонентів.

Вміти: використовувати компоненти RadioButton, RadioGroup, CheckBox та MainMenu в створенні власних програм.

Обладнання: технічне забезпечення: ПЕОМ, програмне забезпечення: інтегроване середовище Delphi.

КОРОТКИ ТЕОРЕТИЧНІ ВІДОМОСТІ

За допомогою не візуального компонента MainMenu можна створити в середовищі Delphi головне меню форми (рис. 1, а). Цей елемент складається з рядка меню та асоційованими з ним пунктами меню, що розкриваються. Після того, як компонент розміщений на формі подвійний клік мишкою активізує редактор меню, де налаштовуються основні параметри.

Компонент RadioButton представляє собою залежну кнопку, стан якої визначається станом інших кнопок групи (рис. 1, б). Властивість, яка відповідає за включення – Checked.

Якщо кнопка вибрана, то Checked = True;

якщо кнопка не вибрана, то Checked = False.

Компонент CheckBox (рис. 1, в) представляє собою незалежну кнопку (премикач). Властивість, яка відповідає за включення, як і в компонента RadioButton – Checked.

Якщо кнопка вибрана, то Checked = True;

якщо кнопка не вибрана, то Checked = False.

Компонент RadioGroup представляє собою групу елементів, подібних RadioButton. Встановлення опцій групи взаємовиключаюча (рис. 1, г). Властивість Items відповідає за елементи групи та їх кількість.

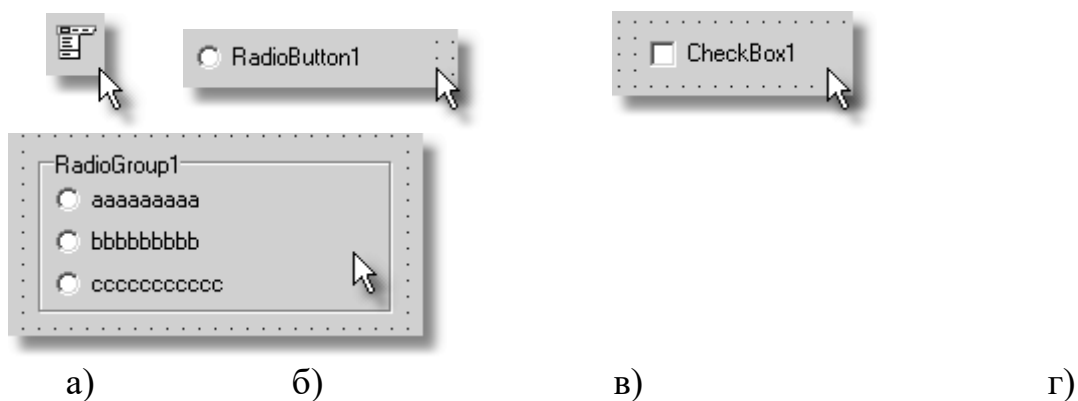


Рис. 1. Компоненти MainMenu, RadioButton, CheckBox та RadioGroup.

САМОСТІЙНА ПІДГОТОВКА ДО ВИКОНАННЯ ЛАБОРАТОРНОЇ РОБОТИ

1. Записати в зошит тему, мету, обладнання, практичні завдання свого варіанту (вибирається згідно номеру комп'ютера).
2. По літературі до лабораторної роботи вивчити основи роботи з компонентами RadioButton, RadioGroup, CheckBox, MainMenu та їх властивостями в інтегрованому середовищі розробки Delphi.
3. По інструкції до лабораторної роботи ознайомитись з порядком виконання роботи.

ПОРЯДОК ВИКОНАННЯ РОБОТИ

1. Запустити Delphi.
2. Виконати практичні завдання відповідно до свого варіанту.
3. Оформити звіт (завдання до роботи, тексти програм, висновки).

Варіант №1

1. Написати програму, яка підраховує вартість пального у циліндричному баку, якщо відома вартість 1 літра пального. На формі передбачити вибір російської та української розкладки для компонент форми.
2. Внесіть зміни у друге завдання лабораторної роботи №2 таким чином, щоб відкривати та зберігати файли через пункти меню. Крім того додати в меню відомості про автора та дату створення програми вихід з програми.

Варіант №2

1. Умовно скласти розклад занять на два дні. За вибраним днем та номером пари висвітлити прізвище викладача, що буде її проводити, номер аудиторії, де буде проводитись пара.
2. Внесіть зміни у друге завдання лабораторної роботи №2 таким чином, щоб відкривати та зберігати файли через пункти меню. Крім того додати в меню відомості про автора та дату створення програми вихід з програми.

Варіант №3

1. За вибраним типом жанру фільму, і часом сеансу висвітлити вартість квитка в кінотеатр.
2. Внесіть зміни у друге завдання лабораторної роботи №2 таким чином, щоб відкривати та зберігати файли через пункти меню. Крім того додати в меню відомості про автора та дату створення програми вихід з програми.

Варіант №4

1. Скласти фрагмент контролюючої програми, в якому передбачити виведення питання з трьома варіантами відповіді. Після вибору відповіді висвітлити інформацію про правильність відповіді.
2. Внесіть зміни у друге завдання лабораторної роботи №2 таким чином, щоб відкривати та зберігати файли через пункти меню. Крім того додати в меню відомості про автора та дату створення програми, вихід з програми.

Варіант №5

1. Обчислити вартість путівки в домі відпочинку в залежності від введеної кількості днів відпочинку, вибраного типу номеру (люкс, двомісна, трьохмісна кімната), вибраного типу харчування (без харчування, з повним харчуванням, шведський стіл).
2. Внесіть зміни у друге завдання лабораторної роботи №2 таким чином, щоб відкривати та зберігати файли через пункти меню. Крім того додати в меню відомості про автора та дату створення програми вихід з програми.

Варіант №6

1. На послуги Internet в залежності від часу доби існують різні тарифи оплати. У вікні програми передбачити вибір режиму роботи в Internet та висвітлення відповідного тарифу.
2. Внесіть зміни у друге завдання лабораторної роботи №2 таким чином, щоб відкривати та зберігати файли через пункти меню. Крім того додати в меню відомості про автора та дату створення програми вихід з програми.

Варіант №7

1. Скласти програму для знаходження оптимальної ваги людини відповідно до віку і зросту:
для чоловіків: $50 + (\text{зріст} - 150) * 0,52 + (\text{вік} - 21) : 4$;
для жінок: $50 + (\text{зріст} - 150) * 0,75 + (\text{вік} - 21) : 5$.
2. Внесіть зміни у друге завдання лабораторної роботи №2 таким чином, щоб відкривати та зберігати файли через пункти меню. Крім того додати в меню відомості про автора та дату створення програми вихід з програми.

Варіант №8

1. Ввести два числа. Виконати вибрану арифметичну дію з цими числами та висвітлити результат. Передбачити можливість знаходження суми, добутку, віднімання, ділення.
2. Внесіть зміни у друге завдання лабораторної роботи №2 таким чином, щоб відкривати та зберігати файли через пункти меню. Крім того додати в меню відомості про автора та дату створення програми вихід з програми.

Варіант №9

1. Знайти значення однієї з функцій. Номер функції вибирає користувач.
1) $\sin(x) + 5$ 2). $\cos x - x^2$ 3). $X^3 - \cos^2 x$ 4). $\sin^2 2x + x^2$
2. Внесіть зміни у друге завдання лабораторної роботи №2 таким чином, щоб відкривати та зберігати файли через пункти меню. Крім того додати в меню відомості про автора та дату створення програми вихід з програми.

Варіант №10

1. Скласти програму для знаходження ціни пального на автомобільній заправці. Ціна підраховується в залежності від вибраного типу пального, крім того передбачити використання 30%, 40%, 50%-их знижок.
2. Внесіть зміни у друге завдання лабораторної роботи №2 таким чином, щоб відкривати та зберігати файли через пункти меню. Крім того додати в меню відомості про автора та дату створення програми вихід з програми.

Варіант №11

1. Написати програму нарахування заробітної плати співробітнику фірми в залежності від введеної кількості робочих днів та вибраної посади.
2. Внесіть зміни у друге завдання лабораторної роботи №2 таким чином, щоб відкривати та зберігати файли через пункти меню. Крім того додати в меню відомості про автора та дату створення програми вихід з програми.

ПИТАННЯ ДЛЯ САМОСТІЙНОЇ РОБОТИ

1. Яка різниця між компонентами RadioButton та RadioGroup?
2. Як створити контекстне меню для певного компонента на формі?
3. Яка різниця між компонентами GroupBox та CheckBox?
4. Опишіть призначення на основні властивості компонента RadioButton?
5. Опишіть призначення на основні властивості компонента RadioGroup?
6. Опишіть призначення на основні властивості компонента CheckBox?
7. Опишіть призначення на основні властивості компонента MainMenu?
8. Яка різниця між компонентами RadioButton та CheckBox?
9. Опишіть призначення інспектора об'єктів (Object Inspector)?
10. Опишіть, які файли входять в структуру проекту та їх призначення?

ЛІТЕРАТУРА

1. Бадд Т. Объектно-ориентированное программирование в действии / Перев. с англ. – СПб.: Питер, 1997. – 464 с.
2. Бобровский С. Delphi: учебный курс – СПб: Питер, 2001. – 640 с.
3. Буч Г. Объектно-ориентированный анализ и проектирование, 2-е изд./ Пер. с англ. – М.: «Издательство Бином», СПб.: «Невский диалект», 1999. – 560 с.
4. Дантеманн Д., Мишель Д., Тейлор Д. Программирование в среде Delphi. Киев: Диа-Софт Лтд., 1995.
5. Дарахвидзе П., Марков Е. Delphi-среда визуального программирования. СПб.: ВНУ-Санкт-Петербург, 1995.
6. Жуков А. Изучаем Delphi – СПб: Питер, 2000. – 352 с.
7. Конопка Р. Создание оригинальных компонентов в среде Delphi. К.: Диа-Софт Лтд., 1996.
8. Миллер Т., Пауэл Д. Использование Delphi 3. – К.: Диалектика, 1997. – 768 с.
9. Оузьер Д., Гробман С, Батсон С. Delphi 2. Освой самостоятельно: Пер. с англ. – М.: Восточная Книжная Компания, 1997. – 624 с.
10. Соколов В.В. Основы програмування в Delphi // Комп'ютер у школі та сім'ї. – №3, 2001. – С. 21-25.
11. Черняхівський В.В. Delphi-4: сучасна технологія візуального програмування. – Львів: БаК, 1999. – 196 с.

ЗВІТ ДО РОБОТИ

4. Оформити лабораторну роботу в зошиті. Записати тему, мету, відповіді на запитання для самостійної підготовки.
5. Виконати завдання згідно свого варіанту.
6. Зробити висновок.

ЛАБОРАТОРНА РОБОТА №4 УПРАВЛІННЯ РОБОТОЮ ПРОЕКТУ

Мета роботи: Ознайомитись з компонентами `ListBox`, `ComboBox`, `CheckBox`, `ProgressBar`, `Gauge` їх властивостями. Навчитися створювати програми, які опрацьовують списки даних та відображають індикацію виконання певних процесів.

Знати: основні властивості розглядаємих компонентів.

Вміти: використовувати компоненти `ListBox`, `ComboBox`, `CheckBox`, `ProgressBar`, `Gauge` в створенні власних програм.

Обладнання: технічне забезпечення: ПЕОМ, програмне забезпечення: інтегроване середовище Delphi.

КОРОТКИ ТЕОРЕТИЧНІ ВІДОМОСТІ

Компонент `TListBox` – відображає список рядків і дозволяє користувачеві вибрати з нього необхідні рядки.

У список автоматично додаються смуги прокручування, якщо всі рядки не вміщуються у вікні компонента.

Основна властивість компонента, що містить список рядків, – **Items**, що має тип **TStrings**. Заповнити його під час проектування можна, нажавши кнопку із крапками біля цієї властивості у вікні Інспектора Об'єктів. Під час виконання працювати із цією властивістю можна, користуючись властивостями й методами класу **TStrings** – **Clear**, **Add** і іншими.

У компоненті **TListBox** є властивість **MultiSelect**, що дозволяє користувачеві множинний вибір у списку. Якщо **MultiSelect = false** (значення за замовчуванням), то користувач може вибрати тільки один елемент списку. У цьому випадку можна довідатися індекс обраного рядка із властивості **ItemIndex**, доступного тільки під час виконання. Якщо жодна рядок не обраний, то **ItemIndex = -1**. Початкове значення **ItemIndex** неможливо задати під час проектування. За замовчуванням **ItemIndex = -1**. Це означає, що жоден елемент списку не обраний. Якщо ви хочете задати цій властивості якесь інше значення, тобто встановити вибір за замовчуванням, що буде показаний у момент початку роботи додатка, то зробити це можна, наприклад, в оброблювачі події **OnCreate** форми, увівши в нього оператор виду `ListBox1.ItemIndex:=0`.

Якщо допускається множинний вибір (**MultiSelect = true**), те значення **ItemIndex** відповідає тому елементу списку, що перебуває у фокусі. При множинному виборі перевірити, чи обраний даний елемент, можна перевіривши властивість **SelectedCIndex: Integer** типу **Boolean**.

На спосіб множинного вибору при **MultiSelect = true** впливає ще властивість **ExtendedSelect**. Якщо **ExtendedSelect = true**, то користувач може виділити інтервал елементів, виділивши один з них, потім нажавши клавішу **Shift** і перевіривши курсор до іншого елемента. Виділити не прилягаючий друг до друга елементи користувач може, якщо буде втримувати під час вибору

натиснутої клавішу Ctrl. Якщо ж **ExtendedSelect = false**, те клавіші Shift і Ctrl при виборі не працюють. Властивість **Columns** визначає число стовпців, у яких буде відображатися список, якщо він не міститься цілком у вікні компонента **ListBox**.

Властивість **Sorted** дозволяє впорядкувати список за алфавітом. При **Sorted = true** нові рядки в список додаються не в кінець, а за алфавітом.

Властивість **Style**, установлене в **lbStandard** (значення за замовчанням) відповідає списку рядків. Інші значення **Style** дозволяють відображати в списку не тільки текст, але й зображення.

Компонент **TComboBox** поєднує функції компонентів **TListBox** – списку, і **TEdit** – вікна редагування. Компонент дозволяє користувачеві вибрати зі списку необхідний рядок або задати як вибір власний текст. Список може відображатися в розгорнутому виді або як, що випадає список. Відмінність **TComboBox** від схожого по функціях компонента **TListBox** полягає в наступному:

- **TComboBox** дозволяє користувачеві редагувати список, а **TListBox** не дозволяє;

- в **TComboBox** список може бути розгорнутий або згорнутий, а в **TListBox** він завжди розгорнутий;

- **TListBox** може допускати множинний вибір, а в **TComboBox** користувач завжди повинен вибрати тільки один елемент.

Основне властивість компонента, що містить список рядків, – **Items**, що має тип **TStrings**. Заповнити його під час проектування можна, нажавши кнопку з крапками біля цієї властивості у вікні Інспектора Об'єктів. Під час виконання працювати із цією властивістю можна, користуючись властивостями й методами класу **TStrings** – **Clear**, **Add** і іншими. Стиль зображення списку визначається властивістю **Style**:

csDropDown список, що випадає, з рядками однакової висоти й з вікном редагування, що дозволяє користувачеві вводити або редагувати текст;

csSimple розгорнутий список з рядками однакової висоти й з вікном редагування, що дозволяє користувачеві вводити або редагувати текст;

caDropDownList список, що випадає, з рядками однакової висоти, не утримуючий вікна редагування.

Вибір користувача або уведений їм текст можна визначити за значенням властивості **Text**. Індекс обраного користувачем елемента списку можна визначити по властивості **ItemIndex** (доступно тільки під час виконання). Якщо у вікні проводилося редагування даних, то **ItemIndex = -1**. Початкове значення **ItemIndex** неможливо задати під час проектування. За замовчуванням **ItemIndex = -1**. Якщо ви хочете задати цій властивості якесь інше значення, тобто установити вибір за замовчуванням, що буде показаний

у момент початку роботи додатка, то зробити це можна, наприклад, в оброблювачі події **OnCreate** форми, увівши в нього оператор виду `ComboBox1.ItemIndex:=0;`

Властивість **MaxLength** визначає максимальне число символів, які користувач може ввести у вікно редагування. Якщо **MaxLength** = 0, то число символів, що вводять, не обмежено.

Свойство **Sorted** дозволяє впорядкувати список за алфавітом. При **Sorted** = **true** нові рядки в список додаються не в кінець, а за алфавітом.

Текст обраної або написаної користувачем рядка перебуває у властивості **Text**. Індекс обраного рядка можна довідатися з властивості **ItemIndex**.

Основна подія компонента – **OnChange** настає при зміні тексту у вікні редагування в результаті прямого редагування тексту або в результаті вибору зі списку.

САМОСТІЙНА ПІДГОТОВКА ДО ВИКОНАННЯ ЛАБОРАТОРНОЇ РОБОТИ

1. Записати в зошит тему, мету, обладнання, практичні завдання свого варіанту (вибирається згідно номеру комп'ютера).

2. По літературі до лабораторної роботи вивчити основи роботи з компонентами `ListBox`, `ComboBox`, `CheckBox`, `ProgressBar`, `Gauge` їх властивостями в інтегрованому середовищі розробки Delphi.

3. По інструкції до лабораторної роботи ознайомитись з порядком виконання роботи.

ПОРЯДОК ВИКОНАННЯ РОБОТИ

1. Запустити Delphi.
2. Виконати практичні завдання відповідно до свого варіанту.
3. Оформити звіт (завдання до роботи, тексти програм, висновки).

Варіант №1

Написати програму нарахування заробітної плати співробітнику фірми в залежності від введеної кількості робочих днів, вибраної із списку посади, наявності пільги (чорнобилець, мати неповнолітніх дітей тощо). Пільга, якщо вона є, вибирається зі списку. В залежності від пільги податок вираховується частково або зовсім не вираховується.

Варіант №2

Дано ділянку 22х33 метрів. Необхідно її засадити деревами. Вид дерева вибирається зі списку, вказується, яку площу відводиться на одне дерево (до 6 м²) та кількість дерев даного виду. Індикатор стану відображає зайняту площу ділянки. У разі перевищення кількості необхідної площі для саджанців, вивести повідомлення, скільки саджанців, що залишились.

Варіант №3

Обчислити вартість путівки в домі відпочинку в залежності від введеної кількості днів відпочинку, вибраного типу номеру (люкс, двомісна, трьохмісна кімната), вибраного типу харчування (без харчування, з повним харчуванням, шведський стіл). Всі дані вибирати зі списків.

Варіант №4

Написати програму, яка підраховує вартість пального у циліндричному баку, якщо відома вартість 1 літра пального. Зі списку вибирається тип пального та об'єм баку.

Варіант №5

Створити програму в якій ви створюєте суміш соків. Суміш може складатись найбільше з 10 видів фруктів (по 100 мл кожного виду соку). Користувач зі списку вибирає вид фрукту та вводить необхідну кількість мілілітрів, щоб в результаті набрати об'єм 1 літр. На індикації повинно відображатися загальний об'єм суміші. В разі доливання соку так, що загальна кількість суміші більше 1 літра, необхідно вивести повідомлення, що така кількість соку перелилася через банку.

Варіант №6

Написати програму-розклад. Зі списку вибирається день тижня, а програма виводить пари цього дня по розкладу.

Варіант №7

За вибраним типом жанру фільму, і часом сеансу висвітлити вартість квитка в кінотеатр. Жанр та час сеансу вибирати зі списку.

Варіант №8

На послуги Internet в залежності від дня тижня та часу доби існують різні тарифи оплати. У вікні програми передбачити вибір режиму роботи в Internet та висвітлення відповідного тарифу.

Варіант №9

Скласти фрагмент контролюючої програми, в якому передбачити послідовне виведення п'яти питань з певними варіантами відповіді. Якщо учень правильно відповідає на питання, то відобразити в компоненті індикації процесу. При неправильній відповіді, програма повинна виводити повідомлення: „Неправильна відповідь!!!!”.

Варіант №10

Розробити програму для розрахунку завантаженості вантажного теплоходу «Зоря» для перевозу автомобілів. В програмі передбачити вибір моделі автомобіля, введення його ваги в кілограмах та вибору похибки цієї ваги. Вивести числове значення завантаженості та відобразити за допомогою ProgressBar, якщо максимальна завантаженість 2000 тон.

Варіант №11

Ввести два числа. Виконати вибрану арифметичну дію з цими числами та висвітлити результат. Передбачити можливість знаходження суми, добутку, віднімання, ділення. Арифметичні операції вибирати зі списку.

ПИТАННЯ ДЛЯ САМОСТІЙНОЇ РОБОТИ

1. Опишіть призначення та основні властивості компонента ListBox?
2. Опишіть призначення та основні властивості компонента ComboBox?
3. Опишіть призначення та основні властивості компонента CheckBox?
4. Опишіть призначення та основні властивості компонента ProgressBar?
5. Опишіть призначення та основні властивості компонента Gauge?
6. Яка різниця між компонентами ListBox та ComboBox?
7. Яка різниця між компонентами ProgressBar та Gauge?
8. На яких закладках інтегрованого середовища Delphi розміщені вищезгадані компоненти?
9. Перечисліть основні методи компонента ListBox?
10. Перечисліть основні методи компонента ComboBox?

ЛІТЕРАТУРА

1. Бадд Т. Объектно-ориентированное программирование в действии / Перев. с англ. – СПб.: Питер, 1997. – 464 с.
2. Бобровский С. Delphi: учебный курс – СПб: Питер, 2001. – 640 с.
3. Буч Г. Объектно-ориентированный анализ и проектирование, 2-е изд./ Пер. с англ. – М.: «Издательство Бином», СПб.: «Невский диалект», 1999. – 560 с.
4. Дантеманн Д., Мишель Д., Тейлор Д. Программирование в среде Delphi. Киев: Диа-Софт Лтд., 1995.
5. Дарахвидзе П., Марков Е. Delphi-среда визуального программирования. СПб.: ВНУ-Санкт-Петербург, 1995.
6. Жуков А. Изучаем Delphi – СПб: Питер, 2000. – 352 с.
7. Конопка Р. Создание оригинальных компонентов в среде Delphi. К.: Диа-Софт Лтд., 1996.
8. Миллер Т., Пауэл Д. Использование Delphi 3. – К.: Диалектика, 1997. – 768 с.
9. Оузьер Д., Гробман С, Батсон С. Delphi 2. Освой самостоятельно: Пер. с англ. – М.: Восточная Книжная Компания, 1997. – 624 с.
10. Соколов В.В. Основи програмування в Delphi // Комп'ютер у школі та сім'ї. – №3, 2001. – С. 21-25.
11. Черняхівський В.В. Delphi-4: сучасна технологія візуального програмування. – Львів: БаК, 1999. – 196 с.

ЗВІТ ДО РОБОТИ

1. Оформити лабораторну роботу в зошиті. Записати тему, мету, відповіді на запитання для самостійної підготовки.
2. Виконати завдання згідно свого варіанту.
3. Зробити висновок.

ЛАБОРАТОРНА РОБОТА №5 ЗАПИСИ У СЕРЕДОВИЩІ DELPHI

Мета роботи: Отримати практичні знання та уміння в роботі з записами середовища Delphi, отримати теоретичні знання з цього питання.

Знати: формат записів, правила роботи з масивами записів, оператор with.

Вміти: використовувати записи для раціоналізації написання програм у середовищі Delphi.

Обладнання: технічне забезпечення: ПЕОМ, програмне забезпечення: інтегроване середовище Delphi.

КОРОТКИ ТЕОРЕТИЧНІ ВІДОМОСТІ

При програмуванні інженерних задач досить часто доводиться мати справу з даними, які природно складаються з інших даних. Наприклад, відомості про студентів містять прізвище, ім'я та по батькові, місяць та рік народження, домашню адресу та інші дані. Для подання подібної інформації мова Object Pascal використовує структуру, що носить ім'я «запис» (record). З одного боку, запис можна розглядати як єдину структуру, а з іншого як набір окремих елементів, компонентів. Характерною особливістю запису є те, що кожен компонент запису має своє ім'я.

Запис це структура даних, яка складається з окремих іменованих компонентів різного типу, що називаються *полями*. На відміну від масиву, компоненти (поля) запису можуть бути різного типу. Для того щоб посилання на той чи інший компонент запису був можливий, поля іменуються.

Структура оголошення гину запису така:

ім'я типу - **record** *список полів* **end**;

Тут: *ім'я типу* - правильний ідентифікатор; **record**, **end** - зарезервовані слова (запис, кінець); *список полів* – список полів, що являє собою послідовність *розділів запису*, між якими ставиться крапка з комою. Кожен розділ запису складається з одного або декількох ідентифікаторів полів, які відділяються один від одного комами. За ідентифікаторами ставиться двокрапка та опис типу полів, наприклад:

```
type  
  student=record  
    prizv:string[15];  
    stat:string 12];  
    dity:string[4];  
    rik,dohod:integer;  
end;  
var  
  m,n: Student ;  
  ...
```

У цьому прикладі тип student (студент) є записом з полями **prizv**, **stat**, **dity**, **rik**, **dohod** (прізвище, стать, діти, рік, доход); змінні m та n містять запис типу **student**.

Так само, як і в масивах, значення змінних типу «запис» можна присвоювати іншим змінним того ж типу, наприклад:

```
r:=m;
```

При зверненні до елемента запису спочатку зазначається ім'я змінної, потім ставиться крапка та назва відповідного елемента.

```
var  
k:student;  
  begin  
    k.prizv:='Іванов';  
    k.stat:='М';  
  ...
```

В разі потреби задати значення великій кількості елементів змінній-запису, кожного разу попередньо указувати її ім'я незручно. Object Pascal має ключове слово **with**, яке дозволяє опускати ім'я змінної у логічному блоці. При цьому перед кожним згадуванням елемента у цьому блоці ім'я змінної буде ставитися автоматично.

```
with k do  
  begin  
    prizv:='Іванов';  
    stat:='М';  
  end;
```

Якщо у логічному блоці, охопленому словом **with**, необхідно використати також змінну, ім'я якої співпадає з назвою одного з елементів запису, то перед нею потрібно вказати назву модуля (ім'я програми), в якому ця змінна описана.

```
var  
k: student;  
name:string;  
  begin  
    name:='Київ';  
    with k do  
      begin  
        name: = 'Динамо';  
        city:=Prjectl.Name;  
      end;
```

Якщо у блоці **with** модуль для конкретної змінної не зазначено, то компілятор перш за все шукає її ім'я у списку назв елементів запису і тільки потім - серед звичайних змінних.

Після ключового слова **with** можна указувати не одне, а декілька імен змінних-записів:

```
with k, t, p do  
  begin
```

...

Object Pascal дозволяє використовувати записи з так званими *варіантними* полями, наприклад:

```
type  
Forma=record  
  Name:=string;  
  case byte of  
    0:(Birthplace:string[50]);  
    1:(Country: string [25];  
      EntryPcrt: string [15];
```



```

EntryDate: 1..31;
ExitDate: 1..31);
end;

```

У цьому прикладі тип **Forma** визначає запис з одним зафіксованим полем **Name** та варіантною частиною, яка додається реченням **case...of**. Варіантна частина складається з декількох варіантів (у прикладі - з двох варіантів: 0 та 1). Кожен варіант визначається константою вибору, після якої стоїть двокрапка та список полів, замкнений у круглі дужки. У будь-якому запису може бути тільки одна варіантна частина; в разі її присутності вона має розташовуватися після усіх фіксованих полів.

Приємною особливістю варіантної частини є та обставина, що усі задані у ній варіанти «накладаються» один на другий, тобто кожному з них виділяється одна й та ж область пам'яті. Це відкриває додаткові можливості перетворення типів, наприклад:

```

var
Mem4: record
  case byte of
    0:(by: array[0..4] of byte);
    1:(wo: array[0..2] of word);
    2:(lo: LongInt);
  end;

```

У цьому прикладі запис **Mem4** має три варіанти, кожен з яких займає у пам'яті одну і ту саму ділянку з 4 байтів. У залежності від того, до якого поля запису ми звергаємося в ОР-програмі, ця ділянка може розглядатися як масив з 4 байтів (поле **by**), масив з двох цілих типу **word** (поле **wo**) та, нарешті, як одне ціле число типу **LongInt** (поле **lo**). Наприклад, цьому запису можна спочатку присвоїти значення довгого цілого, а потім проаналізувати результат за байтами або словами:

```

var
  x: word;
  xb: byte;
  xl: LongInt;
begin
  ...
  with m do
    begin
      lo:=trunk(2*ln(x));
      if wo[1]=0
      then
        if by[1]=0
        then
          xb:=x[0]
        else
          x:=wo[0]
        else
          xl:=lo
      end;
    end;
  ...
end.

```

Речення **case...of**, що відкриває варіантну частину, зовні схоже на відповідний оператор вибору, але практично лише відіграє роль своєрідного службового слова, яке означає початок варіантної частини. Ось тому у кінці варіантної частини не слід ставити **end** як пару до **case...of**. (Оскільки варіантна частина - завжди остання у запису, за нею все ж таки стоїть **end**, але лише як пара до **record**.) Ключ вибору у реченні **case...of** фактично ігнорується компілятором: єдина вимога до нього в Object Pascal полягає у тому, що ключ має визначати деякий стандартний або попередньо оголошений порядковий тип. Імена полів мають бути унікальними у межах того запису, де вони оголошені, але якщо записи містять поля-записи, тобто вкладені одне в одне, імена можуть повторюватися на різних рівнях вкладення.

Оператор (інструкція) **with** дозволяє використовувати імена полів без вказівки імені змінної-запису. У загальному вигляді оператор **with** записують таким чином:

```
with ім'я do  
  begin  
    {оператори ОП-програми}  
  end;
```

Тут: **with** - зарезервоване слово мови Pascal, яке визначає, що далі до слова **end** при зверненні до полів змінної-запису ім'я можна не зазначати ім'я змінної-запису; ім'я - ім'я змінної-запису.

Наприклад, якщо у програмі оголошено змінну-запис

```
student: record      // Інформація про студента  
  name: string [20]; // прізвище  
  name1: string [15]; // ім'я  
  rik: integer;      // рік народження
```

```
end;
```

та дані про студента знаходяться у полях **Edit1**, **Edit2**, **Edit3** діалогового вікна, то для заповнення полів запису замість операторів

```
student.name:=Edit1.Text;  
student.name1:=Edit2.Text;  
student.rik:=Edit3.Text;
```

можна записати:

```
with student do  
  begin  
    name:=Edit1.Text;  
    name1:= Edit2.Text;  
    rik:= Edit3.Text;  
  end;
```

САМОСТІЙНА ПІДГОТОВКА ДО ВИКОНАННЯ ЛАБОРАТОРНОЇ РОБОТИ

4. Записати в зошит тему, мету, обладнання, практичні завдання свого варіанту (вибирається згідно номеру комп'ютера).

5. По літературі до лабораторної роботи вивчити основи роботи з записами в інтегрованому середовищі розробки Delphi.

6. По інструкції до лабораторної роботи ознайомитись з порядком виконання роботи.

ПОРЯДОК ВИКОНАННЯ РОБОТИ

4. Запустити Delphi.

5. Виконати практичні завдання відповідно до свого варіанту.

6. Оформити звіт (завдання до роботи, тексти програм, висновки).

Завдання

Написати програму для опрацювання таблиці «Облік студентів академічної групи» згідно завдання свого варіанту.

Облік студентів академічної групи

Прізвище	Стать	Наявність дітей	Рік народження	Дохід на члена сім'ї, грн.	Оцінка з фізики	Оцінка з математики
Priz	Stat	Dity	Rik	Dohid	OcinkaF	OcinkaM
Байрака	Ч	Немає	1984	500	5	4
Гавриленко	Ж	Є	1989	680	4	5
Кукаріна	Ж	Є	1982	850	5	5
Фесенко	Ч	Немає	1983	600	3	2
Щербак	Ж	Немає	1984	700	2	3
Бондаренко	Ж	Є	1988	630	5	5
Суцев	Ч	Є	1989	650	3	2
Яхонтов	Ч	Немає	1989	810	3	4
Шпак	Ч	Немає	1984	715	4	5
Фоміна	Ж	Є	1988	680	5	4
Зиковий	Ч	Немає	1989	930	2	3
Жиленко	Ч	Немає	1990	658	4	2
Довбун	Ж	Є	1990	658	3	4
Панленко	Ж	Немає	1989	695	2	3
Порохня	Ч	Є	1989	755	4	2

Варіант 1

Визначити кількість студентів, які мають дітей та які народилися у 1978 році. Надрукувати їх прізвища, наявність дітей та рік народження

Варіант 2

Визначити кількість студентів, молодших за 1980 рік, та підрахувати, скільки з них мають дітей. Надрукувати їх прізвища та рік народження

Варіант 3

Визначити кількість студентів жіночої статі, які мають дітей та чий дохід більший за 250 грн. Надрукувати їх прізвища, наявність дітей та дохід

Варіант 4

Студентам заплатили стипендію: тим, які мають дітей,- по 210 грн., тим, які не мають дітей, по 150 грн. Скільки усього грошей отримали студенти?

Варіант 5

Підрахувати середній бал успішності в фізики та математики студентів:
а) які мають дітей; б) які не мають дітей

Варіант 6

Підрахувати середній бал успішності з фізики та математики студентів:
а) чоловічої статі; б) жіночої статі

Варіант 7

Підрахувати кількість студентів, які мають дохід: а) більший за 262 грн.; б) менший за 217 грн. Надрукувати їх прізвища

Варіант 8

Підрахувати кількість студентів, які отримали з фізики або математики «2» та мають дітей, надрукувати їх прізвища

Варіант 9

Підрахувати кількість студентів, віком старших за 1988 рік народження, та визначити, скільки з них чоловічої статі

Варіант 10

Підрахувати кількість студентів чоловічої статі та визначити, скільки з них не мають дітей

Варіант 11

Підрахувати кількість студентів жіночої статі, які мають дохід, більший за 190 грн.

ПИТАННЯ ДЛЯ САМОСТІЙНОЇ РОБОТИ

1. Поясніть, як з'явилася конструкція «запис»?
2. Що є характерною особливістю запису?
3. Що це - поля запису?
4. Наведіть структуру оголошення типу запису.
5. Що означають зарезервовані слова **record, end**?
6. Як можна звернутися до елемента запису?
7. Що означає ключове слово **with**? Як воно використовується?
8. Як виконується ввід та вивід записів у файл?

ЛАБОРАТОРНА РОБОТА №6 СТВОРЕННЯ ГРАФІЧНИХ ЗОБРАЖЕНЬ

Мета роботи: Ознайомитись з компонентами Image, Canvas, Timer, поняттям “бітовий образ”. Навчитися створювати програми, які опрацьовують будують графічні зображення, динамічні об’єкти, виводять ілюстрації і т.п.

Знати: основні властивості розглядаємих компонентів.

Вміти: використовувати компоненти Image, Canvas, Timer в створенні власних програм.

Обладнання: технічне забезпечення: ПЕОМ, програмне забезпечення: інтегроване середовище Delphi.

КОРОТКИ ТЕОРЕТИЧНІ ВІДОМОСТІ

Canvas – це поверхня (форми або компонента Image), на якій відповідні методи можуть викреслювати графічні примітиви. Вид графічних елементів визначають властивості поверхні, на якій ці елементи викреслюються.

TextOut(x,y,s) виводить рядок s від точки з координатами (x, y). Шрифт визначає властивість Font поверхні (canvas), на яку виводиться текст, колір зафарбування області виводу тексту залежить від властивості Brush цієї ж поверхні

Draw(x,y,b) виводить від точки з координатами (x, y) бітовий образ b. Якщо значення властивості Transparent поверхні, на яку виконується висновок, дорівнює True, то точки, колір яких збігаються з кольором лівої нижньої точки бітового образу, не відображаються

LineTo(x,y) викреслює лінію з поточної точки в точку із зазначеними координатами. Вид лінії визначає властивість Pen.

MoveTo(x,y) переміщає покажчик поточної точки в точку із зазначеними координатами.

PolyLine(pl) викреслює ламану лінію. Координати точок перегину задає параметр pl – масив структур типу TPoint. Якщо перший і останній елементи масиву однакові, то буде накреслений замкнутий контур. Вид лінії визначає властивість Pen.

Polygon(pl) викреслює й зафарбовує багатокутник. Координати кутів задає параметр pl – масив структур типу TPoint. Перший і останній елементи масиву повинні бути однакові. Вид границі визначає властивість Pen, колір і стиль зафарбування внутрішньої області – властивість Brush.

Ellipse(x1,y1, x2,y2) викреслює еліпс, коло. Параметри x1, y1, x2 і y2 задають розмір прямокутника, у який вписується еліпс. Вид лінії визначає властивість Pen.

Arc(x1,y1,x2,y2,x3,y3,x4,y4) викреслює дугу. Параметри x1, y1, x2, y2 визначають еліпс, з якого вирізує дуга, параметри x3, y3 та x4, y4 – координати кінців дуги. Дуга викреслюється проти годинникової стрілки від

точки (x3,y3) до точки (x4,y4). Вид лінії (границі) визначає властивість Pen, колір і спосіб зафарбування внутрішньої області – властивість Brush.

Rectangle(x1,y1,x2,y2) викреслює прямокутник. Вид лінії визначає властивість Pen, колір і спосіб зафарбування внутрішньої області – властивість Brush

RoundRec(x1,y1,x2,y2,x3,y3) викреслює прямокутник з округленими кутами. Параметри x1,y1,x2,y2 задають координати лівих верхніх і правого нижнього кутів, x3 та y3 – радіус округлення. Вид лінії визначає властивість Pen, колір і спосіб зафарбування внутрішньої області – властивість Brush.

Властивості об'єкта Canvas

Transparent ознака використання "прозорого" кольору при виводі бітового образу методом Draw. Якщо значення властивості дорівнює True, то точки, колір яких збігаються з кольором лівої нижньої точки бітового образу, не відображаються.

Pen властивість Pen визначає: колір, товщину та стиль ліній, що викреслюються методами виводу графічних примітивів.

Brush властивість Brush являє собою об'єкт, що уточнює властивості: колір та стиль зафарбування областей, що викреслюються методами виводу графічних примітивів.

Font властивість визначає шрифт (назва, розмір, колір, спосіб оформлення), що використовується для виводу на поверхню полотна тексту.

Властивості об'єкта Pen

Color колір лінії (clBlack – чорний; clMaroon – каштановий; clGreen – зелений; clOlive – оливковий; clNavy – темно-синій; clPurple – рожевий; clTeal – зелено-блакитний; clGray – сірий; clSilver – срібlistий; clRed – червоний; clLime – салатний; clBlue – синій; clFuchsia – яскраво-рожевий; clAqua – бірюзовий; clWhite – білий).

Style вид лінії. Лінія може бути: psSolid – суцільна; psDash – пунктирна (довгі штрихи); psDot – пунктирна (короткі штрихи); psDashDot – пунктирна (чергування довгого й короткого штрихів); psDashDotDot – пунктирна (чергування одного довгого й двох коротких штрихів); psClear – не відображається (використається, якщо не треба зображувати границю, наприклад, прямокутника).

Width товщина лінії, задається в пікселях. Товщина пунктирної лінії не може бути більше 1.

Властивості об'єкта Brush

Color колір зафарбовування замкнутої області.

Style стиль (тип) заповнення області (bsSolid – суцільна заливка; bsClear – область не зафарбовується; bsHorizontal – горизонтальне штрихування; bsVertical – вертикальне штрихування; bsFDiagonal – діагональне штрихування з нахилом ліній уперед; bsBDiagonal – діагональне штрихування з нахилом ліній назад; bsCross – горизонтально-вертикальне штрихування, у клітину; bsDiagCross – діагональне штрихування, у клітину)

САМОСТІЙНА ПІДГОТОВКА ДО ВИКОНАННЯ ЛАБОРАТОРНОЇ РОБОТИ

1. Записати в зошит тему, мету, обладнання, практичні завдання свого варіанту (вибирається згідно номеру комп'ютера).
2. По літературі до лабораторної роботи вивчити основи роботи з компонентами Image, Canvas, Timer, поняттям “бітовий образ” в інтегрованому середовищі розробки Delphi.
3. По інструкції до лабораторної роботи ознайомитись з порядком виконання роботи.

ПОРЯДОК ВИКОНАННЯ РОБОТИ

1. Запустити Delphi.
2. Виконати практичні завдання відповідно до свого варіанту.
3. Оформити звіт (завдання до роботи, тексти програм, висновки).

Варіант №1

1. Створити програму, яка виводить на екран малюнок 2.
2. З використанням чередування відеосторінок скласти програму виведення на екран образу крокуючого віслучка.

Варіант №2

1. Створити програму, яка зображує на екрані декартову систему координат, початок якої збігається з центром екрана.
2. З використанням чередування відеосторінок скласти програму виведення на екран образу парашута, який розкривається у повітрі.

Варіант №3

1. Створити програму, яка виводить на екран малюнок 3.
2. З використанням чередування відеосторінок скласти програму виведення на екран образу плигаючого м'яча, який деформується при ударі.

Варіант №4

1. Створити програму, яка на формі по діагоналі зображує три прямокутники. В першому прямокутнику 10 різнокольорових кіл (центри кіл знаходяться у центрі прямокутника), другий і третій прямокутники заповнити різнокольоровими прямими та колами відповідно (колір, кількість та координати визначаються випадково).
2. З використанням чередування відеосторінок скласти програму виведення на екран образу колеса зі спицями.

Варіант №5

1. Створити програму, яка виводить на екран вигляд мобільного телефону.
2. З використанням чередування відеосторінок скласти програму виведення на екран образу гриба, що росте.

Варіант №6

1. Створити програму, яка виводить на екран малюнок 1.
2. З використанням чередування відеосторінок скласти програму виведення на екран образу квітки, що росте та розпускається.

Варіант №7

1. Створити програму, яка виводить на екран малюнок 5.

2. З використанням чередування відеосторінок склати програму виведення на екран образу веслувальника в лодці.

Варіант №8

1. Створити програму, яка виводить на екран зображення монітора із заставкою.

2. З використанням чередування відеосторінок склати програму виведення на екран образу летящої пташки, яка взмахає крилами.

Варіант №9

1. Створити програму, яка виводить на екран малюнок 4.

2. З використанням чередування відеосторінок склати програму виведення на екран образу наближаючого космічного об'єкту.

Варіант №10

1. Створити програму, яка виводить на екран малюнок 6.

2. З використанням чередування відеосторінок склати програму виведення на екран образу автомобіля, що рухається.

Варіант №11

1. Створити програму, яка виводить на екран вигляд комп'ютера з заставкою на моніторі "зоряне небо".

2. З використанням чередування відеосторінок склати програму виведення на екран образу наближаючих гір різного кольору (гори збільшуються у розмірі)

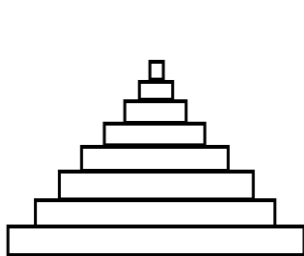


Рис. 1.

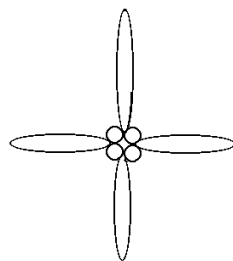


Рис. 2.



Рис. 3.

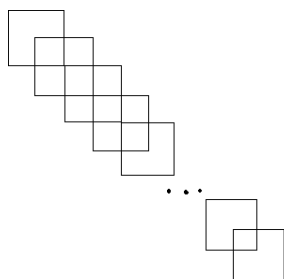


Рис. 4.



Рис. 5.

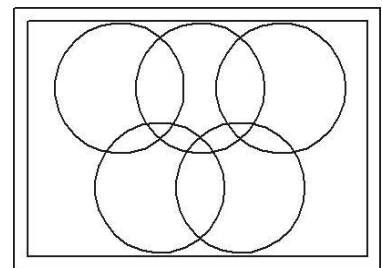


Рис.6

ПИТАННЯ ДЛЯ САМОСТІЙНОЇ РОБОТИ

1. Перечисліть основні властивості Brush об'єкта Canvas?
2. Перечисліть основні властивості Pen об'єкта Canvas?
3. Перечисліть основні властивості Font об'єкта Canvas?

4. Опишіть призначення об'єкта Canvas?
5. Опишіть призначення та формат функцій: TextOut, Draw, LineTo?
6. Опишіть призначення та формат функцій: MoveTo, PolyLine, Polygon?
7. Опишіть призначення та формат функцій: Ellipse, Arc та Rectangle?
8. Опишіть призначення та формат функцій: Rectangle, RoundRec?
9. Що таке бітовий образ?
10. Як можна реалізувати переміщення об'єкта на формі?

ЛІТЕРАТУРА

1. Бадд Т. Объектно-ориентированное программирование в действии / Перев. с англ. – СПб.: Питер, 1997. – 464 с.
2. Бобровский С. Delphi: учебный курс – СПб: Питер, 2001. – 640 с.
3. Буч Г. Объектно-ориентированный анализ и проектирование, 2-е изд./ Пер. с англ. – М.: «Издательство Бином», СПб.: «Невский диалект», 1999. – 560 с.
4. Дантемманн Д., Мишель Д., Тейлор Д. Программирование в среде Delphi. Киев: Диа-Софт Лтд., 1995.
5. Дарахвидзе П., Марков Е. Delphi-среда визуального программирования. СПб.: ВНУ-Санкт-Петербург, 1995.
6. Жуков А. Изучаем Delphi – СПб: Питер, 2000. – 352 с.
7. Конопка Р. Создание оригинальных компонентов в среде Delphi. К.: Диа-Софт Лтд., 1996.
8. Миллер Т., Пауэл Д. Использование Delphi 3. – К.: Диалектика, 1997. – 768 с.
9. Оузьер Д., Гробман С, Батсон С. Delphi 2. Освой самостоятельно: Пер. с англ. – М.: Восточная Книжная Компания, 1997. – 624 с.
10. Соколов В.В. Основы програмування в Delphi // Комп'ютер у школі та сім'ї. – №3, 2001. – С. 21-25.
11. Черняхівський В.В. Delphi-4: сучасна технологія візуального програмування. – Львів: БаК, 1999. – 196 с.

ЗВІТ ДО РОБОТИ

1. Оформити лабораторну роботу в зошиті. Записати тему, мету, відповіді на запитання для самостійної підготовки.
2. Виконати завдання згідно свого варіанту.
3. Зробити висновок.

ЛАБОРАТОРНА РОБОТА №7 ОБРОБКА ОДНОВИМІРНИХ МАСИВІВ

Мета роботи: Придбання практичних знань та умінь розв'язувати інженерні задачі з одновимірними масивами.

Знати: поняття масиву, його структуру та правила роботи з одновимірними масивами.

Вміти: створювати програми у інтегрованому середовищі Delphi на опрацювання одновимірних масивів.

Обладнання: технічне забезпечення: ПЕОМ, програмне забезпечення: інтегроване середовище Delphi.

КОРОТКИ ТЕОРЕТИЧНІ ВІДОМОСТІ

З поняттям масиву доводиться зустрічатися при розв'язати задач обробки сукупності простих значень. Найпростішим прикладом масиву є **лінійна** таблиця. Значення, що утворюють лінійну таблицю, є елементами масиву, а їх рядкові номери в таблиці називаються *індексами*. Взагалі *масив* - це упорядкована сукупність значень однакового тішу. Наприклад, в ОР-програмі можна описувати та обробляти масиви цілих чисел, логічних та символічних значень. Прості значення, які утворюють масив, упорядковані таким чином, що кожному простому значенню відповідає сукупність номерів, які визначають його місцеположення в загальній послідовності (номер називається індексом).

Формат опису:

var

ім'я: array [тип індексів] of тип елементів;

Тип індексів може бути будь-яким скалярним (частіше цілим типом, відрізком цілого типу, переліковним типом), крім типу *real*. Тип елементів масиву може бути будь-яким. Наприклад:

var

M: array[1..8] of real;

Тут масив *M* складається з восьми елементів (*M1, M2, ..., M8*) дійсного типу.

Якщо маємо масив *bal*, що утримує середній бал успішності студентів Іванова, Соколова, Фесенко, а також масиви *mas1* та *mas2*, що утримують оцінки з математики двох груп студентів з 30 осіб, то один з варіантів опису цих масивів має вигляд:

type

fam = (Ivanov, Sckolov, Fesenkc);

var

bal: **array**[fam] **of** real;

mas1, mas2: **array**[1..30] **of** integer;

Вибір окремих компонентів одновимірного масиву здійснюється через вказівку ідентифікатора масиву, за яким у квадратних дужках йде індексний вираз. Індексний вираз має давати значення, які знаходяться в діапазоні,

визначеному типом індексу. Для оголошеного вище масиву *m* в ОР-програмі доступні такі індексні змінні: *M[1]*, *M[2]*, ..., *M[8]*.

Одновимірний масив відповідає поняттю вектора (лінійної таблиці, рядка), двовимірний масив - поняттю матриці (набору векторів, прямокутній таблиці), тривимірний масив поняттю масиву матриць (комплекту матриць, набору матриць).

Багатовимірний масив зображає чисту абстракцію, оскільки пам'ять ЕОМ одновимірна, а багатовимірні масиви зберігаються в пам'яті ЕОМ у вигляді лінійних послідовних значень.

Розглянемо матрицю *a*, яка складається з 2 x 3 елементів:

$$\begin{Bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{Bmatrix}$$

Елементи цієї матриці можуть бути розміщені в пам'яті комп'ютера «рядками», тобто у послідовності: *a₁₁*, *a₁₂*, *a₁₃*, *a₂₁*, *a₂₂*, *a₂₃*. Іншим засобом розташування елементів є організація масиву в пам'яті «колонками», що приводить до послідовності: *a₁₁*, *a₂₁*, *a₁₂*, *a₂₂*, *a₁₃*, *a₂₃*

Крім явного опису масивів у розділі змінних, має іншу форму опису, яка складається з двох етапів. Спочатку в розділі опису типів **type** зазначається тип масиву. Потім у розділі опису змінних **var** переліковуються масиви, що відносяться до зазначеного типу.

Введення типу масиву збільшує розділ опису, але водночас спрощує налагодження програми та утримує від абсурдних помилок.

Оголошення типу масиву та змінних цього типу має вигляд:

type

ім'я типу = **array** [*t1*] of [*t2*];

var

ім'я масиву: *ім'я типу*;

Тут *t1* - тип індексу, *t2* - базовий тип елементів масиву. Наприклад, нехай у програмі використовується масив *r*, який складається з 10 елементів дійсного типу. Позначимо тип масиву іменем *mas*. Тоді опис масиву можна виконати так:

type

mas = **array**[1..10] of real;

var

r:*mas*;

Якщо програма має декілька масивів, наприклад, *r*, *a*, *b*, *c*, що мають тип *mas*, тоді зміниться тільки розділ опису змінних:

var

r,a,b,c:*mas*;

Слід зазначити, що масиви *r*, *a*, *b*, *c* використовуються у розділі інструкцій (операторів) програми. Тип масиву *mas* введено формально тільки у розділі опису, але ніде в програмі він не зазначається та не обробляється.

Ввід одновимірного масиву

Ввести масив означає отримати від користувача (або з файлу) під час роботи програми значення елементів масиву. Найпростіше вирішити питання

вводу елементів масиву для кожного елемента масиву можна шляхом утворення поля вводу (компоненти Edit1, ListBox1). Але якщо необхідно ввести досить великий масив, то таке вирішення неприпустиме.

Послідовність чисел зручно вводити у поле або у таблицю, де кожне число розміщується на окремому рядку або числа розташовуються на одному рядку, відокремлені один від одного пропусками.

Розглянемо ввід одновимірного масиву з використанням компонента форми Edit (поле редагування). Діалогове вікно під час вводу елементів масиву з використанням компонента Edit має такий вигляд.



При натисканні на кнопку «Ввід» програма повинна виділити з вікна вводу (компонент **Edit1**) перший елемент (0.1), перетворити його на число та присвоїти отримане значення першому елементу масиву. Потім виділити другий елемент (2.3), перетворити його на число та присвоїти отримане значення другому елементу масиву, а потім, аналогічно,- для кожного з елементів масиву (48,5.6,11.4,6.619,1.73).

Головний цикл процедури вводу одновимірного масиву має вигляд:

```
for i:=1 to 7 do
    k[i]:=StrToFloat (FK(Edit1.Text, i));
```

Тут: k - масив;

Edit1 - ім'я поля вводу-редагування;

FK - ім'я функції користувача.

Функція FK виділяє з рядка перший елемент (до першого пропуску). Елемент з масиву в полі **Edit1** - це послідовність символів, розташована між двома пропусками.

```
function FK(k:string;П:integer):string;
var
    p,i:integer;
begin
    for i:=1 to n-1 do
        begin
            p:=pos(' ', k);
            k:=copy(k, p+1, length(k)-p);
        end;
    p:=pos(' ', k); //Позиція пропуску у рядку масива
    if p<>0
    then
        result:=copy(k,1,p-1)
    else
        result:=k
    end;
```

Для того, щоб виділити n -ий елемент масиву, функція FK спочатку вилучає (у циклі **for**) попередні $n-1$ елементи, потім знаходить пропуск, який позначає кінець елемента, виділяє елемент та повертає його як значення функції.

Функція сору виділяє фрагмент рядка: сору ($k, 1, p-1$). 1 - номер першого символу; $p-1$ - довжина елемента. Наприклад:

```
k:='студент групи';
```

```
m:=сору(k,9,5);
```

Значенням m буде рядок ' групи'.

Функція *length* обчислює довжину рядка.

```
t:=length('студент'); (значення t-7).
```

Функція pos (' ', k) обчислює позицію пропуску в символному рядку k .

Наприклад:

```
k:='студент групи';
```

```
r:=pos(' ', k);
```

Результат: r=8.

Вивід одновимірного масиву

Під виводом масиву слід розуміти вивід на екран дисплея у діалогове вікно значень елементів масиву. Якщо в ОР-програмі необхідно вивести значення всіх елементів масиву, то для цього можна використати оператор **for**, а змінна лічильника оператора **for** може бути використаною як індекс елемента масиву.

Нехай необхідно надрукувати прізвища семи студентів групи із масиву з іменем den. Наведемо приклад такої процедури:

```
procedure TForm1.Button1Click(Sender: TObject);
```

```
const
```

```
    grup=7;    {Кількість студентів}
```

```
type
```

```
    mas=array [1..grup] of string[10];
```

```
var
```

```
    den:mas;
```

```
    a:string;
```

```
    i:integer;
```

```
begin
```

```
den[1]:='Кукаріна';
```

```
den[2]:='Фесенко';
```

```
den[3]:='Яхонтов';
```

```
den[4]:='Фоміна';
```

```
den[5]:='Павленко';
```

```
den[6]:='Гавриленко';
```

```
den [7]:='Байрака';
```

```
    for i:=1 to grup do
```

```
        a:=a+IntToStr(i)+' '+den[i]+#13;
```

```
        {Вивід списку студентів}
```

```
        ShowMessage(a);
```

```
end;
```

Ця процедура виводить у вікно повідомлення перенумерований список студентів групи den.

Ввід одновимірного масиву з використанням компонента Memo

Якщо необхідно ввести символний масив, причому кожен елемент масиву може складатися з декількох слів, то замість компонента **Edit** зручніше користуватися компонентом Memo, який дозволяє вводити декілька рядків тексту.

Головний цикл процедури вводу символного масиву з **Memo** має такий вигляд:

```
for i:=1 to size do  
a[i ] :=Memo1.Lines [ i ] ;
```

САМОСТІЙНА ПІДГОТОВКА ДО ВИКОНАННЯ ЛАБОРАТОРНОЇ РОБОТИ

1. Записати в зошит тему, мету, обладнання, практичні завдання свого варіанту (вибирається згідно номеру комп'ютера).

2. По літературі до лабораторної роботи вивчити основи роботи з компонентами Image, Canvas, Timer, поняттям “бітовий образ” в інтегрованому середовищі розробки Delphi.

3. По інструкції до лабораторної роботи ознайомитись з порядком виконання роботи.

ПОРЯДОК ВИКОНАННЯ РОБОТИ

1. Запустити Delphi.
2. Виконати практичні завдання відповідно до свого варіанту.
3. Оформити звіт (завдання до роботи, тексти програм, висновки).

Завдання

Для **сукупності** чисел $x=(12.37, 16.55, 2.43, 0.81, 0.32, 1.345, -12.18, 0.15, 4.314, 1.29, 54.38, 0.214, 3.837, 0.918, 75.014)$ та $a=3.14, b=7.8$ необхідно розв'язати задачу, умова якої відповідає Вашому варіанту.

Варіант №1

Обчислити добуток елементів масиву та надрукувати номери елементів масиву, які не дорівнюють нулю та більші за a .

Варіант №2

Знайти найменший елемент масиву та його номер серед чисел, більших за b та менших за a

Варіант №3

Знайти суму елементів масиву, більших за b , та надрукувати номери цих елементів

Варіант №4

Знайти найбільший елемент масиву та його номер серед чисел, більших за b та менших за число 50.2

Варіант №5

Знайти найменший елемент та його номер серед чисел, більших за число 1.29 та менших за число 16.55

Варіант №6

Обчислити суму та надрукувати номери елементів, більших за b , більших за a , з парними номерами

Варіант №7

Обчислити добуток елементів масиву, більших за b , та їх кількість

Варіант №8

Обчислити добуток та надрукувати номери елементів, не рівних нулю, не рівних 54.38 та більших за a

Варіант №9

Обчислити добуток та кількість елементів, менших за b , з парними номерами

Варіант №10

Обчислити суму, кількість додатніх елементів, менших за a , та надрукувати номери цих елементів

Варіант №11

Обчислити суму та кількість від'ємних елементів, які більші за b та менші за a

ПИТАННЯ ДЛЯ САМОСТІЙНОЇ РОБОТИ

1. Які типи даних відносяться до складних?
2. Які змінні називаються структурованими?
3. Що таке масив? Наведіть приклади.
4. Наведіть приклади структури масиву.
5. Що таке індексна змінна масиву?
6. Як та в яких розділах описується масив?
7. Як ввести одновимірний масив з використанням компонента Edit?
8. Як вивести одновимірний масив на екран дисплея?
9. Що означає цей фрагмент?

for i:=1 to n **do**

a[i]:=Memo1.Lines[i-1];

10. Як Ви розумієте цей запис? $n := Memo1.Lines.Count;$

ЛІТЕРАТУРА

12. Бадд Т. Объектно-ориентированное программирование в действии / Перев. с англ. – СПб.: Питер, 1997. – 464 с.
13. Бобровский С. Delphi: учебный курс – СПб.: Питер, 2001. – 640 с.
14. Буч Г. Объектно-ориентированный анализ и проектирование, 2-е изд./ Пер. с англ. – М.: «Издательство Бином», СПб.: «Невский диалект», 1999. – 560 с.
15. Дантеманн Д., Мишель Д., Тейлор Д. Программирование в среде Delphi.

- Київ: Діа-Софт Лтд., 1995.
16. Дарахвідзе П., Марков Е. Delphi-середовище візуального програмування. СПб.: ВНУ-Санкт-Петербург, 1995.
 17. Жуков А. Изучаем Delphi – СПб: Питер, 2000. – 352 с.
 18. Конопка Р. Создание оригинальных компонентов в среде Delphi. К.: Діа-Софт Лтд., 1996.
 19. Миллер Т., Пауэл Д. Использование Delphi 3. – К.: Діалектика, 1997. – 768 с.
 20. Оузер Д., Гробман С, Батсон С. Delphi 2. Освой самостійно: Пер. с англ. – М.: Восточная Книжная Компания, 1997. – 624 с.
 21. Соколов В.В. Основи програмування в Delphi // Комп'ютер у школі та сім'ї. – №3, 2001. – С. 21-25.
 22. Черняхівський В.В. Delphi-4: сучасна технологія візуального програмування. – Львів: БаК, 1999. – 196 с.

ЗВІТ ДО РОБОТИ

4. Оформити лабораторну роботу в зошиті. Записати тему, мету, відповіді на запитання для самостійної підготовки.
5. Виконати завдання згідно свого варіанту.
6. Зробити висновок.

ЛАБОРАТОРНА РОБОТА №8 РЯДКОВІ ТАБЛИЦІ

Мета роботи: закріпити теоретичний матеріал та придбати практичні навички роботи та уміння розв'язувати інженерно-педагогічні задачі, що містять двовимірні масиви.

Знати: поняття та правила роботи з двовимірними масивами, основні властивості компонента `StringGrid`.

Вміти: використовувати компонент `StringGrid` при написанні власних програм для опрацювання двовимірних масивів.

Обладнання: технічне забезпечення: ПЕОМ, програмне забезпечення: інтегроване середовище Delphi.

КОРОТКИ ТЕОРЕТИЧНІ ВІДОМОСТІ

Часто в програмах потрібні сітки для введення даних, аналогічні до електронних таблиць.

У Delphi вам доступні відразу два види сіток - ***TStringGrid*** і ***TDrawGrid***. Різниця в них незначна. У `TStringGrid` ви можете вводити дані, і вони там будуть зберігатися і відображатися, а в `TDrawGrid` дані можуть вводитися, але за відображення їх має відповідати ваш код. Іншими словами, `TStringGrid` - це сітка рядків, а `TDrawGrid` - це сітка малюнків. Розглянемо роботу тільки з `TStringGrid`, тому що він більш поширений.

Таблиця ділиться на дві частини – фіксовану та робочу. Фіксована частина слугує для відображення заголовків колонок та рядків, а також для ручного управління їх розмірами. За замовчуванням фіксована частина займає ліву колонку та верхній ряд таблиці, однак за допомогою властивостей `FixedCols` та `FixedRows` можна задати іншу кількість фіксованих колонок та рядків (якщо дані властивості дорівнюють 0, то таблиця не містить фіксованої зони). Робоча зона містить довільну кількість рядків та стовпчиків. Якщо робоча частина не поміщається у розмір компонента, то автоматично з'являються полоси прокрутки.

ВЛАСТИВОСТІ TStringGrid

BorderStyle - стиль обрамлення. Тут можливі варіанти `bssingle` або `bsNone`.

Cells – визначає вміст комірки з табличними координатами. Це двомірний масив комірок, кожна з яких може містити довільний текст. Конкретна комірка визначається парою чисел – номером колонки та номером рядка, на перетині яких вона знаходиться (нумерація починається з 0).

Col – містить номер колонки з коміркою, в яку відбувається введення.

Row - містить номер стовпця з коміркою, в яку відбувається введення

ColCount – містить кількість колонок таблиці.

`RowCount` – містить кількість стовпців таблиці.

DefaultColWidth – містить значення ширини колонок.

DefaultRowHeight – містить значення висоти рядків.

EditorMode –

FixerColor – визначає колір фіксованої зони.

FixedCols – визначає кількість колонок фіксованої зони.

FixedRows – визначає кількість рядів фіксованої зони.

GridHeight – містить значення висоти таблиці.

GridLineWidth – визначає товщину ліній, які розчерчують таблицю.

GrieWidth – містить значення ширини таблиці.

Options – містить параметри таблиці. Сюди відносяться:

goFixedVertLine - малювати вертикальні лінії сітки у фіксованих комірках;

goFixedHorzLine - малювати горизонтальні лінії сітки у фіксованих комірках;

goVertLine - малювати вертикальні лінії сітки у нефіксованих комірках;

goHorzLine - малювати горизонтальні лінії сітки у нефіксованих комірках;

goRangeSelect - дозволяти виділяти кілька комірок;

goDrawFocusSelected - малювати фокус виділеної комірки;

goRowsizing - чи можна змінювати розмір рядків перетягуванням мишки;

goCoisizing - чи можна змінювати розмір стовпців перетягуванням мишки;

goRowMoving - чи можна переміщувати рядки (якщо true, то можна натиснути кнопку миші, встановивши її покажчик на фіксовану комірку рядка, і перетягнути в нове положення);

goColMoving - чи можна переміщувати стовпці (якщо true, то можна натиснути кнопку миші, встановивши її покажчик на фіксовану комірку колонки, і перетягнути її в нове положення);

goEditing - чи можна вводити з клавіатури дані у сітку.

САМОСТІЙНА ПІДГОТОВКА ДО ВИКОНАННЯ ЛАБОРАТОРНОЇ РОБОТИ

4. Записати в зошит тему, мету, обладнання, практичні завдання свого варіанту (вибирається згідно номеру комп'ютера).

5. По літературі до лабораторної роботи вивчити основи роботи з двовимірними масивами та компонентом TStringGrid в інтегрованому середовищі розробки Delphi.

6. По інструкції до лабораторної роботи ознайомитись з порядком виконання роботи.

ПОРЯДОК ВИКОНАННЯ РОБОТИ

4. Запустити Delphi.

5. Виконати практичні завдання відповідно до свого варіанту.

6. Оформити звіт (завдання до роботи, тексти програм, висновки).

Завдання

Напишіть програму, яка розв'язує задачу, відповідну Вашому варіанту. Назву і розмір початкового двовимірного масиву необхідно брати з таблиці 1, в якій:

M – кількість елементів у початковому масиві;

N – кількість колонок у початковому масиві;

I – початковий номер рядка для вибору значень елементів з таблиці 1.

J – початковий номер колонки для вибору значень елементів з таблиці 1.

Числові початкові дані двовимірного масиву визначаються у таблиці 2.

Таблиця 1.

Варіант	Назва	M	N	I	J
1	Z	6	4	1	3
2	H	5	3	2	1
3	A	3	5	1	3
4	P	5	4	2	3
5	F	2	5	2	3
6	Y	3	4	4	2
7	R	5	4	2	7
8	H	4	5	1	4
9	A	3	5	1	3
10	D	4	5	3	1
11	A	3	4	2	1

Таблиця 2.

Значення елементів двовимірного масиву

Номери рядків	Номери колонок									
	1	2	3	4	5	6	7	8	9	10
1	0.25	2.18	-1.35	2.34	-5.3	-2.4	-3.2	0.19	-2.8	0
2	-5.9	1.26	3.08	-1.34	2.25	-6.2	12.5	8.4	1.25	3.24
3	-5.14	-3.24	-1.13	2.31	0	2.4	0	1.32	0	1.39
4	3.17	-2.3	-1.15	-3.2	1.45	-1.3	2.4	0	1.7	0
5	0	-2.7	0	2.6	-3.54	11.3	0	14.8	0.82	12.21
6	3.1	4.52	11.8	14.5	0	0.2	1.35	-0.52	17	-2.4

Варіант №1

Для кожної колонки двовимірного масиву Z обчислити добуток елементів, що не дорівнюють нулю.

Варіант №2

Для кожної колонки двовимірного масиву H визначити номери рядків, що мають від'ємні елементи.

Варіант №3

Обчислити суму позитивних елементів двовимірного масиву A .

Варіант №4

Знайти найменший елемент та його індекси двовимірного масиву P .

Варіант №5

Обчислити суму елементів непарних рядків для кожної колонки двовимірного масиву F .

Варіант №6

Обчислити суму від'ємних елементів кожної колонки двовимірного масиву Y .

Варіант №7

Обчислити суму від'ємних елементів двовимірного масиву R .

Варіант №8

Знайти найбільший елемент та його індекси у двовимірному масиві H .

Варіант №9

Обчислити суму позитивних елементів двовимірного масиву A .

Варіант №10

Для кожного рядка двовимірного масиву D обчислити суму елементів.

Варіант №11

Обчислити суму позитивних елементів кожного рядка двовимірного масиву A .

ПИТАННЯ ДЛЯ САМОСТІЙНОЇ РОБОТИ

1. Що таке динамічні масиви?
2. як оголошуються динамічні масиви?
3. Яке призначення компонента StringGrid?
4. Розкажіть про параметри, які можна встановити у властивості Options компонента StringGrid?
5. Поясніть, для чого існують властивості RowCount та ColCount компонента StringGrid?
6. Які помилки найбільш поширені при роботі з масивами?

ЛІТЕРАТУРА

23. Бадд Т. Объектно-ориентированное программирование в действии / Перев. с англ. – СПб.: Питер, 1997. – 464 с.
24. Бобровский С. Delphi: учебный курс – СПб.: Питер, 2001. – 640 с.
25. Буч Г. Объектно-ориентированный анализ и проектирование, 2-е изд./ Пер. с англ. – М.: «Издательство Бином», СПб.: «Невский диалект», 1999. – 560 с.
26. Дантемманн Д., Мишель Д., Тейлор Д. Программирование в среде Delphi.

- Київ: Діа-Софт Лтд., 1995.
27. Дарахвідзе П., Марков Е. Delphi-середовище візуального програмування. СПб.: ВНУ-Санкт-Петербург, 1995.
 28. Жуков А. Изучаем Delphi – СПб: Питер, 2000. – 352 с.
 29. Конопка Р. Создание оригинальных компонентов в среде Delphi. К.: Діа-Софт Лтд., 1996.
 30. Миллер Т., Пауэл Д. Использование Delphi 3. – К.: Діалектика, 1997. – 768 с.
 31. Оузер Д., Гробман С, Батсон С. Delphi 2. Освой самостійно: Пер. с англ. – М.: Восточная Книжная Компания, 1997. – 624 с.
 32. Соколов В.В. Основи програмування в Delphi // Комп'ютер у школі та сім'ї. – №3, 2001. – С. 21-25.
 33. Черняхівський В.В. Delphi-4: сучасна технологія візуального програмування. – Львів: БаК, 1999. – 196 с.

ЗВІТ ДО РОБОТИ

7. Оформити лабораторну роботу в зошиті. Записати тему, мету, відповіді на запитання для самостійної підготовки.
8. Виконати завдання згідно свого варіанту.
9. Зробити висновок.

ЛАБОРАТОРНА РОБОТА №9 РОБОТА З ТАБЛИЧНОЮ ІНФОРМАЦІЄЮ

Мета роботи: Ознайомитись з компонентами Chartfx, F1Book, VtChart. Навчитися створювати програми, які опрацьовують табличні дані, створюють діаграми..

Знати: основні властивості розглядаємих компонентів.

Вміти: використовувати компоненти Chartfx, F1Book, VtChart в створенні власних програм.

Обладнання: технічне забезпечення: ПЕОМ, програмне забезпечення: інтегроване середовище Delphi.

КОРОТКИ ТЕОРЕТИЧНІ ВІДОМОСТІ

Chartfx

Даний компонент є закінчений редактор діаграм з вбудованою інструментальною панеллю. Натискаючи кнопки інструментальної панелі користувач може задавати нові дані.

Доступ до властивостей Chartfx під час проектування може здійснюватися за допомогою Інспектора Об'єктів або клацанням правої кнопки миші і вибором з контекстного меню команди *Properties* (Властивості). При виборі цієї команди ви потрапите в багатосторінкове діалогове вікно, що дозволяє задати властивості компоненту. Зупинимось тільки на декількох з них, які ви можете задавати в цьому діалозі, в Інспекторіві Об'єктів або програмно.

Властивість *Series* на сторінці *Data Values* діалогу позначає число серій даних. Властивість *Points* на тій же сторінці діалогу позначає число значень по осі аргументів. Сторінка діалогу *Elements* дозволяє задати якісь характерні рівні, виділити кольором якісь смуги рівнів (опція *From - To - Color*), задати текст в рядку стану (опція *ID - Width - Text*). Інші властивості дозволяють задати тексти зверху діаграми, внизу, зліва, справа, задати координатні сітки і багато що інше. Слід звернути увагу на вибір шрифтів на сторінці Шрифти.

Під час роботи програми з'являється ще один набір кнопок.. Перша і друга зліва швидкі кнопки забезпечують відповідно читання і збереження діаграми. Діаграма зберігається у файлі з розширенням *.chf* і може бути прочитана в подальших сеансах роботи. Третя кнопка зліва заносить діаграму в буфер обміну, звідки її можна узяти в якійсь іншій програмі, наприклад, в MS Word, і вставити у документ. Кнопки у центральній частині панелі дозволяють змінювати тип діаграми або графіка. Друга справа група кнопок дозволяє вводити на діаграмі або графіці координатну сітку. Права група кнопок забезпечує завдання написів на зображенні, вибір шрифту написів і т.п. Головною з цих кнопок є друга справа. Вона викликає меню, що випадає, яке містить, зокрема, розділ *Data Editor*. Якщо ви виберете цей розділ, замість діаграми ви побачите вікно редактора даних, що відображаються на графіці або в діаграмі. Зробивши подвійне клацання на тому або іншому

числі, ви можете змінити його. Після того, як ви відредагували дані, знову клацніть на другій справа кнопці інструментальної панелі і зніміть виділення з розділу *Data Editor*. Ви знову побачите діаграму, що відображає введені вами дані.

F1Book

Цей компонент дозволяє вбудувати у вашу програму таблиці типу Excel, які користувач може заповнювати відповідними числами, а компонент проводитиме по заданих формулах обчислення і тут же відображатиме їх результати у вказаних комірках. У таблицю можна вбудувати діаграми і графіки різних типів. І всі зміни, що вносяться користувачем в дані таблиці, негайно відображатимуться в діаграмах. Таким чином, ви можете включати в свою програму різні бланки кошторисів, рахунків, відомостей, з якими працюватиме користувач, різні таблиці, що проводять статистичні або технічні розрахунки і т.п.

Перенесіть на форму компонент F1Book і клацніть на ній правою кнопкою миші. Виберіть у контекстному меню команду *Workbook Designer*. Перед вами з'явиться діалогове вікно проектування. Це вікно є декілька спрощеним варіантом Excel. Проектування таблиці проводиться фактично за тими ж правилами, що і в Excel. Ви можете писати в комірках необхідні написи, задаючи шрифт, його стиль, обрамлення. Можете записувати формули.

Права швидка кнопка дозволяє ввести на сторінку діаграми і графіки. Щоб задати діаграму, треба спочатку виділити курсором в таблиці дані, які повинні відобразитися в діаграмі, потім натиснути кнопку введення діаграми, після цього вказати курсором рамку, в якій повинна відобразитися діаграма. В результаті ви потрапите діалогове вікно, в якому зможете вибрати тип діаграми і необхідні атрибути.

Клацнувши правою кнопкою миші на компоненті F1Book, ви можете вибрати ще одну команду – *Properties*. У діалоговому вікні, що з'явилося при цьому, ви можете, задати опції, що визначають, що буде видне або не видно в таблиці при роботі додатку: заголовки рядків і стовпців ("Row Heading" і "Column Heading"), сітка ("Gridlines"), формули обчислень ("Formulas") і т.п.

VtChart – діаграма

Призначений для графічного представлення числових даних.

Схема використання:

1. Компонент поміщають на форму.
2. У контекстному меню компонента вибрати *Edit Chart*, що викликає редактор компонента. Закладка *Series* відкриває доступ до так званих серій, які відображають набори чисел у графічному вигляді. Щоб відобразити дані, необхідно створити як мінімум одну серію – для цього у вікні редактора клацнути на кнопці Add та вибрати потрібний тип діаграми.

3. Після закриття вікна реактора компонент буде містити діаграму, реальний вигляд якою залежить від фактичних даних.

САМОСТІЙНА ПІДГОТОВКА ДО ВИКОНАННЯ ЛАБОРАТОРНОЇ РОБОТИ

4. По літературі до лабораторної роботи вивчити основи роботи з компонентами Chartfx, F1Book, VtChart в інтегрованому середовищі розробки Delphi.
5. Відповісти на контрольні питання

ПОРЯДОК ВИКОНАННЯ РОБОТИ

7. Ознайомитись з компонентами з компонентами Chartfx, F1Book, VtChart та основними їх властивостями.
8. Виконати практичні завдання відповідно до свого варіанту.
9. Оформити звіт (завдання до роботи, тексти програм, висновки).

Варіант №1

У цирку можна купити різні по вартості білети:

- місця навколо арени коштують 6468 грн.;
- в передніх рядах – 5345 грн.;
- в задніх рядах – 3937 грн.

Кількість білетів, проданих на вказані місця на 6 днів тижня, вводяться в таблицю. Розрахуйте суму виручки від продажу білетів на кожен із 6 днів тижня та загальну вартість виручки. Побудувати діаграму.

День тижня	Місця навколо арени	Місця в передніх рядах	Місця в задніх рядах	Денна виручка
Вівторок	5298	5108	1112	
Середа	44121	5209	2353	
Четвер	3326	4598	2401	
П'ятниця	6422	5107	3203	
Субота	6131	4558	3445	
Неділя	6468	5225	3544	
Загальна виручка				

Варіант №2

Клієнт відкрив рахунок в банку на деяку суму під 18% річних. Яка сума буде на його рахунку через 13 років? Відобразити щорічні зміни у вигляді таблиці та побудувати діаграму. Приклад таблиці див. Варіант 2.

Варіант №3

Створити таблицю класного журналу на 10 учнів. Побудуйте діаграму.

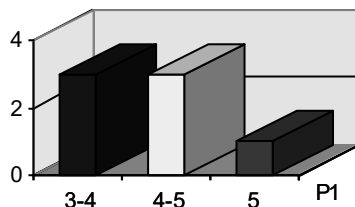
ПІБ	Вік	Математика	Укр. мова	кількість макулатури
-----	-----	------------	-----------	----------------------

Знайти середній бал з укр. мови та математики, загальну кількість зібраної макулатури.

Варіант №4

Створити таблицю для визначення середньої успішності групи учнів. Побудувати діаграму розподілу середнього балу по інтервалам (2,3),(3,4),(4,5) як це показано нижче:

ПІБ	Навчальні предмети				
	Алгебра	Укр. мова	Фізика	Ін. мова	Серед. бал
Алексєєв О.О.	3	4	5	5	4,25
Борисов Б.Б.	3	4	3	3	3,25
Іванов І.І.	4	4	4	4	4
Ніколаєв Н.Н.	5	5	5	5	5
Петров П.П.	4	4	4	4	4
Сергєєв С.С.	4	3	4	4	3,75
Сидоров С.С.	4	3	3	3	3,25



Варіант №5

Клієнт відкрив рахунок в банку на деяку суму під 12% річних. Яка сума буде на його рахунку через 10 років? Відобразити щорічні зміни у вигляді таблиці та побудувати діаграму.

Нарахування % протягом 10 років			
Рік	Сума	Приріст	Всього
1999	2500,00	300	2800,00

Варіант №6

Створити та оформити по зразку таблицю, виконати усі необхідні розрахунки, побудувати діаграму.

Прізвище	тариф	кіль-ть годин	До оплати	податок	на руки
1.Іванов	20	60			
2.Петров	20	50			
3.Сидоров	20	100			
4.Сидоров	20	45			
5.Сидоров	20	45			
ВСЬОГО					
Середня зарплата					

Варіант №7

Створити таблицю класного журналу на 10 учнів. Побудуйте діаграму.

ПІБ	Кількість пропусків	Математика	Укр. мова	Трудове навчання	Сер. успішність

Побудувати діаграму, яка відображує пропуски та діаграму середньої успішності учнів.

Варіант №8

Створити таблицю наявності товару на складі 7 найменувань. Побудувати дві діаграми: по кількості товару та по його вартості.

№	Найменування	Кількість (шт.)	Ціна (за шт.)	Вартість
1				

Варіант №9

Підготувати товарний чек, де зафіксована купівля декількох найменувань (шести-восьми) товарів та побудувати діаграму. Приклад наведений нижче.

Товарний чек				
Номер	Назва	Ціна	Кількість	Сума
1	Зошит 1	0,45	4	1,80
2	Зошит 2	0,6	5	3,00
3	Ручка 1	2,3	4	9,20
4	Ручка 2	1,4	5	7,00
5	Олівець	0,2	8	1,60
6	Гумка	0,4	1	0,40
7	Лінійка	0,35	2	0,70
			Всього	23,70

Варіант №10

Створити таблицю наукових статей в книжковому магазині. (10 записів). Побудувати дві діаграми: по кількості статей та по їх вартості.

№	Автор	Тема	Кількість (шт.)	Вартість
1				

Варіант №11

Створити таблицю по продажу автомобілів (10 записів). Побудувати дві діаграми: по марках проданих автомобілів та ціни за 1 шт.

№	Модель авто	Кількість (шт.)	Ціна (за шт.)	Вартість
1				

ПИТАННЯ ДЛЯ САМОСТІЙНОЇ РОБОТИ

1. Опишіть дію та основні властивості компонента Chartfx.
2. Опишіть дію та основні властивості компонента F1Book.
3. Опишіть дію та основні властивості компонента VtChart.
4. Яка різниця між компонентами Chartfx та F1Book?
5. Яка різниця між компонентами VtChart та F1Book?
6. Яка різниця між компонентами Chartfx та VtChart?

ЛІТЕРАТУРА

1. Бадд Т. Объектно-ориентированное программирование в действии / Перев. с англ. – СПб.: Питер, 1997. – 464 с.
2. Бобровский С. Delphi: учебный курс – СПб: Питер, 2001. – 640 с.
3. Буч Г. Объектно-ориентированный анализ и проектирование, 2-е изд./ Пер. с англ. – М.: «Издательство Бином», СПб.: «Невский диалект», 1999. – 560 с.
4. Дантеманн Д., Мишель Д., Тейлор Д. Программирование в среде Delphi. Киев: Диа-Софт Лтд., 1995.
5. Дарахвидзе П., Марков Е. Delphi-среда визуального программирования. СПб.: ВНУ-Санкт-Петербург, 1995.
6. Жуков А. Изучаем Delphi – СПб: Питер, 2000. – 352 с.
7. Конопка Р. Создание оригинальных компонентов в среде Delphi. К.: Диа-Софт Лтд., 1996.
8. Миллер Т., Пауэл Д. Использование Delphi 3. – К.: Диалектика, 1997. – 768 с.
9. Оузьер Д., Гробман С, Батсон С. Delphi 2. Освой самостоятельно: Пер. с англ. – М.: Восточная Книжная Компания, 1997. – 624 с.
10. Соколов В.В. Основи програмування в Delphi // Комп'ютер у школі та сім'ї. – №3, 2001. – С. 21-25.
11. Черняхівський В.В. Delphi-4: сучасна технологія візуального програмування. – Львів: БаК, 1999. – 196 с.

ЗВІТ ДО РОБОТИ

10. Оформити лабораторну роботу в зошиті. Записати тему, мету, відповіді на запитання для самостійної підготовки.
11. Виконати завдання згідно свого варіанту.
12. Зробити висновок.

ЛАБОРАТОРНА РОБОТА №10 СТВОРЕННЯ РЕЛЯЦІЙНОЇ БАЗИ ДАНИХ

Мета роботи Ознайомитись з програмами Database Desktop та BDE Administrator, а також з компонентами для роботи з базами даних DataSource, Table, DBGrid, DBNavigator, DBText, DBEdit, DBMemo. Навчитися створювати програми, які виводять дані з файлу бази даних та вносити в них зміни.

Знати: основні властивості розглядаємих компонентів.

Вміти: використовувати компоненти DataSource, Table, DBGrid, DBNavigator, DBText, DBEdit, DBMemo в створенні власних програм.

Обладнання: технічне забезпечення: ПЕОМ, програмне забезпечення: інтегроване середовище Delphi.

КОРОТКИ ТЕОРЕТИЧНІ ВІДОМОСТІ

Table

Компонент Table представляє всю таблицю бази даних. Властивості компоненту:

Name Ім'я компоненту.

DatabaseName Ім'я бази даних, частиною якої є таблиця (файл даних), для доступу до якої використовується компонент. Як значення властивості слід застосовувати псевдонім бази даних.

TableName Ім'я файлу даних (таблиці даних), для доступу до якого використовується компонент

TableType Тип таблиці. Таблиця може бути набором даних у форматі Paradox (ttParadox), dBase (ttDBase), FoxPro (ttFoxPro) або бути форматованим текстовим файлом (ttASCII).

Active Ознака того, що таблиця активна (файл даних відкритий). В результаті привласнення властивості значення True відбувається відкриття файлу таблиці.

Query

Компонент Query представляє частину бази даних – записи, вміст яких задовольняють критеріям SQL-запиту до таблиці. Властивості компоненту приведені:

Name Ім'я компоненту. Використовується компонентом DataSource для зв'язку результату виконання запиту (набору записів) з компонентом, що забезпечує проглядання записів, наприклад DBGrid.

SQL Записаний на мові SQL-запит до бази даних (до таблиці).

Active При привласненні властивості значення True активізує виконання запиту.

DataSource

Компонент DataSource забезпечує зв'язок між даними, представленими компонентом Table або Query та компонентами відображення даних (DBEdit, DBMemo, DBGrid).

Властивості компоненту:

Name Ім'я компоненту.
DataSet Компонент, який представляє собою вхідні дані (Table або Query).

Компоненти DBEdit і DBMemo забезпечують перегляд і редагування полів записів бази даних, компонент DBText – тільки перегляд. Властивості компонентів:

DataSource Компонент-джерело даних
DataField Поле бази даних, для відображення або редагування якого використовується компонент.

DBGrid

Компонент DBGrid використовується для перегляду і редагування бази даних в режимі таблиці. Властивості компоненту приведені в табл. 2.21.

DataSource Джерело що відображаються в таблиці даних (компонент DataSource).

Columns Властивість Columns є масив компонентів типу Column, кожен з яких визначає колонку таблиці і інформацію, що відображається в ній.

Options.dgTitles Дозволяє виведення рядка заголовка стовпців.

Options.dgIndicator Дозволяє виведення колонки індикатора. Під час роботи з базою даних поточний запис позначається в колонці індикатора трикутником, новий запис – зірочкою, редагована – спеціальним значком.

Options.dgColumnResize Дозволяє змінювати під час роботи програми ширину колонок таблиці.

Options.dgColLines Дозволяє виводити лінії, що розділяють колонки таблиці.

Options.dgRowLines Дозволяє виводити лінії, що розділяють рядки таблиці.

САМОСТІЙНА ПІДГОТОВКА ДО ВИКОНАННЯ ЛАБОРАТОРНОЇ РОБОТИ

1. Записати в зошит тему, мету, обладнання, практичні завдання свого варіанту (вибирається згідно номеру комп'ютера).

2. По літературі до лабораторної роботи вивчити основи роботи з компонентами DataSource, Table, DBGrid, DBNavigator, DBText, DBEdit, DBMemo в інтегрованому середовищі розробки Delphi.

3. По інструкції до лабораторної роботи ознайомитись з порядком виконання роботи.

ПОРЯДОК ВИКОНАННЯ РОБОТИ

1. Запустити Delphi.

2. Виконати практичні завдання відповідно до свого варіанту.

3. Оформити звіт (завдання до роботи, тексти програм, висновки).

Варіант №1

Студенти.

Інформація про студентів групи зберігається в базі даних, зокрема, зберігається: код, прізвище, оцінка з інформатики, оцінка з математики, оцінка з фізики, оцінка з педагогіки.

Варіант №2

Зоопарк.

В таблиці даних зберігається інформація про тварин: код, назва тварини, продукт, який надається тварині, кількість одиниць за добу.

Варіант №3

Виконавці.

В таблиці зберігається інформація про відомих виконавців, зокрема, зазначається: код, прізвище, назва найпопулярнішої пісні, кількість проданих дисків з цієї піснею.

Варіант №4

Товар на складі.

В таблиці зберігається інформація про товар (код, назва, наявність на складі, фірма виробник, ціна одиниці на складі в у.о., кількість одиниць).

Варіант №5

Читачі в бібліотеці.

В абонементі у вигляді таблиці зберігається інформація про читачів (номер читацького квитка, прізвище, адреса, дата останнього візиту).

Варіант №6

Робітники на фірмі.

В таблиці зберігається інформація про робітників фірми (код, прізвище, адреса, посада, оклад).

Варіант №7

Прайс-лист.

Інформація про пропозиції комп'ютерної фірми зберігаються в таблиці, зокрема, в кожному рядку міститься інформація про конфігурацію (код, тип процесора, тип монітора тощо в окремих полях).

Варіант №8

Білет на потяг.

Інформація про людей в касі зберігається в таблиці, зокрема, зазначається: код, прізвище, станція призначення, кількість км до місця призначення, тип місця.

Варіант №9

Банк.

Інформація про клієнтів банку зберігається в таблиці, зокрема, зазначається: код, прізвище, адреса, сума початкового вкладу, тип вкладу.

Варіант №10

Бібліотечний каталог.

Інформація про написані статті зберігається в таблиці, зокрема, зазначається: код статті, автор статті, назва статті, науковий напрямок, дата написання.

Варіант №11

Туристична фірма.

Інформація про тури зберігається в таблиці, зокрема, зазначається: код туру, місце призначення, кількість діб, вартість квитка.

ПИТАННЯ ДЛЯ САМОСТІЙНОЇ РОБОТИ

1. Для чого призначені компоненти DataSource та Table?
2. Опишіть програму Database Desktop та розкрийте її можливості?
3. Опишіть призначення програми BDE Administrator?

ЛІТЕРАТУРА

1. Бадд Т. Объектно-ориентированное программирование в действии / Перев. с англ. – СПб.: Питер, 1997. – 464 с.
2. Бобровский С. Delphi: учебный курс – СПб: Питер, 2001. – 640 с.
3. Буч Г. Объектно-ориентированный анализ и проектирование, 2-е изд./ Пер. с англ. – М.: «Издательство Бином», СПб.: «Невский диалект», 1999. – 560 с.
4. Дантемманн Д., Мишель Д., Тейлор Д. Программирование в среде Delphi. Киев: Диа-Софт Лтд., 1995.
5. Дарахвидзе П., Марков Е. Delphi-среда визуального программирования. СПб.: ВНУ-Санкт-Петербург, 1995.
6. Жуков А. Изучаем Delphi – СПб: Питер, 2000. – 352 с.
7. Конопка Р. Создание оригинальных компонентов в среде Delphi. К.: Диа-Софт Лтд., 1996.
8. Миллер Т., Пауэл Д. Использование Delphi 3. – К.: Диалектика, 1997. – 768 с.
9. Оузьер Д., Гробман С, Батсон С. Delphi 2. Освой самостоятельно: Пер. с англ. – М.: Восточная Книжная Компания, 1997. – 624 с.
10. Соколов В.В. Основи програмування в Delphi // Комп'ютер у школі та сім'ї. – №3, 2001. – С. 21-25.
11. Черняхівський В.В. Delphi-4: сучасна технологія візуального програмування. – Львів: БаК, 1999. – 196 с.

ЗВІТ ДО РОБОТИ

1. Оформити лабораторну роботу в зошиті. Записати тему, мету, відповіді на запитання для самостійної підготовки.
2. Виконати завдання згідно свого варіанту.
3. Зробити висновок.

ЛАБОРАТОРНА РОБОТА №11 СТВОРЕННЯ ЗАПИТІВ ДО БАЗ ДАНИХ

Мета роботи: Ознайомитись з програмами Database Desktop та BDE Administrator, а також з компонентами для роботи з базами даних DataSource, Table, DBGrid, DBNavigator, DBText, DBEdit, DBMemo, TQuery. Навчитися створювати SQL-запити за допомогою компонентів TQuery.

Знати: основні властивості розглядаємих компонентів.

Вміти: використовувати компоненти DataSource, Table, DBGrid, DBNavigator, DBText, DBEdit, DBMemo, TQuery в створенні власних програм.

Обладнання: технічне забезпечення: ПЕОМ, програмне забезпечення: інтегроване середовище Delphi.

КОРОТКИ ТЕОРЕТИЧНІ ВІДОМОСТІ

Компонент **Table** представляє всю таблицю бази даних. Властивості компоненту:

Name Ім'я компоненту.

DatabaseName Ім'я бази даних, частиною якої є таблиця (файл даних), для доступу до якої використовується компонент. Як значення властивості слід застосовувати псевдонім бази даних.

TableName Ім'я файлу даних (таблиці даних), для доступу до якого використовується компонент

TableType Тип таблиці. Таблиця може бути набором даних у форматі Paradox (ttParadox), dBase (ttDBase), FoxPro (ttFoxPro) або бути форматованим текстовим файлом (ttASCII).

Active Ознака того, що таблиця активна (файл даних відкритий). В результаті привласнення властивості значення True відбувається відкриття файлу таблиці.

Компонент **Query** представляє частину бази даних – записи, вміст яких задовольняють критеріям SQL-запиту до таблиці. Властивості компоненту приведені:

Name Ім'я компоненту. Використовується компонентом DataSource для зв'язку результату виконання запиту (набору записів) з компонентом, що забезпечує проглядання записів, наприклад DBGrid.

SQL Записаний на мові SQL-запит до бази даних (до таблиці).

Active При привласненні властивості значення True активізує виконання запиту.

Компонент **DataSource** забезпечує зв'язок між даними, представленими компонентом Table або Query та компонентами відображення даних (DBEdit, DBMemo, DBGrid).

Властивості компоненту:

<i>Name</i>	Ім'я компоненту.
<i>DataSet</i> (Table або Query).	Компонент, який представляє собою вхідні дані

Компоненти **DBEdit** і **DBMemo** забезпечують перегляд і редагування полів записів бази даних, компонент **DBText** – тільки перегляд. Властивості компонентів:

<i>DataSource</i>	Компонент-джерело даних
<i>DataField</i>	Поле бази даних, для відображення або редагування якого використовується компонент.

Компонент **DBGrid** використовується для перегляду і редагування бази даних в режимі таблиці. Властивості компоненту приведені в табл. 2.21.

DataSource Джерело що відображаються в таблиці даних (компонент DataSource).

Columns Властивість Columns є масив компонентів типу Column, кожен з яких визначає колонку таблиці і інформацію, що відображається в ній.

Options.dgTitles Дозволяє виведення рядка заголовка стовпців.

Options.dgIndicator Дозволяє виведення колонки індикатора. Під час роботи з базою даних поточний запис позначається в колонці індикатора трикутником, новий запис – зірочкою, редагована – спеціальним значком.

Options.dgColumnResize Дозволяє змінювати під час роботи програми ширину колонок таблиці.

Options.dgColLines Дозволяє виводити лінії, що розділяють колонки таблиці.

Options.dgRowLines Дозволяє виводити лінії, що розділяють рядки таблиці.

САМОСТІЙНА ПІДГОТОВКА ДО ВИКОНАННЯ ЛАБОРАТОРНОЇ РОБОТИ

1. Записати в зошит тему, мету, обладнання, практичні завдання свого варіанту (вибирається згідно номеру комп'ютера).

2. По літературі до лабораторної роботи вивчити основи роботи з компонентами DataSource, Table, DBGrid, DBNavigator, DBText, DBEdit, DBMemo, TQuery в інтегрованому середовищі розробки Delphi.

3. По інструкції до лабораторної роботи ознайомитись з порядком виконання роботи.

ПОРЯДОК ВИКОНАННЯ РОБОТИ

1. Запустити Delphi.
2. Виконати практичні завдання відповідно до свого варіанту.
3. Оформити звіт (завдання до роботи, тексти програм, висновки).

Варіант №1

Студенти.

- 1) Вивести прізвища відмінників;
- 2) Вивести інформацію про студентів, середній бал яких менше заданого числа, тобто їм не нараховується стипендія;
- 3) Вивести всю інформацію про заданого студента.

Варіант №2

Зоопарк.

- 1) Вивести назву тварини, яка найдорожче обходиться зоопарку;
- 2) Вивести назви тварин, що їдять м'ясо;
- 3) Підрахувати загальну кількість витрат зоопарку на всіх тварин.

Варіант №3

Виконавці.

- 1) Визначити найпопулярнішого виконавця;
- 2) Вивести всю інформацію про заданого виконавця;
- 3) Вивести прізвища виконавців, прізвища яких починається із заданої літери.

Варіант №4

Товар на складі.

- 1) Вивести назву фірми виробника, який випускає заданий товар найменшої вартості;
- 2) Вести інформацію про товар, що випускається заданою фірмою (одна фірма може випускати товар кількох типів);
- 3) Підрахувати, скільки одиниць і на яку суму є заданого товару на складі незалежно від фірми, що його виготовила.

Варіант №5

Читачі в бібліотеці.

- 1) Знайти читачів, яким вже час відвідати бібліотеку, термін повернення книжки – 14 діб.
- 2) Знайти середній термін, на який взяли книжки.
- 3) Вивести інформацію про заданого читача.

Варіант №6

Робітники на фірмі.

- 1) Знайти середню заробітну плату робітників фірми;
- 2) Вивести інформацію про всіх робітників, прізвища яких починаються на літеру О.
- 3) Вивести прізвища робітників, заробітна плата яких перевищує задану суму.

Варіант №7

Прайс-лист.

- 1) Вивести інформацію про найдешевший комп'ютер;
- 2) Вивести інформацію про комп'ютери із зазначеним типом процесора;
- 3) Визначити, скільки коштує комп'ютер із заданою конфігурацією.

Варіант №8

Білет на потяг.

- 1) Визначити всіх пасажирів, що їдуть до заданої станції;
- 2) Визначити, скільки білетів на купейні та плацкартні місця продані;
- 3) Вивести інформацію про заданого пасажера.

Варіант №9

Банк.

- 1) Вивести інформацію про вкладників, сума вкладу яких більше заданого значення;
- 2) Вивести повну інформацію про вкладників, прізвище яких починається на літеру А;
- 3) Підрахувати загальну суму початкового вкладу.

Варіант №10

Бібліотечний каталог.

- 1) Знайти всі статті певного автора;
- 2) Знайти відповідну статтю;
- 3) Вивести всі статті напрямку «Інформатика».

Варіант №11

Туристична фірма.

- 1) Знайти інформацію про тур у задану країну;
- 2) Знайти найдорожчий тур;
- 3) Знайти середню вартість квитків у фірмі.

ПИТАННЯ ДЛЯ САМОСТІЙНОЇ РОБОТИ

1. Призначення мови SQL?
2. Призначення та формат запису виразу SELECT?
3. Призначення та формат запису виразу UPDATE?
4. Призначення та формат запису виразу DELETE?
5. Призначення та формат запису виразу INSERT?
6. Призначення службових слів мови SQL – WHERE, BETWEEN?
7. Реалізація запитів в Delphi?
8. Створення таблиць в Database Desktop?

ЛІТЕРАТУРА

1. Бадд Т. Объектно-ориентированное программирование в действии / Перев. с англ. – СПб.: Питер, 1997. – 464 с.
2. Бобровский С. Delphi: учебный курс – СПб: Питер, 2001. – 640 с.
3. Буч Г. Объектно-ориентированный анализ и проектирование, 2-е изд./ Пер. с англ. – М.: «Издательство Бином», СПб.: «Невский диалект», 1999. – 560 с.
4. Дантемманн Д., Мишель Д., Тейлор Д. Программирование в среде Delphi. Киев: Диа-Софт Лтд., 1995.
5. Дарахвидзе П., Марков Е. Delphi-среда визуального программирования. СПб.: ВНУ-Санкт-Петербург, 1995.

6. Жуков А. Изучаем Delphi – СПб: Питер, 2000. – 352 с.
7. Конопка Р. Создание оригинальных компонентов в среде Delphi. К.: Диа-Софт Лтд., 1996.
8. Миллер Т., Пауэл Д. Использование Delphi 3. – К.: Диалектика, 1997. – 768 с.
9. Оузьер Д., Гробман С, Батсон С. Delphi 2. Освой самостоятельно: Пер. с англ. – М.: Восточная Книжная Компания, 1997. – 624 с.
10. Соколов В.В. Основы програмування в Delphi // Комп'ютер у школі та сім'ї. – №3, 2001. – С. 21-25.
11. Черняхівський В.В. Delphi-4: сучасна технологія візуального програмування. – Львів: БаК, 1999. – 196 с.

ЗВІТ ДО РОБОТИ

1. Оформити лабораторну роботу в зошиті. Записати тему, мету, відповіді на запитання для самостійної підготовки.
2. Виконати завдання згідно свого варіанту.
3. Зробити висновок.

ЛАБОРАТОРНА РОБОТА №12 СТВОРЕННЯ ЗВІТІВ ДО БАЗИ ДАНИХ

Мета роботи: Отримати знання та практичні вміння з виведення інформації з бази даних або запиту у звіт у середовищі Delphi.

Знати: основні властивості компонентів TQuickRep, TQRBand.

Вміти: виводити інформацію у звіт засобами середовища Delphi.

Обладнання: технічне забезпечення: ПЕОМ, програмне забезпечення: інтегроване середовище Delphi.

КОРОТКИ ТЕОРЕТИЧНІ ВІДОМОСТІ

В більшості випадків дані, які зберігаються у таблицях БД, необхідно тим чи іншим способом опублікувати її, створюючи таз звані **звіти**. Вже в першу версію Delphi були включені засоби генерації звітів. Компоненти, які відповідають за побудову звітів у Delphi (до сьомої версії) розміщуються на закладці **QReport** (у сьомій і далі – Rave).

Центральним компонентом при створенні звіту є компонент **TQuickRep**, який визначає властивості звіту в цілому. Даний компонент завжди розміщують на окремій порожній формі, створюючи за її допомогою візуальне середовище конструювання звіту.

Порожнє внутрішнє поле компонента імітує чистий лист паперу. На дане поле поміщаються інші компоненти точно так, якби створювалась звичайна форма, однак, на відміну від звичайної форми, форма з компонентом TQuickRep, аналогічно до модуля даних, ніколи не візуалізується у робочій програмі.

ВЛАСТИВОСТІ TQuickRep

Bands – містить логічні властивості, які після вибору в них значень True включають у звіт заголовки стовпців, детальну інформацію, нижній колонтитул, верхній колонтитул, заголовок звіту.

DataSet – визначає набір даних, на основі яких створюється звіт.

Frame – визначає параметри рамки звіту: колір ліній, наявність ліній зверху, знизу, зліва, справа, стиль ліній, товщина.

PrintIfEmpty – дозволяє або забороняє друк звіту в тому випадку, коли він не містить даних.

ReportTitle – ім'я звіту.

Багато властивостей звіту можна встановити на етапі конструювання за допомогою редактора властивостей – контекстне меню компонента TQuickRep та виберіть команду *Report Setting*.

Звіт в основному будується з компонентів-полос **TQRBand**, за допомогою яких формуються різні його фрагменти, такі як загальний заголовок, заголовок кожної сторінки, заголовок колонок звіту, область для відображення даних з таблиці БД тощо. Вміст полоси залежить від таких її властивостей:

ВЛАСТИВОСТІ TQRBand

BandType – може містити одне з наступних значень, яке визначає, що буде містити полоса:

rbTitle – заголовок звіту в цілому;

rbPageHeader – верхній колонтитул;

rbDetail – дані з набору даних (який конкретно НД буде відображатися у цій полосі визначається значенням властивості *DataSet* компонента *TQuickRep*);

rbPageFooter – нижній колонтитул, в якому звичайно розміщується номер сторінки звіту та інша допоміжна інформація;

rbSummery – загальні для звіту в цілому стогові дані;

rbGroupHeader – заголовок групи даних;

rbGroupFooter – підвал групи даних;

rbColumnHeader – заголовки колонок звіту.

На полосах *TQRBand* розміщуються інші компоненти звіту, які діляться на компоненти для оформлення та для відображення. Компоненти для оформлення містять різну допоміжну інформацію – надписи, зображення тощо. Дана інформація не є частиною даних з БД, але слугує для зручності перегляду звіту. Компоненти, що відображають дані отримують інформацію з відповідних полів НД або обчислюють її на основі цієї інформації.

За структурою розрізняються три види звітів: прості, типу головний-детальний та ком позитивний. У простих звітах відображуються дані з одного НД, наприклад, список всіх накладних, пов'язаних з відпуском книг конкретному покупцю, упорядковані за датою або за сумою. У простому звіті повинна використовуватися хоча б одна полоса *TQRBand* типу *rbDetail*, яка при перегляді звіту буде автоматично повторюватися стільки разів, якільки записів містить НД.

Будь-який звіт можна надрукувати, якщо звернутися до його методу *Print* або переглянути у спеціальному вікні перегляду, якщо викликати метод *Preview*.

Для виводу звіту у файл необхідно спочатку підготувати його за допомогою звернення до методу *Prepare*, потім зберегти у файлі методом *Save* об'єкту *TQuickRep.QRPrinter*, після чого видалити цей об'єкт та помістити *NIL* у властивість *TQuickRep.QRPrinter*.

САМОСТІЙНА ПІДГОТОВКА ДО ВИКОНАННЯ ЛАБОРАТОРНОЇ РОБОТИ

1. Записати в зошит тему, мету, обладнання, практичні завдання свого варіанту (вибирається згідно номеру комп'ютера).

2. По літературі до лабораторної роботи вивчити основи роботи з компонентами *TQuickRep*, *TQRBand*, *QRLabel*, *QRDBText*, *QRMemo* в інтегрованому середовищі розробки *Delphi*.

3. По інструкції до лабораторної роботи ознайомитись з порядком виконання роботи.

ПОРЯДОК ВИКОНАННЯ РОБОТИ

1. Запустити *Delphi*.

2. Виконати практичне завдання.
3. Оформити звіт (завдання до роботи, тексти програм, висновки).

ЗАВДАННЯ

Вивести інформацію з вашої бази даних (лабораторна робота №10) у звіт.

ПИТАННЯ ДЛЯ САМОСТІЙНОЇ РОБОТИ

1. Що таке звіт? Яке його призначення?
2. Поясніть призначення закладки QReport.
3. Поясніть призначення компонента TQuickRep.
4. Поясніть призначення властивостей Bands та DataSet компонента TQuickRep.
5. Поясніть призначення компонента TQRBand.
6. Поясніть призначення властивості BandType компонента TQRBand.
7. Поясніть призначення компонентів QRLabel та QRDBText.
8. Як зберегти звіт у файл?
9. Як вивести звіт на друк?

ЛІТЕРАТУРА

1. Бадд Т. Объектно-ориентированное программирование в действии / Перев. с англ. – СПб.: Питер, 1997. – 464 с.
2. Бобровский С. Delphi: учебный курс – СПб.: Питер, 2001. – 640 с.
3. Буч Г. Объектно-ориентированный анализ и проектирование, 2-е изд./ Пер. с англ. – М.: «Издательство Бином», СПб.: «Невский диалект», 1999. – 560 с.
4. Дантемманн Д., Мишель Д., Тейлор Д. Программирование в среде Delphi. Киев: Диа-Софт Лтд., 1995.
5. Дарахвидзе П., Марков Е. Delphi-среда визуального программирования. СПб.: ВНУ-Санкт-Петербург, 1995.
6. Жуков А. Изучаем Delphi – СПб.: Питер, 2000. – 352 с.
7. Конопка Р. Создание оригинальных компонентов в среде Delphi. К.: Диа-Софт Лтд., 1996.
8. Миллер Т., Пауэл Д. Использование Delphi 3. – К.: Диалектика, 1997. – 768 с.
9. Оузьер Д., Гробман С, Батсон С. Delphi 2. Освой самостоятельно: Пер. с англ. – М.: Восточная Книжная Компания, 1997. – 624 с.
10. Соколов В.В. Основи програмування в Delphi // Комп'ютер у школі та сім'ї. – №3, 2001. – С. 21-25.
11. Черняхівський В.В. Delphi-4: сучасна технологія візуального програмування. – Львів: БаК, 1999. – 196 с.

ЗВІТ ДО РОБОТИ

4. Оформити лабораторну роботу в зошиті. Записати тему, мету, відповіді на запитання для самостійної підготовки.
5. Виконати завдання згідно свого варіанту.
6. Зробити висновок.