# **MІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**КОЛЕДЖ ІНФОРМАЦІЙНИХ СИТЕМ І ТЕХНОЛОГІЙ КНЕУ**

##### ЗАТВЕРДЖУЮ

# Заступник директора з

# навчальної роботи

\_\_\_\_\_\_\_\_\_\_\_***Зубченко Н.І.***

“\_\_\_\_” \_\_\_\_\_\_\_\_\_ 200 р.

***МЕТОДИЧНІ РЕКОМЕНДАЦІЇ ДО ВИКОНАННЯ***

***лабораторних та практичних робіт***

***oб’єктно-орієнтованого програмування МП Delphi***

з дисципліни “ ***Алгоритмічні мови та програмування***”

**для спеціальності 5.080405 “Програмування для ЕОТ та АС ”**

Розробив викладач ***Карлюченко О.Г.***

Обговорено і схвалено на засіданні предметної комісії обчислювальної техніки КІСІТ КНЕУ

Протокол №\_\_\_ від\_\_\_\_\_\_\_\_\_\_200 р.

Голова предметної комісії:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ ***Іванова Л.Д.***

## **КИЇВ**

***Особиста бібліотека програміста***

При накопиченні значної кількості процедур і функцій, найчастіше досить простих, але часто застосовуваних розроблювачем, має сенс поєднувати їх в особистий комплекс, названий особистою бібліотекою програміста. Найчастіше в бібліотеку заносяться процедури побудови рамок, окантовок, вікон із заголовками, форматування видимого і невидимого курсору, установка кольору і фону, подача звукового сигналу (при аварійних ситуаціях). Побудова різних типів меню (вертикальне, горизонтальне, об'єднане), різні варіанти введення даних.

Кожний з перерахованих модулів повинeн бути оформлений у вигляді процедури, ім'я якої потрібно давати загальне та зрозуміле. Текстовий файл з модулями бібліотеки має розширення **.pas**

**Структура модуля** :

***Unit < Servis -*** ім***’***ямодуля > **;**{ заголовок бібліотеки}

***Interface*** { розділ декларації, опис підпрограм }

***Uses*** crt, dos; {підключення стандартних бібліотек}

***Type ... ;*** {опис глобальних типів , міток, const }

***Label ... ;***

***Const ... ;***

***Procedure menu V;*** {опис підпрограм}

***Procedure menu G;***

***Implementation*** {розділ реалізації, виконуюча частина}

***Type ... ;*** {опис локальних типів , міток, const }

***Label ... ;***

***Const ... ;***

***Procedure menu V ;***

<тіло підпрограми V>

***Procedure menu G;***

# <тіло підпрограми G>

***Begin*** {ініціююча частина , виконується до

передачі управління головній програмі}

***End***.

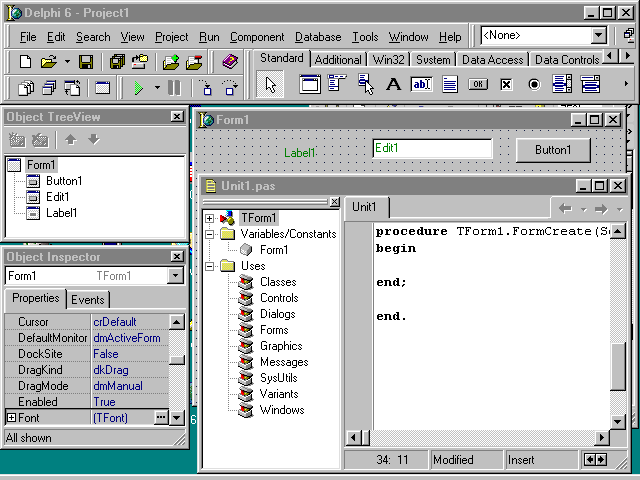
Для створення бібліотеки цей файл **Servis. pas** компілюємо через головне меню чи **Alt + F9**. У результаті компіляції одержуємо файл **Servis. tpu**.

При компіляції звичайного тексту програми з заголовками ***Program*** одержуємо завантажувальний модуль з розширенням **.EXE**. Часто необхідно вказати в опціях директорію для збереження зкомпільованого файлу.

Підключення бібліотеки в головній програмі :

***Program proba;***

***Uses Servis , Crt , Dos;***



**Borland Delphi. Середовище Delphi ,його складові.**

**Delphi** – це інтегроване середовище розробки, має гнучкий інструментарій для швидкого створення додатків, контекстні спливаючі підказки: основні і допоміжні.

Можливе використання додатків: баз даних, *Ассемблера*, *C*, але алгоритмічна основа *Object Pascal* (це розширення мови програмування Turbo Pascal 7.0.).

Існує могутній компілятор, що без проміжного (як би віртуального) π – коду, перетворює безпосередньо в машинний код. Працює в 10 –20 разів швидше звичайного компілятора.

Готові Delphi – додатки можуть бути у вигляді виконуючого модуля, чи у вигляді динамічних бібліотек, які можна використовувати в інших додатках.

Існує можливість швидкого візуального інтерфейсу і додатка.

У Delphi 2 основні фази розробки додатків:

1) конструювання форми;

2) кодування.

***Після запуску Delphi*** - ***на екрані 4 вікна:***

**I** Головне вікно; - верхній рядок екрана з заголовком: Delphi (версія) і Project1.

**II** Вікно проектувальника форм (вікно форми) – FORM1.

**III** Вікно редактора коду (вікно коду програми) – файл .PAS – code Editor.

**IV** Вікно інспектора об'єктів (Object inspector).

Усі вікна мають інтерфейсом Windows. Розташоваіо вікна мозаїкою.

***I* *Головне вікно:***

**1.** Рядок заголовка, де вказується версія Delphi, ім'я проекту (за замовчуванням Project1).

**2.** Основне меню (існує система допоміжних меню, через праву кнопку миші).

**3.** Третій, четвертий рядок екрана лівого поля – панель інструментів. Це кнопки з

спливаючими підказками, що повторюють пункти головного меню (піктограми).

**4.** Третій, четвертий рядок правого поля – палітра компонентів. Це піктограми з закладками

(Component palette), як багатосторінковий блокнот.

При закритті головного вікна відбувається вивантаження з DELPHI

***II******Вікно проектувальника форм***

з'являється автоматично при завантаженні DELPHI, чи F12.

**Форма** – це основа додатка Delphi – це розмічене поле, у якому розташовуються об'єкти, тобто компоненти з палітри компонентів.

Містить стандартний інтерфейс Windows. Компоненти: візуальні і невізуальні.

Робота у вікні форми - як гра з дитячим конструктором: з палітри компонентів вибираються об'єкти і розташовуються на формі вікна. Це і є основа «візуального» програмування, тобто заповнення вікна форми інтерфейсними елементами: кнопками, мітками, панелями.

***III******Редактор коду ( оброблювач події):***

Викликається подвійним клацанням миші у вікні інспектора об'єкта праворуч від необхідної події в сторінці Events. На вікні форми з”являэться вікно редактора коду з заготівкою тексту процедури оброблювача подій.

Ліворуч від вікна редактора коду з'являється ***вікно провідника*** (дослідника, браузера) коду. Він полегшує пошук потрібних елементів для великої кількості рядків коду.

**Головне меню:**

***File*** (файл); ***Edit*** (редагування); ***Searc****h* (пошук); ***View***(вид); ***Project*** (проект); ***Run*** (запуск); ***Component*** (компонент); ***Data base*** (база даних); ***Tools*** (інструменти); ***Workgroup*** (робоча група); ***Help*** (довідка).

**Панель інструментів:**

***Debug*** (налагодження); ***Run***(запуск); ***Pause***(пауза); ***StepOver*** (крок через);***TraceIntro*** (крок в).

***Standard*** (стандартна); ***New*** (створити); ***Open*** (відкрити); ***Save*** (зберегти);

***Save all*** (зберегти усі); ***Open project*** (відкрити проект); ***Add project*** (додати в проект); ***Remove*** ***from project*** (видалити з проекту).

**Панель інструментів:**

Custom (користувача);

Кнопка Help (довідка);

Палітра компонентів – компонента VCL (Visual Component Library) і Active X.

***IV*** Для зміни параметрів об'єктів використовується ***інспектор об'єктів***, що має дві закладки – ***властивості*** (Properties) та ***події*** (Events) ***.*** Для настроювання об'єкт активізується одним клацанням лівої кнопки миші.

У DELPHI існує ***компонентний підхід*** (метод): готові компоненти – це частини машинного коду, що можуть бути додані до програми клацанням миші.

Праворуч у вікні знаходиться палітра компонентів, компоненти яких мають ***властивості***, тобто характеристики. Властивості настроюються за допомогою інспектора об'єктів. Сукупність властивостей відображає видиму ( візуальну) сторону компонента:

• Положення щодо лівого верхнього кута

• Розміри

• Шрифт

• Текст

Властивості об'єкта відсортовані за алфавітом і по категоріях

Компоненти характеризуються ***методами*** (тобто програмні коди обробки властивостей) і ***подіями,*** котрі компонент приймає від додатка при виконанні дій.

***Сукупність подій*** – поведінкова сторона компонента, наприклад – чи буде компонент реагувати на клацання миші чи натискання кнопки, клавіші; як він буде поводитися в момент появи на екрані чи в момент зміни розмірів вікна і т.д.

Рядки таблиці інспектора об'єктів вибираються клацанням миші і можуть відображати ***прості і складені (складні) властивості***. Наприклад: прості властивості визначаються одним значенням (Caption).

Складні – це шрифт, стиль, розмір і т.д. закрити – відкрити список складних властивостей можна подвійним клацанням миші на властивості, чи клацанням по « – ».

Клацання на правій кнопці активізує зазначені властивості і з'являється кнопка для виклику діалогового вікна.

У складної властивості з'являється кнопка - це розкриваючийся список значень.

Виклик (активізація) інспектора об'єктів виконується через головне меню View – Object Inspector чи F11.

***Основні поняття об’єктно - орієнтованого програмування***

**Клас**- спеціальний тип даних, як категорія, яка має своі поля, місце і властивість.

Сама категорія визначається загальною властивістю, яку мають всі єкземпляри цієі категоріі, тобто конкретні об’єкти класу.

Клас - структурний тип. Наприклад в Паскалі - тип RECORD (записи).

Прикладом класу і класифікації може бути ієрархічне дерево „музичні інструменти”. Всі музичні інструменти мають загальну властивість - видавати звуки. Але методи вилучення звуків різні: вдаряти по струнах, дути у мунштук, натискати на клавіші.

***Класи мають:***

а) **поля** - дані об’єкту одного типу ( як тип RECORD );

б) **властивості** - поля даних, що впливають на поведінку об’єкта;

застосовуються для опису об’єкта, викликаючи методи їх обробки;

в) **методи** - підпрограммb з вихідним кодом обробки властивостей і полів.

При описі змінної типу клас для полів і властивостей цієї змінної в пам’яті виділяється (резервується) відповідний об’єм. При створенні об’єкту працює ***метод конструктора*** для початковоі ініціалізаціі полів об’єкта.

При знищенні об’єкта для звільнення пам’яті визивається ***метод деструктора***. Ці методи не можна визивати із програмного коду, вони виконуються тільки автоматично.

Змінна описана як клас, який являється вказівкою на єкземпляр класу (але без символу **^** ). Звернення до такоі змінної - звичайне, а до полів, властивостей, методів класу (тобто до членів класу) - через крапку, як до полів запису RECORD .

**ЗАУВАЖЕННЯ!!!**

***Нe можна описувати змінну типу клас всередині програми як локальну.***

***Наслідування, інкапсуляціі, поліморфізм –***

***три основні властивості класу***

**Наслідування** - це можливість породженого класу нащадка (descendant) наслідувати властивості свого предка (ansestor). При цьому наслідуються всі поля, властивості і методи батьківського класу, не потрібно іх засновувати і розробляти додатково. Але породжений клас може володіти більшими можливостями, наприклад, властивість видавати звуки перенесено з батьківського класу „музичні інструменти” на клас „гітара”. Механізм наслідування дозволяє багаторазово застосовувати програмний код. Відповідно класи м.б. систиматизовані в ієархію, як у візуальній біблітеці компонент Delphi (VCL)

**Інкапсуляція** - це сполучення даних і методів іх обробки в одному об’єкті. Mетоди об’єкту визначають спосіб змінення даних. Наприклад : немає необхідності знати спосіб отримання звуку - через мунштук для духових, або сруни для струнних. Код реалізації метода не важливий, це **принцип закриття інформаціі**. Спосіб добуття звуку заключений всередині інструмента. **Принцип „закриття об’єкту”** - оточення не може випадково змінити цей об’єкт.

**Поліморфізм** - це одий і той же метод, виконується по різному для різних об’єктів, при необхідності конкретного об’єкта можна перевизначити методи, враховуючи особливості даного об’єкта.

**Лабораторна робота №2-3.**

**Тема:** Створення простого додатка. Знайомство із системою

об'єктно-орієнтованого візуального проектування Delphi.

**Мета:**Ознайомиться з ООВП на прикладі створення додатка.

**Хід роботи:**

*I****. Конструювання форми*** *:*

1. Запустити Delphi ***{Меню Windows Пуск 🡪Програми}***, відкрити новий додаток ***{File 🡪 New Application 🡪 «No»}.*** На екрані відкрилася порожня нова **Forma** (Form1).

2. На Form розмістити кнопку **Tbutton** **{**виділити на сторінці **Standard** палітри компонентів піктограму кнопки одним клацанням миші, вибрати місце на полі **Form1** і клацнути **M** **🡪** з'явиться кнопка, ім'я якої за замовчуванням – **«Button1»}.**

3. На Form розмістити мітку **Label** **{**Аналогічно: **Standard** **🡪** піктограма **Label** **🡪** клацнути на формі **🡪** мітка **Label1}.**

4. Розмістити кнопку під міткою {перетаскуючи мишею}, зменшити розмір вікна {мишею}.

5. Перейменувати кнопку на **«Пуск»,** а мітку на **«Перший додаток»** **{**виділити на Form1 кнопку, перейти в Інспектор Об'єктів (якщо його немає на екрані – F11 чи **File** **🡪** **View** **🡪** **Object Inspector**), змінити властивість **Caption** (напис, заголовок)**}**.

6. Напис на мітці зробити жирним шрифтом {виділити мітку, у вікні Інспектор Об'єктів розкрити подвійним клацанням миші властивість **Font** (шрифт) **🡪** аналогічно – властивість **Style** (стиль) **🡪** встановити в **True** властивість **fsBold** (жирний)}.

7. Стерти текст у властивості Caption мітки, щоб він не висвічувався, поки користувач не натисне **«Пуск».** Текст на мітці повинний з'являтися тільки після натискання кнопки.

***II. Кодування*** ( розробка підпрограми обробки події OnClick ).

8. *Виділити кнопку на* ***Form1 🡪 Інспектор Об'єктів 🡪*** *відкрити**сторінку подій* ***Events 🡪*** *подвійне клацання мишею у вікні праворуч від імені події* ***OnClick 🡪*** *перейшли у вікно Редактора Коду з текстом процедури (тобто оброблювач події):*

***Procedure TForm1.Button1Click (Sender: TObject);***

Клас форм ім'я компонента ім'я події (без префікса On)

## **Begin**

***Label1.Caption: =’Перший додаток’;***

# Мітка Заголовок

**End** **;**

9.У вікні Редактора Коду вбудоване вікно Провідника (Дослідника) Коду. Його можна закрити.

10. Властивiсть **Caption** компонента **Label1** змінена на текст “Пер-ший додаток” за допомогою метода - підпрограми обробки події - **Procedure TForm1.Button1Click**.

11. Якщо немає помилок Error і зауважень Warning, компілюємо додаток і виконуємо його ***{File 🡪 Run 🡪 Run чи F9}.***

**Контрольні питання.**

1. Типи вікон Delphi.

2. Основні компоненти палітри.

3. Формування та конструювання форми.

4. Основні об'єкти форми.

5. Обробка події (Редактор Коду).

6. Провідник Коду.

7. Знавець Коду.

Захист програми паролем

Програміст може показати діалогове вікно з запитом пароля і блокувати виклик методу **Application.Run**, якщо введений користувачем пароль невірний. У наступному прикладі, у проекті використовуються дві форми: стандартна форма **InputQuery** і звичайна головна форма **Form1**. Форма **InputQuery** створюється при звертанні до однойменної функції, визначеної в модулі **Dialogs**. Вона являє собою невелике діалогове вікно з однорядковим текстовим полем **Tedit** і двома кнопками – **OK** і **Cancel**. У вікні користувач повинний ввести пароль (Delphi) і натиснути клавішу **Enter**.

**program Password;**

**uses**

**Forms,**

**Dialogs,** // У цьому модулі визначена функція InputQuery

**Until in ‘Unit1.pas’ {Form1};**

{$R \*.RES}

**var**

**Passwrd: String;**

**Begin** // Запитуємо пароль:

**if InputQuery (‘Вікно введення пароля ’ , ‘ Введіть пароль: ‘ , Passwrd)**

**then**

**if Passwrd = ‘Delphi’ then**

**begin** // Все в порядку, пароль вірний

**Application.CreateForm (Tform1, Form1);**

**Application.Run;**

**end else**

**ShowMessage ( ‘** Пароль не вірний! **’ ):**

**end.**

**Лабораторна робота N 4-5**

**Тема:** **Вивчення палітри компонентів**.

**Мета:** Розробити додаток для виконання арифметичних операцій.

**Хід роботи:**

**I. *1****.* Відкрити новий додаток.

***2.*** Перенести на форму зі сторінки бібліотеки **Standard** компоненти :

|  |  |  |  |
| --- | --- | --- | --- |
| ***Form1*** | \_ |  | X |
| **Label1**  **Edit1**  **Label2**  **Edit2**  **Label3**  **Panel1**  ***Button1*** | | | |

три мітки **Label** для написів у

послідовному порядку ;

два вікна редагування **Edit**;

панель **Panel**;

кнопку **Button**;

***3.*** Замінити надписи в мітках {властивість

***Caption***} на наступні:

Наприклад: “ число1 “,” число2 “,”результат”,

надпис кнопки – “Розрахунок”.

***4.*** Задати для міток жирний шрифт , а для кнопки – курсив.

***5.*** Очистити властивість ***Caption*** у панелі.

***6***. Для властивості ***Text***(текст) вікон редагування задати «1»- початкове зна-

чення тексту .

***7.*** Змінити вид панелі:

основне поле - ***Bevellnner =* *BvRaised*** («опуклий»).

рамка – ***Bevelouter = BvLowered*** (»утоплений»)

***8.*** У редакторі коду написати обробку події – клацання кнопки OnClick

Panel1.Caption:= Edit1.Text + ’ \* ‘ + Edit2.Text + ’ = ’+

FloatToStr (StrToFloat (Edit1.text) \* StrToFloat (Edit2.text));

***9.*** Зберегти додаток {створити свій підкаталог (як папку Windows) :

***{ File 🡪 Save all*** чи швидкими кнопками головного вікна}.

***10****.* Зкомпілювати і виконати розроблений проект (додаток).

**I I.****Розробити проект згідно варіанта завдання для розрахунку:**

1.Площі трикутника за сторонами А та В і кутом між ними α : S= ½ A\*B\*Sinα

2.Площі прямокутного трикутника за двома його катетами А та В : S=½ A\*B

3.Площі рівностороннього трикутника за довжиною його сторони : S=√3/4 a2.

4.Площі трикутника за довжинами його сторін А, В та С за формулою Герона

S= √P∙ (P-A) ∙ (P-B) ∙ (P-C), де Р=½(А+В+С)

5.Площі трапеції за двома основами А та В і висотою Н : S=(А+В)\*Н.

6.Довжини кола по заданому радіусу R : L=2πR.

7.Площі кругу по заданому радіусі R : S= πR2

8.Периметра Р і площі S прямокутника за його сторонами А та В

9.Сторону трикутника С за двома іншими сторонам А та В і кутом між ними :

C= А2+B2-2A\*B\*Cos β

10.Площі ромба за його діагоналями Н1 і Н2 : S=H1\*H2

**Контрольні питання :**

1.Компоненти Edit і Panel, їх властивості.

2.Візуальні властивості «Вид».

3.Функції перетворення типів FloatToStr , StrToFloat (IntToStr, StrToInt).

4.Компіляція і виконання проекта (додатка).

5.Розширення файлів Delphi.

Лабораторна робота № 6-7

**Тема:** **Розробка додатків з використанням меню.**

**Мета:** освоїти розробку додатків з використанням меню.

**Хід роботи:**

***I****.****Створення головного меню.***

**1.**Відкрити новий додаток (проект).

**2.**Активізувати FORMу , змінити значення властивості ***Caption*** на МMenu ,а **Name-** на Mform.

**3.**Зберегти проект **{*File 🡪 Save Project As 🡪*** з'явиться діалогове вікно ***Save Unit1 As*** 🡪 створюємо

нову папку , вибравши осмислену назву проекту (наприклад ЛР\_6\_7).Розкриваємо створену папку

у поле ***Ім'я файлу🡪*** дати ім'я програмного модуля***🡪*** ***Зберегти***. Відкриється діалогове вікно

***Save Project As.***(зберегти проект). У поле ***Ім'я файлу🡪і***м'я файлу проекту(LR 6-7).

Розширення файлу модуля -.pas, файл проекту -.dpr**}**

**4.**На формі в лівому верхньому кутку розмістити компонент ***TMainMenu* з *Standard*** (це не візуальний

компонент, тобто його піктограма не відповідає явному виду, а миша показує його наявність.

Настроювання його ведеться за допомогою спеціального редактора **Menu Designer**, що викликається

Подвійним клацанням на об'єкті MainMenu1).

**5.**Створити пункт головного меню **File {** в **Menu Designer** змінити властивість ***Caption*** на ***&File*** і ,таким

чином визначили клавішу швидкого виклику (Мнемонічного доступу), тобто <ALT>+<F>**}**.

**6.**Внести підпункти меню **Save** і **eХit** **{**змінюючи ***Caption*** на ***&Save*** та ***e&xit* }**.

**7.**Вибрати клавішні еквіваленту **F2 і F3 {**властивість ***ShortCut*** ***🡪*** вибрати відповідні F? чи

набрати їх у віконці **}**.

**8.**Анологічно створити пункт гол. меню **Help**(праворуч від File) і його підпункти **?Info(F1)**

та **Exit.**

**9.** Вставити між **?** та **Exit** розділову лінію **{**у властивості ***Caption*** у першій позиції набрати “дефіс – “}.

**10**.Відзначити пункт **Save** символом **√ .**

**11**.Розробити код обробки подій, що виникають при виборі пунктів меню, наприклад – Exit.

У редакторі меню подвійним клацанням миші на пункті **Exit** визиваємо редактор коду

***Procedure TMForm.Exit1Click(Sender:Tobject);***

**Begin**

**If CloseQuery then Close; {чи Close}**

**End;**

**12**.Зкомпілювати додаток **{F9}** і перевірити функції головного меню.

***I I.Створення Контекстного (Спливаючого) меню.***

**1**.На формі розмістити не візуальний компонент **TРopUpMenu.**

**2**.Аналогічно пунктам 5-7(першої частина LR) створити спливаюче меню.

**3**.Після створення спл. меню в редакторі Menu закрити вікно редактора Menu([x]) 🡪 клацнути на формі

(для скасування вибору всіх компонентів) .🡪 у списку вікна, що *розкривається,* ObjectInspector буде

**MForm:TMForm**. Властивості ***PopUpMenu*** форми надати значення ***PopUpMenu1.***

**4**.Аналогічно пункту 11 (I ч. LR) обробити пункт спливаючого меню ***«?Info»*** для інформаційного

діалогового вікна повідомлень.

Використати процедуру **MessageDlg,** що має 4 аргументи:

1 – симв. рядок з «переходом рядка» (код #13) і «поверненням каретки» (код #10) -для виведення

тексту у декілька рядків;

2 – **mtInformation** – вибір стилю діалогового вікна ;

3 – **[mbOк]** – множина кнопок у вікні;

4 – **0** – ознака контекстної довідки;

Текст процедури обробки події :

***Procedure TMForm.Info1Click (Sender:Tobject);***

**Begin**

**MessageDlg ('Меню розробив(ла)'#13#10'студент(ка) гр. № ФИО', mtInformation,[mbOk],0);**

**End;**

**5**.Зкомпілювати додаток і перевірити його роботу вибором пункту мишею ,клавішами еквіваленту та мнемонічного коду.

**Контрольні питання:**

**1**.Візуальні та невізуальні компоненти. **2**.Етапи збереження проекту.

**3**.Створення головного ( контекстного ) меню. **4**.Клавіші еквіваленту.

**5**.Мнемонічний код. **6.**Інформаційне діалогове вікно повідомлень.

**Тема : Сторінка системних компонентів**

**Сторінка System** : містить компоненти, що дозволяють мати доступ до програмного та апаратного забезпечення ПК.

Наприклад: Tіmer, Audіo - vіdіoplayer, вікно для малювання Paіnt - Box**.**

Компоненти Tіmer використовується для запуску процедур і функцій по обробці подій у зазначений інтервал часу, для контролю часу, тобто для виконання дій у реальному режимі часу. Це не візуальний компонент.

Має властивості:

- Enable (True, False) – використовується для генератора повідомлень.

- Іnterval – задається цілим числом у мілісекундах.

Має подію OnTіmer**.**

**Сторінка Win32 :** містить 32-бітні компоненти (після Delphi-2) загального призначення, що дозволяє розробку додатків у стилі Win98, 2000, NT.

Наприклад: календар, можливість корегування дат і часу, створення структури дерева, відтворення кліпів і т.д.

Візуальний компонент TdateTimePicker – поле введення, корегування дати і часу .

Має властивості :

Kind два значення:

dtk Date – корегування чи відображення дати.

dtk Time – корегування часу.

Властивість Data Mode дозволяє змінювати зовнішній вигляд календаря для дати, при цьому у властивостях Kind повинне бути значення dtkDate.

Значення dmComboBox – це поле зі списком (називається спадаючий чи календар, що випадає).

Значення dmUpDown – це коректування дати у вигляді лічильника.

Властивість Time - корегування часу.

Візуальний компонент TMonthCalendar – дозволяє швидко корегувати дату, готовий до роботи відразу після розміщення, ліворуч – колонка – номер тижня, використовуються кнопки переміщення, повернення на поточну дату з лічильником на сьогодні.

Лабораторна робота №8-9

**Тема: *Розробка інтерфейсу користувача з використанням компонентів SYSTEM і WIN32****.*

**Ціль:** засвоїти інтерфейс користувача з використанням компонентів сторінок SYSTEM. і WIN32.

**Хід роботи:**

***I.Створення цифрового годинника.***

1.Відкрити новий додаток (проект).

2.Вибрати форму *{клацнувши на ній}*, змінити Caption- на FORMLR8, Name- на FLR8.

3.Видалити кнопки мінімізації і максимізації вікна {*на властивості BorderIcons 🡪 розкрити [+] 🡪 встановити biSystemMenu – у TRUE, biMinimize і biMaximize- у FALSE*

Результат буде видний на формі лише після компіляції}.

4.Розмістити в центрі форми мітку *{ компонент LABEL сторінки STANDART}.*

Для резервування місця під розташовувані цифри змінити :

Caption-00:00:00 AM; Name-TimeLable1.

5.У властивості *Font* об'єкта *TimeLabel* вибрати колір, шрифт, розмір, стиль для годин; наприклад – жирний курсив, розмір 24 і т.д.

6.Перенести на форму зі сторінки SYSTEM компонент *Timer* (його ім'я за замовчуванням Timer1).

7.Обробити подію *OnTimer* {*вибрати об'єкт Timer1 і клацнути на закладці Events у вікні Інспектора Об'єктів*.

Викликати процедуру (метод) обробки подвійним клацанням миші на значенні праворуч від події:

**Procedure TFLR8.Timer1Timer(Sender: Tobject);**

**Begin**

**TimeLable1.Caption:=TimeToStr(Time);**

**End;** Цей код конвертує (переводить) поточний час у рядок String }.

8.Зкомпілювати додаток і запустити його. Вікно форми з ім'ям FormLR8 – без кнопок максимізації і мінімізації.

9.Вийти в Delphi на форму FormLR8.

***II. Маніпуляція з кольорами форми – мерехтіння кольору форми.***

1.Вибрати об'єкт Timer1 на формі.

2.Установити властивість Interval = 100

3.Код обробки події OnTimer доповнити процедурою (методом):

**FLR8.Color := trunc ( $02ffffff \* random );**

Ім'я (Name) форми властивість кольору

4.Зкомпелювати і запустити проект.

***III. Робота з календарем. Виведення повідомлення.***

1.Розмістити на формі кнопку, змінити заголовок Caption на «Час», ім'я Name – Button1.

2.Розмістити на формі зі сторінки WIN32 візуальний компонент *TDateTimePicker.*

3.Встановити режим відображення дати з розкритим календарем (поле зі списком):

властивість DateMode = dmComboBox; властивість Kind = dtkDate.

4.Встановити режим у вигляді лічильника:

властивість DateMode = dmUpDown.

5.Обробити подію натинення кнопки «Час» для виведення в інформаційному діалоговому вікні повідомлень поточного часу:

**MessageDlg('Поточний час**

**+ TimeToStr(DateTimePicker1.Time),mtInformation, [mbOK],0);**

6.Впорядкувати розміщення об'єктів на формі {*вибрати всі об'єкти форми: меню Edit ( Select all};*

*{Впорядкувати об'єкти : Edit→ Align → діалогове вікно - по горизонталі → Center (центрувати)*

*по вертикалі → SpaceEqual (рівна відстань)}*

7.Зкомпілювати і запустити додаток.

8.Зберегти модуль і проект.

**Контрольні питання.**

1.Призначення компонентів сторінки SYSTEM.

2.Компонент Timer, його властивості і події.

3.Призначення компонентів сторінки WIN32.

4.Властивості і значення компонента TDateTimePicker.

5.Властивості і значення компонента TMonthCalendar

6.Корегування дати і часу.

7.Впорядкуванння розміщення об'єктів на формі.

8.Видалення кнопок максимізації та мінімізіції на формі.

***Директиви компілятора***

Використовуються для зміни режимів функціонування (роботи) компілятора, впливають на ефективність виконання програми.

Розташовують директиви в коді програми безпосередньо перед текстом. Кожна директива заключається в фігурні дужки і має ознаку „$”. В одному рядку можна вказати декілька директив через кому: { $R+, $I+, $I\_ Funct1.pas }

**Т*ипи директив*:**

**1. перемикачі -** вкл **/** відкл режим компілятора („+” / ”-„ , ON / OFF) . { $R+, $I+ }

Можуть бути локальними і глобальними.

**2. з параметрами** - конкретне значення. Наприклад: ім’я файла. { $I\_ Funct1.pas }

1. **умовної компіляціі** - дозволяють в залежності від умов компілірувати або ні окреміфрагменти копій.

Для перевірки встановлених директив за замовчуванням -- функційні клавіші Ctrl + О або О (Options Compiler ).

Налагодити сторінку компілятора можна в діалоговому вікні Project Options установкою відповідних індикаторів.

1. {$R} – перевірка допустимих значень індексів масивів .

Дозволяє перевірити діапазон цілочисельних значень індексів, автоматично контролює

результати введення і виведення (за замовчуванням R- ).

1. {$І}- перевірка помилок введення /виведення, автоматично контролює результати введення - виведення (за замовчуванням І+).

Аналізується код помилки процедурою І / О RESULT

0 – Відсутність помилки;

2 – Файл не знайдений;

3 – Помилка в імені файла;

4 – Забагато відкритих файлів: збільшити кількість відкритих файлів в налаштуванні;

5 – Файл ушкоджений;

100 – Кінець файла;

101 – Диск переповнений;

106 – Помилка введення при роботі з файлами;

3. {$I\_ім’я файла} – підключає на етапі компіляції розроблені раніше файли;

4. {$Q} – переповнення при цілочисельних операціях;

5. {$Р} – розширення параметрів процедур і функцій у вигляді рядків;

6. {$С+} – перевірка тверджень.

**Лабораторна робота №10**

***Тема:*** ***”Консольний додаток”.***

***Ціль***: У режимі консольного додатка відпрацювати основний модуль задачі курсового проекту.

***Послідовність виконання:***

**1.**Запустити Delphi; відкрити новий додаток *{File🡪 New Application}.*

**2.** Закрити наступні вікна:

форми Form1;

інспектора об'єктів Object inspector;

модуль додатку Unit1.pas( без збереження ).

🡪на екрані лише головне вікно Delphi.

**3.** З меню Project🡪 View Source (перегляд)🡪 з'являється вікно головної

процедури додатка, назва Project1.dpr.

**4.** Набираємо текст (код) програми на Паскалі.

**5.** Перший рядок🡪{$Apptype Console} – директива компілятора.

**6.** У тексті програми перед (end.) повинeн бути оператор Readln;

зкомпілювати код програми меню *{Project🡪 Compile}* і запустити в роботу

*{Run🡨🡪F9}.*

**7.** На екрані – вікно Dos.

Контрольні запитання :

1. Директиви компілятора.
2. Класифікація директив.

**Лабораторна робота № 11**

**Тема: „Налагодження інтегрованого середовища розробника”**

**Хід роботи:**

1. **Відминити - відновити вивід на екран палітри компонентів**

*{View→ToolBars→Component Palette} {Вид→Панель инструментов→Палитра компонентов}.*

1. **Відминити - відновити вивід на екран панелі інструментів** *{View→ToolBars→Standard} {Вид→Панель инструментов→Стандарт}.*
2. **Додати кнопку на панелі інструментів,** Comands *{View→ToolBars→Customize(налагодження)→ вiкно зі сторінками Categories (мелю) та Comands (команди). Вибрати меню файл і команду мишкою перетягнути кнопку на панель. Видалити кнопку з панелі}.*
3. **Відминити - відновити вивід на екран інспектора об'єктів** *{View→ToolBars→Object Inspeclor(Інспектор об’єктів)(F11)}.*
4. **Додати елемент в меню** *{Tools(інструменти)→Configure Tools(конфігурація інструментів)→ у вікні діалогової панелі Tool Options натиснути Add→Tools Properties(властивосиі інструментів), натиснути Brows→ зі списку вибрати задачу, відкрити її; у вікні Title дати заголовок - наприклад Matric→Ok y вікні Tools Properties→Close→* *задача включена в Tools}.*
5. **Змінити порядок задач у списку Tools** *{у списці Tool Option вибрати потрібну задачу і перемістити її на початок списку кнопками Move Up}.*

7.**Видалити задачу зі списка Tools** *{вибрати задачу зі списку Tools→Delete}.*

1. **Перемістити сторінку палітри компонентів System перед Standart** *{натиснути правою кнопкою миші на палітрі компонентів →* *висвітиться контексне меню Properties (властивості) → діалогове вікно Palette Properties з двома колонками →* *у списці сторінок Page вибираємо System та переміщуємо її}.*
2. **Додати нову сторінку в палітру з назвою "37 "** *{в діалоговому вікні Palette Properties натиснути на Add і вводимо нове ім 'я сторінки GR37 }.*
3. **Видалення сторінки** *{в діалоговому вікні Palette Properties натиснути на сторінці GR37\_(mo6mo вибираємо її) та на кнопку Delete}.*
4. **Доповнити в Знавець Коду (Code Insight) шаблон для оператору цикла з**

**параметром - 1**:

For |: = downto do // Ctrl+J - виклик знавця коду

*{Tools→Edition Option→Code Insight→Code Templates→ натиснути Add у вікні для виводу нового шаблону shotcut+name в його короткий опис Description набираємо текст*

*шаблону, встановлюємо " │ " перед " :* =" *→* *зберігаємо та виходимо }.*

1. **Видалити шаблон із Знавця Коду** *{ Tools→Editer Option→Code Insight→* *знайти шаблон і.Del}.*
2. **Змінити затримку автоматичного спрацьовування Code Insight на максимум** (Delay) *{Tools→Edition Option→Code Insight→Delay→ повзунок на max = 1.5 sec→ Ok}.*

**Контрольні питання:**

1. Налагодження палітри компонентів.
2. Налагодження панелі інструментів.
3. Налагодження меню Tools.
4. Корегування знавця коду.

**Тема: Введення й обробка елементів масиву**

***І***Визначити конкретне значення елементів масиву ми можемо з використанням компонентів сторінки **Standard Edit** ( вікно редагування), при цьому на вихідній формі необхідно передбачити таку кількість віконець **Edit**, скільки елементів має масив.

**Edit .Text** - інформація символьного типу, для обробки числових значень необхідне перетворення типів:

* **StrToFloat, StrToInt** – перетворення символьного в числове;
* **FloatToStr, IntToStr** – зворотне перетворення для висновку результату.

При такому заданні масиву необхідно заздалегідь визначати і чітко фіксувати кількість елементів ( розмірність масиву).

***ІІ***При великій розмірності масиву незручно використовувати величезну кількість вікон редагування. Масив можна вводити по рядкам в таблицю елементів, де кожне число знаходитися в окремому клітинці. При цьому елементи розділяються пробілом. Компонент **StringGrid** знаходитися на сторінці **Add itional** .

Властивості компонента **String Grid:**

1. ***Name*** – ім'я, що використовується в програмі для звернення до об'єкта;
2. ***ColCount*** – кількість стовпчиків (стовпців);
3. ***RowCount*** – кількість рядків;
4. ***Cells –*** ім'я масиву ,відповідного таблиці.
5. ***FixedCols*** – кількість фіксованих зліва стовпчиків;
6. ***FixedRows*** – кількість фіксованих зверху рядків;
7. ***Options.goEditing*** – можливість редагування вмісту клітинок таблиці;
8. ***Options.goTab*** – можливість переходу до наступного елементу клавішею “**Tab**”;
9. ***DefaultColWidth*** – визначає ширину стовпчиків таблиці;
10. ***DefaultRowWidth*** – висота рядків таблиці.
11. ***OptionsGoAlwaysShowEditor*** – при значенні **False**, після набору тексту необхідно натиснути F2, чи клацнути миша при значенні **True** – редагування виконується автоматично.

***ІІІ.*** Використання компонента **Мемо** дозволяє вводити в масив елементи символьного типу у вигляді рядків тексту.

Властивості **Memo**:

**Name --** ім'я компонента, що використовується в тексті програми;

**Lines** – номер рядка, номерація починається з нуля (номер задається в «[ ]»);

**Text** – символьна інформація, що визначає значення елементів масиву.

*IV.* Введення даних у масив з використанням списку **List Box** (Standard).

Властивості **ListBox :**

**Items** – елементи списку;

**ItemIndex** – номер обраного елемента списку ( на початку роботи «0»).

Список може бути сформований під час створення форми додатка, чи динамічно під час роботи програми.

У вікні **Object Inspector** вибрати властивості **Items** і клацнути на кнопці....., відкривається діалогове вікно редактора списку рядків ( StringList Editor). Кожен елемент набирати з нового рядка і завершувати **«Enter».** Після введення всіх елементів списку натиснути «**ОК».**

Процедура обробки події OnKeyPress

***( перевірка коректності введення цілих чисел)***

**Procedure Tform1.StringGridKeyPress (Sender:Tobject; Var Key : char );**

**Begin**

# Case Key of

**#8,’0’..’9’ : ; //** *Цифри і клавіша <Back Space>*

**#13*;*** *// Клавіша <Enter>*

**If StringGrid1. Col *<* StringGrid1.ColCount – 1**

**Then StringGrid . co l : = StringGrid1.col + 1**

**Else key : = Chr(0);***// iнші символи заборонені*

**End;**

**End;**

### Перевірка коректності введення дійсних чисел

**Procedure Tform1.StringGridKeyPress (Sender:Tobject; Var Key : char );**

**Begin**

## Case Key of

**#8,’0’..’9’ : ; //** *Цифри і клавіша <Back Space>*

**#13; *//*** *Клавіша <Enter****>***

**If StringGrid1. Col < StringGrid1.ColCount – 1**

**Then StringGrid . co l : = StringGrid1.col + 1;**

**‘. ‘ , ’,’ : //** Роздільник цілої і дробової частин числа

Begin

**If Key <> DecimalSeparator then**

**Key : = DecimalSeparator ; //***Замінимо роздільник на припустимий*

**If Pos ( StringGrid1. cells [ StringGrid . Col , 0] , DecemalSeparator ) <> 0**

**Then Key : = chr ( 0); //** *Заборона введення повторного роздільника*

**End;**

**‘ – ‘ : //** *мінус можна ввести тільки першим символом*

*// тобто коли комірка порожня*

**If Length ( StringGrid1.Cells [ StringGrid1 . Col , 0 ] ) <> 0**

**then Key : = Chr ( 0 )**

**else */ /*** *Інші* символи заборонені

**key : = Chr (0);**

**end;** *{ end case}*

**end;** *{ end proc}*

**Лабораторна робота № 12.**

**Тема**: Обробка двовимірного масиву.

**Мета**: Освоїти методи обробки масиву.

**Постановка задачі**: Підрахувати середній бал студентів за підсумками сесії. За запитом виводити середній бал по всій групі.

**Хід роботи**:

1. Відкрити новий додаток. Дати заголовок **Form1.Caption** ― 'Підсумки сесії':

Розмістити на формі:

1. Зі сторінки **Standard** 🡪 мітку **Label1**; властивість **Caption** ― 'Підсумки сесії по групі 37\_';
2. Зі сторінки **Additional** 🡪 таблицю рядків StringGrid1; встановити властивості:

**ColCount** ― 5 *{Кількість стовпців}*;

**RowCount** ― 5 *{Кількість рядків}*;

**FixedCols** ― 0 *{Кількість фіксованих стовпців}*;

**FixedRows** ― 1 *{Кількість фіксованих рядків}*;

**Name** ― StringGrid1 ( або SG1 );

**Options.goEditing** ― True *{Редагування клітинок таблиці}*;



**Options.goTab** ― True;

1. Зі сторінки **Standard** 🡪 прапорець **CheckBox**; встановити його властивість **Caption** ― 'Середній бал по групі';
2. Зі сторінки **Standard** 🡪 кнопку **Button1**; встановити властивість **Caption** ― 'Пуск';
3. Зі сторінки **Standard** для виведення середнього бала по групі 🡪 **Label2** (чи **Panel1**); **Caption** ― ' ' (очистити).
4. Для початкового програмного заповнення заголовка таблиці обробити подію активації форми:

**Procedure TForm1.FormActivate(Sender: TObject);**

begin

StringGrid1.Cells[0,0]:=' Ф И О';

StringGrid1.Cells[1,0]:=' Програм.';

StringGrid1.Cells[2,0]:=' Мат.Ан.';

StringGrid1.Cells[3,0]:=' Диф.Ур.';

StringGrid1.Cells[4,0]:=' Ср.Бал';

end;

1. Основну обробку і введення елементів масиву (при заповненні таблиці) оформити як процедуру обробки події натискання клавіші "Пуск":

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

var

a:array [1..5,1..10] of integer; {робочий масив}

S, {середній бал студента}

SR:real; {середній бал по групі}

i, {лічильник стовпців}

j:integer; {лічильник рядків}

begin

SR:=0;

For j:=1 to SG1.RowCount-1 do {цикл по рядках}

begin

S:=0;

For і:=1 to 3 do {цикл по стовпцях}

begin

a[i,j]:=StrToInt(SG1.Cells[i,j]);

S:=S+a[i,j];

end; {For і}

S:=S/3; {підрахунок середнього балу студента}

SG1.Cells[4,j] := FloatToStr(s);

SR:=SR+S;

end; {For j}

{аналіз значення прапорця - обчислення середнього бала по групі}

If CheckBox1.State=cbChecked then

begin

SR:=SR/(SG1.RowCount-1);

Label2.Caption:=FloatToStr(SR);

end; {Then}

end; {Proc}

1. Виконати перевірку коректності введення даних процедурою **OnKeyPressed**. У випадку помилки вивести повідомлення

"Дані не входять у діапазон [1..5]" і повторити введення.

1. Перевірити, чи введені дані в клітинку, якщо ні ― повідомлення на екран і корегування таблиці.

Заповнити початковий масив випадковими числами ( RANDOM ).

**Контрольні питання**:

1. Компонент **StringGrid**, його властивості і події.
2. Компонент **CheckBox**, його властивості і події.
3. Методи введення й обробки елементів масиву.
4. Перевірка коректності введення даних.

Лабораторна робота №13

**Тема: ”Анкета студента”**

**Мета:** Використання зображення (фото, картинки), створення ехе-файла.

**Постановка задачі:**

Розробити додаток із зображеннями, що перекриваються.

**Хід роботи:**

***I.***  ***Конструювання форми:***

**1.** Відкрити новий додаток. Дати заголовок Form1.Caption – ‘Анкета студента’

**2.** Вибрати піктограму (іконку) для запуску задачі ***{Icon→***

***→ Load*** *(завантажити)****→*** *вибираємо зі списку файлів* ***\*.ico →***

***переглядаємо → відкрити → ОК*** *→ іконка в системному рядку форми змінилася на обрану****}***

**3.** За бажанням змінити колір, фон, стиль, розмір форми.

**4.** Забрати кнопки системного меню форми ( ) ***{Borderconst → biSystemMenu; biMinimize; biMaximize установити False }***

**На формі розмістити:**

**5.** Зі сторінки **Standard** – мітку **Label1**; властивість Caption – **'Студент КТРЕ'**

**6.** Зі сторінки **Standard** – поле Мемо (чи вікно Edit, чи Label) – для введення анкетних даних.

**7.** Розмістити 2 кнопки Button (з Standard)

*Установити:*

**Caption.Button1** – ‘портретнe фото’

**Caption.Button2** – ‘художнє фото’

**8.** Для коректного виходу з задачі (див.п.4) використовувати кнопку **BitBtn** зі сторінки палітри компонентів **Additional**. Установити властивість **Kind**(сорт) – вид кнопки на **bkClose *→***на кнопці з'явилася піктограма і напис **Close**.

**9.**На кнопці виведемо спливаючу підказку – “натяк”:

***{властивість Hint –*** *(це текст підказки, що з'являється при наведенні покажчика миші)* ***–*** *'вихід із задачі’****;*** *властивість* ***ShowHint –*** *(показувати/непоказувати підказку)* ***→*** *установити* ***True}***

**10.**Розмістити на формі об'єкт **Image1** («картинка») зі сторінки **Additional**.

**11.**Уставити своє чи фото малюнок: ***{Image1→ Picture →*** *вікно редактора* ***Picture → Load →*** *вибираємо файл із* ***\*.jpg; \*.bmp; \*.ico; \*.wmf → OK →*** *картинка у вікні****}***

**12.**Привести розмір зображення до заданих розмірів об'єкта **Image**: **{***властивість* ***Stretch→ TRUE*}**

**13.**Аналогічно п.п.10-13 вибрати 2-і фото (картинку), наклавши її цілком на першу картинку і сполучивши розміри.

***II. Розробка кодів (процедур):***

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

begin

Image1.Visible:=true; *{портретне фото візуальне}*

Image2.Visible:=false; *{художнє фото не візуальне}*

button1.Visible:=false; {*кнопка1 не візуальна*}

button2.Visible:=true; {*кнопка2 візуальна*}

end;

**procedure** TForm1.Button2Click(Sender: TObject);

begin

Image2.Visible:=true; *{портретне фото візуальне}*

Image1.Visible:=false; *{художнє фото не візуальне}*

button2.Visible:=false; {*кнопка2 не візуальна*}

button1.Visible:=true; {*кнопка1 візуальна*}

end;

**procedure** TForm1.BitBtn1Click(Sender: TObject);

begin

close;

end;

***III. Зберегти розроблений додаток:***

***{гол. меню → File → Save all*** *(зберегти усі) чи* ***Shift+Ctrl+S}***

чи для іншої папки:

***{гол.меню → File → Save as…(SaveUnitas…)→шлях →****ім'я* ***Anketa.pas.***

***гол.меню → Save Project as…→шлях→*** *ім'я* ***Foto.dpr}***

***У в а г а!!!***

Файли проекту і тексту програми повинні мати різні імена.

***IV.*** ***Створити ехе-файл проекту:***

1. **{гол.меню *→ Project→ Build all*** *(сконструювати все)***}**
2. Закрити *Delphi*:

Запустити задачу як ехе-файл через піктограму

**КОНТРОЛЬНІ ПИТАННЯ :**

1. Властивості компонента Image.
2. Властивості і події компонента BitBtn.
3. Створення ехе-файла.
4. Вибір піктограми задачі
5. Спливаюча підказка.

**Лабораторна робота № 14**

**Тема : Створення довідкової системи додатка**

**Мета : Навчитися створювати довідкову систему проекту**

**Складається з двох етапів:**

# ***Створення файлу документа***

***Компіляції файлу довідки.***

***I******Етап***

Файл документа створюється будь-яким текстовим процесором (напр. *Microsoft Word*) і зберігається як **rtf** файл.

**1.** Набираємо текст розділів довідки.

**2.** Заголовки розділів оформляємо одним зі стилів «*Заголовок*».

**3.** Текст кожного розділу довідки повинен бути на окремій сторінці документа, тобто закінчуватися «*розривом сторінки*».

**4.** Позначаємо заголовки розділів виносками. Універсальна виноска для ідентифікатора теми, по якому на цю виноску можуть посилатися інші теми - #. Її потрібно ставити перед кожним розділом довідки; ***{Вставка → посилання → виноска → інша → # у поле введення → Ok}****.* У нижній частині вікна редагуваннярозкривається вікно введення тексту виноски, де вводимо ідентифікатор уставленої виноски; ***{# розділ1 # razd 2}.***

**5.** Перехід до інших розділів довідки по виділеному слову (ключу): ключове слово в загальному тексті **rtf** файлу підкреслюємо подвійною ***лінією {Формат → Шрифт → Підкреслення → Подвійне → Ок застосовуємо до виділеного ключового слова}****.* За ключовим словом у тексті rtf файлу набираємо ідентифікатор посилання, куди необхідний перехід у довідці і «ховаємо» цей набраний ***ідентифікатор {Формат → Шрифт → Схований → Ок}****.* При цьому опція «подвійне підкреслення» повинна бити відключена.

**6.** Зразок фрагменту **rtf** – файлу:

***# sozd*** *Створення довідкової системи {Заголовок1} складається з двох етапів:*

*I – Створення файлу документів razdel1*

*II – Компіляція файлу довідки razdel2*

***# razdel1*** *Файл документа можна створити засобами процесора MS Word*

*{ текст розділу }*

***# razdel2*** *Компілюємо файл довідки програмою MS Help Workshop*

*{ текст розділу }*

|  |  |
| --- | --- |
| *Усі виноски* | *▼* |

|  |
| --- |
| *Закрити* |

***# sozd***

***# razdel1***

***# razdel2***

***II*** ***Компіляція файлу довідки .***

**1.** Запускаємо MS Help Workshop, його шлях:

## C:\Program Files\Borland\Delphi\Help\Tools\Hcw.exe

**2.** У головному вікні програми:

File → New → Help Project → **відкривається вікно** Project File Name → **вибираємо папку з проектом і з файлом** rtf → **у полі** Ім'я файлу **вводимо ім'я файлу довідкової системи** → Зберегти.

**3.** Відкрилося вікно проекту довідкової системи .

**4.** Додаємо файл – документ довідкової системи.

*Кнопка* ***Files →*** *у вікні* ***Topic File******→ Add →*** *вікно* ***«Відкриття файлу» →*** *вибираємо потрібний* ***rtf – файл.***

**5.** Задаємо характеристики вікна довідкової системи:

*Кнопка* ***Windows → Windows Properties →*** *у поле* ***Create a window named →*** *слово* ***“Main” –*** *основний тип створюваного вікна* ***→ Ok →*** *повернення у вікно* ***Windows Properties.***

**6.** *Вкладка* ***General →*** *поле* ***Title bar text → вводимо заголовок головного вікна (напр. Довідка).***

**7.** Можна вибрати колір фону заголовка і тексту довідки

**{** Вкладка **Color →** вибір **Change → Ok}**

**8.** Призначимо числові значення ідентифікаторам розділів довідки: ***{****у вікні проекту довідкової системи натиснути кнопку* ***Map →*** *сховалося діалогове вікно Map* ***→ Add →*** *у поле* ***Topic*** *ID вводимо ідентифікатор посилання* ***→*** *у поле* ***Mapped numeric Value*** *– відповідне значення* ***}.***

**9.** Компілюємо файл довідки:

*Меню* ***File → Compile → Compile a Help File*** *установлюємо прапорець для автоматичного показу створеної довідки* ***→ Compile.***

**10.** Створено файл довідки ***ім'я.hlp.***

**11.** Підключаємо розроблений файл довідки в проект *Delphi*, зв'язуючи довідку з клавішею F1:

***Object Inspector Form1 → Help File → ім'я.hlp;***

***→Help Context → 1.***

**12**. Для виклику довідки натисканням на кнопку – процедура (Button чи пункт меню)

**Procedure TForm1.Button1Click(Sender:TObject);**

### Begin

**Winhelp (Form1.Handle,’ім'я.hlp’, Help\_Context,1);**

**End;**

{властивість Handle вікна доступно лише під час роботи роботи програми}/

**13.** Розроблений проект *Delphi*, файл – **rtf** і файл довідки**.hlp** повинні знаходиться в одній папці.

**Література до курсу алгоритмічних мов та програмування:**

**Література**

1. Архангельский А.Я. Программирование в Delphi 6.-

М.: ЗАО “Изд-80 БИНОМ”,2002г.-1120с.:ил.

2. Бобровский С. Delphi 5:Учебный курс-С-П:Питер, 2001.-640 c.: ил.

1. Глинский Я.Н., Анохин В. Е., Ряжская В. А. Turbo Pascal 7.0 и Delphi.

Учебное пособие .-СПб: ООО “ДиаСофтЮП” 2001.-208 c.

4. Культин Н.Б. Программирование на Obgect PASCAL в Delphi 5.

-СПБ.: БХВ-С.-П.,2000,-464 с.:ил.

5. Фаронов В. В. Delphi5. Учебный курс.-М.:’’Нолидж”, 2001.-608c.,ил.

6. Черняхівський В.В.Delphi-4: сучасна технологія візуального програмування.-

Львів: БаК, 1999. – 196 с.

7. Фаронов В.В. Turbo Pascal 7.0 Научный курс. Учебное пособие,

изд.- 7-е М.: „ Нолидж” , 2001, -576 с., ил.

|  |
| --- |
| МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  КІСІТКНЕУ імені Вадима Гетьмана  **Лабораторна робота №12**  з предмету: **"Алгоритмічні мови та програмування".**  **Тема:**” Робота з двовимірним масивом”  Фах 5.080.405.027 ПЗ |

**Мета**: Освоїти методи обробки масиву.

**Постановка задачі**: Порахувати середній бал студентів за підсумками сесії. По запиту розрахувати і вивести середній бал по всій групі.

**Хід роботи**:

1. Відкрити новий додаток. Дати заголовок **Form1.Caption** ― 'Підсумки сесії':

Розмістити на формі:

1. Зі сторінки **Standard** 🡪 мітку **Label1**; властивість **Caption** ― 'Підсумки сесії по групі 37\_';
2. Зі сторінки **Additional** 🡪 таблицю рядків StringGrid1; встановити властивості:

**ColCount** ― 5 *{Кількість стовпців}*;

**RowCount** ― 5 *{Кількість рядків}*;

**FixedCols** ― 0 *{Кількість фіксованих стовпців}*;

**FixedRows** ― 1 *{Кількість фіксованих рядків}*;

**Name** ― StringGrid1;

**Options.goEditing** ― True *{Редагування осередків таблиці}*;

**Options.goTab** ― True;

1. Зі сторінки **Standard** 🡪 прапорець **CheckBox**; установити його властивість **Caption** ― 'Середній бал по групі';
2. Зі сторінки **Standard** 🡪 кнопку **Button1**; установити властивість **Caption** ― 'Пуск';
3. Зі сторінки **Standard** для висновку середнього бала по групі 🡪 **Label2** (чи **Panel1**); **Caption** ― ' ' (очистити).
4. Для початкового програмного заповнення заголовка таблиці обробити подія активації форми:

**procedure** TForm1.FormActivate(Sender: TObject);

**begin**

StringGrid1.Cells[0,0] := 'Ф И О';

StringGrid1.Cells[1,0] := 'Програм.';

StringGrid1.Cells[2,0] := 'Мат.Ан.';

StringGrid1.Cells[3,0] := 'Диф.Ур.';

StringGrid1.Cells[4,0] := 'Ср.Бал';

**end**;

1. Основну обробку і введення елементів масиву (при заповненні таблиці) оформляти у виді процедури обробки події натискання клавіші "Пуск":

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

**var**

a : **array** [1..5,1..10] **of** integer; *{робочий масив}*

S, *{середній бал студента}*

SR : real; *{середній бал по групі}*

i, *{лічильник стовпців}*

j : integer; *{лічильник рядків}*

**begin**

SR := 0;

**For** j := 1 **to** StringGrid1.RowCount-1 **do** *{цикл по рядках}*

**begin**

S := 0;

**For** і := 1 **to** 3 **do** {цикл по стовпцях}

**begin**

a[i,j] := StrToInt(StringGrid1.Cells[i,j]);

S := S + a[i,j];

**end**; {For і}

S := S/3; {ви числение середнього бала студента}

SR := SR+S;

**end**; {For j}

*{аналіз значення прапорця - обчислення середнього бала по групі}*

If CheckBox1.State=cbChecked then

**begin**

SR := SR/(StringGrid1.RowCount-1);

Label2.Caption := FloatToStr(SR);

**end**; *{Then}*

**end**; *{Proc}*

IV Виконати перевірку коректності введення даних процедурою **OnKeyPressed**. У випадку помилки вивести повідомлення "Дані не входять у діапазон [1..5]" і повторити введення.

V Перевірити, чи введеноі дані в осередок, якщо немає ― повідомлення на екран і коректування таблиці.

**Контрольні питання**:

1. Компонент **StringGrid**, його властивості і події.
2. Компонент **CheckBox**, його властивості і події.
3. Методи введення й обробки елементів масиву.
4. Перевірка коректності введення даних.

Протокол роботи:

**unit** lr12;

**interface**

**uses**

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,

Dialogs, StdCtrls, Grids;

**type**

TForm1 = **class**(TForm)

Label1: TLabel;

StringGrid1: TStringGrid;

CheckBox1: TCheckBox;

Button1: TButton;

Label2: TLabel;

Button2: TButton;

**procedure** FormActivate(Sender: TObject);

**procedure** Button2Click(Sender: TObject);

**procedure** Button1Click(Sender: TObject);

**procedure** StringGrid1KeyPress(Sender: TObject; var Key: Char

private

*{ Private declarations }*

public

*{ Public declarations }*

**end**;

**var**

Form1: TForm1;

implementation

*{$R \*.dfm}*

**procedure** TForm1.FormActivate(Sender: TObject);

**begin**

StringGrid1.Cells[0,0] := 'ФИО';

StringGrid1.Cells[1,0] := 'Програм.';

StringGrid1.Cells[2,0] := 'Мат.ан.';

StringGrid1.Cells[5,0] := ‘Стипендия';

end;

**procedure** TForm1.Button2Click(Sender: TObject); *{Процедура виходу з програми}*

**begin**

form1.Close;

**end**;

**procedure** TForm1.Button1Click(Sender: TObject);{ Процедура обробки Button }

**var**

S, Sr : real;

i, j : integer;**begin**

Sr := 0;

**for** j := 1 to StringGrid1.RowCount **do**

**begin**

S := 0;

**for** i := 1 to 3 do

**begin** **if** StringGrid1.cells[i,j]<>''

**then** S := S+StrToInt(StringGrid1.Cells[i,j])

**else** StringGrid1.Cells[i,j] := '0';

**end**; //end i

S := S/3;

StringGrid1.Cells[4,j] := FloatToStrF(s,ffFixed,4,2);

**if** s=5 **then** StringGrid1.Cells[5,j] := '40.00'

**else** **if** (S<5)and(s>=4) then StringGrid1.Cells[5,j] := '32.50'

**else** StringGrid1.Cells[5,j] := 'не получает';

sr := Sr+s;

**end**; *//end i*

**if** CheckBox1.State=cbChecked then

**begin**

Sr := sr/(stringGrid1.RowCount-1);

label2.Caption := 'Средний балл ='+FloatToStrF(Sr,ffFixed,4,2);

**end**; *//end check*

**end**; *//end proc*

**procedure** TForm1.StringGrid1KeyPress(Sender: TObject; var Key: Char);

**begin**

**case** key **of**

#8, '2'..'5':;

**else** **begin**

MessageDlg('Оценка по 5-ой смистеме(2..5)',mtError,[mbOk,MbCancel],0);

Key := chr(0);

**end**;

**end**;

**end**;

**end**. *//end program*

**Форми**:

