

Візуальне програмування

Виконайте ОДНУ (на Ваш вибір) лабораторну роботу у системі "візуального програмування"

Наприклад:

[Lazarus \(Delphi\)](#)

або використовуючи [Visual Studio](#):

[Windows Forms C#](#)

[WPF \(Windows Presentation Foundation\) на платформі .NET підсистема для побудови графічних інтерфейсів \(C#\)](#)

Ви також можете використовувати будь-яку іншу систему "візуального програмування"

Лабораторні роботи

(методичний опис дано для системи [Lazarus](#))

[Лабораторна робота №1](#)

[Використання компонентів в інтегрованому середовищі Lazarus](#)

[Лабораторна робота №2](#)

[Створення найпростішого Windows-програми](#)

[Моя перша програма](#)

[Лабораторна робота №3](#)

[Найпростіша математична програма](#)

[Лабораторна робота №4](#)

[Проста обробка подій](#)

[Лабораторна робота №5](#)

[Компоненти вибору та налаштування параметрів](#)

[Лабораторна робота №6](#)

[Цифровий годинник](#)

[Лабораторна робота №7](#)

[Використання списків](#)

[Лабораторна робота №8](#)

[Програма-жарт](#)

Лабораторна робота №1

Використання компонентів в інтегрованому середовищі Lazarus

Цілі роботи: знайомство із середовищем Lazarus

0. Встановіть програмне забезпечення, якщо необхідно.

1. Запуск Lazarus

Щоб запустити Lazarus, виберіть Пуск/Програми/Lazarus/Lazarus.

2. Основні елементи екрану

Знайдіть такі основні елементи екрану Lazarus:

a) Вікно програми;

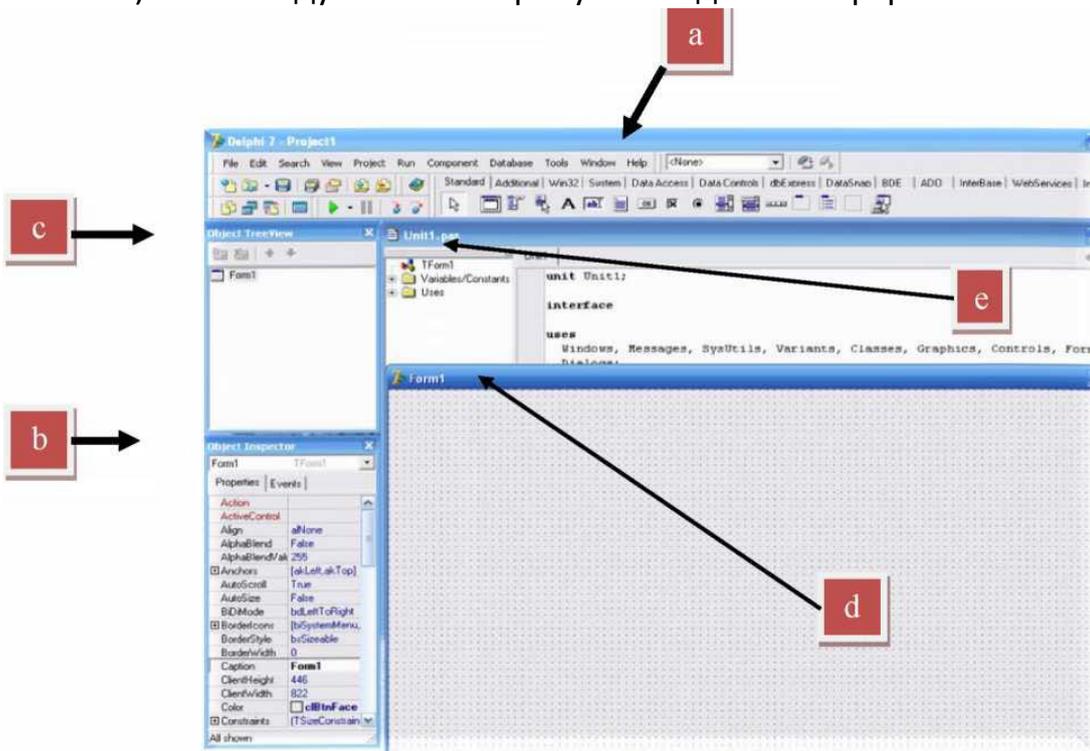
- Рядок меню;
- панелі інструментів;
- Палітру компонентів (Visual Component Library);

b) Вікно Інспектора Об'єктів (Object Inspector);

c) вікно дерева перегляду об'єктів (Object TreeView);

d) Вікно форми (Form1);

e) Вікно модуля може перебувати під вікном форми



3. Закриття проекту

Закрийте проект, створений за промовчанням, не зберігаючи його за допомогою File/Close AP.

4. Створення нового проекту

Створіть новий проект, вибравши File/New/Application.

У програмному кодї модуля, що з'явився, розберіть призначення основних його

частин, використовуючи коментарі:

```
unit Unit1;           //Це ім'я створеного модуля -Unit1
interface           //Початок інтерфейсної частини
uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs;
type
  TForm1 = class (TForm) // Опис класу форми
  private
    {Private declarations}
  public
    {Public declarations}
  end;
var
  Form1:TForm1;      //Об'єкт форми
implementation      //Початок реалізації
{$R *,DFM}
end.
```

5. Збереження всього проекту

Створений проект збережіть в окремій папці за допомогою File/Save АП.

6. Склад проекту

При створенні проекту та його збереженні Lazarus створює різні файли за своїм призначенням. За допомогою провідника відкрийте папку, в яку ви зберегли проект і подивіться на створені файли.

7. **Перемикання між вікном модуля і відповідним вікном форми**Міняти місцями вікно модуля та відповідної форми зручно за допомогою клавіші F12. Клацніть на вікні форми. Натисніть клавішу F12. Відкриється вікно модуля. Ще раз натисніть F12. Відкриється вікно форми.

8. Робота з Менеджером Проекту

Закрийте вікно форми. Для відкриття вікна форми, додавання або видалення модулів використовується Менеджер проекту. Викличте вікно форми за допомогою View/Project Manager.

9. Запуск програми виконання

Запустіть програму на виконання кнопкою F9 або Run/Run або кнопка |> на панелі інструментів. Знайдіть на панелі завдань значок програми, що виконується.

Завершити виконання програми можна клацанням по відповідній кнопці на рядку заголовка вікна або комбінацією клавіш Alt + F4.

10. Зміна піктограми для програми

Змініть значок програми, використовуючи Project/Options/Application/LoadIcon. Виберіть будь-яку з наявних піктограм у папці Піктограми, яка знаходиться в тій же

директорії, що й лабораторна робота.

Запустіть програму на виконання та проконтролюйте, що ваша програма відображається вибраним значком.

11. Встановлення компонентів на форму

Будь-який компонент можна помістити на форму різними способами:

1 спосіб: вибрати компонент у палітрі, клацнути у потрібному місці форми мишею.

2 спосіб: вибрати компонент у палітрі, розтягнути мишею на формі область для об'єкта.

3 спосіб: подвійне клацання мишею по компоненту на палітрі призводить до створення обраного об'єкта на формі, причому об'єкт з'явиться в центрі форми.

Помістіть на форму трьома різними способами 3 компоненти Button.

12. Видалення компонентів з форми

Видалити компоненти з форми можна двома способами:

1 спосіб: виділити компонент на формі, видалити клавішею Delete.

2 спосіб: викликати контекстне меню над компонентом, у якому вибрати Edit/Delete. Видалити двома різними способами 2 компоненти Button з форми.

13. Робота з властивостями компонента

Інспектор Об'єктів (Object Inspector) відображає властивості та події компонента, обраного на даний момент.

Для зміни кольору фону компонента Label необхідно виділити його, на закладці Properties Інспектора Об'єктів у властивості Color змінити значення константи кольору.

Змінити заголовок у компонента Labell на 'Приклад'.

Редагувати ім'я компонента Buttonl на Кнопка.

Змініть колір фону, розмір, стиль, колір шрифту компонентів Editl і Memol.

14. Робота з групою компонентів

Якщо в кількох компонентів потрібно змінити загальні для всієї групи властивості, необхідно виділити ці компоненти. При виділенні групи на сторінці Properties Інспектора Об'єктів буде видно лише загальні властивості - шрифт, колір тощо. А індивідуальні – ім'я, заголовок компонента зникнуть. Виділити можна 2 способами:

1 спосіб: якщо компоненти розставлені безпосередньо формі поруч друг з одним, то виділення групи досить обвести курсором рамку навколо них.

2 спосіб: виділити потрібні компоненти можна, натиснувши та не відпускаючи клавішу Shift.

Виділіть на формі компоненти Button та змініть колір фону компонентів на блакитний.

15. Закрийте проект.

Лабораторна робота №2

2.1 Створення найпростішого Windows-програми

Цілі роботи: Створення найпростішого Windows-програми із заданим заголовком вікна та кольором форми

1. Створити папку для збереження розроблених програм
2. Запустити Lazarus
3. Змінити заголовок вікна форми з Form1 на Привіт: у вікні інспектора об'єктів (Object Inspector) встановити для властивості Caption значення Привіт
4. Змінити колір форми зі стандартного на інший: у вікні інспектора об'єктів встановити значення Color clAqua для властивості Color.
5. Виконати додаток:
 - 5.1. Запустити програму - меню Run, Run або F9, або кнопка на панелі інструментів.
 - 5.2. Змінити розмір вікна.
 - 5.3. Поекспериментувати зі стандартними кнопками мінімізації та максимізації вікна.
 - 5.4. Закінчити роботу програми, закривши її вікно.
6. Зберегти форму та проект на диску:

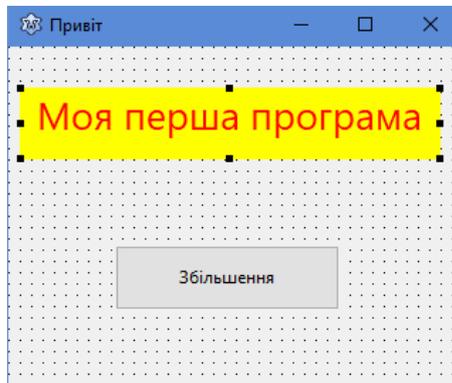
Меню File, Save All, встановити свою папку, створити нову папку (з ім'ям Лабораторна робота №1), відкрити її, ввести ім'я проекту

2.2 Моя перша програма

Цілі роботи: Створення програми Windows, яка містить текст "Моя перша програма!" та кнопки, що дозволяють змінювати розмір шрифту та рухати текст

1. Розмістити об'єкт Label у вікно форми Form1:
2. Перемістити об'єкт Label1 на бажане місце у формі.
3. Змінити властивості об'єкту Label 1:
У вікні інспектора об'єктів (Object Inspector) встановити такі значення властивостей об'єкта:

Об'єкт	Властивість	Значення
Label1	Caption	Моя перша програма!
	Font	12 p., червоний
	Alignment	TaCenter
	Color	Жовтий (Yellow)
	AutoSize	False



4. Виконати програму: меню Run/Run або F9.
5. Зберегти форму та проект на диску: МенюFile, Save A11, встановити свою папку, запровадити ім'я Лабораторна робота №2.
6. Розмістити об'єктButton (командна кнопка) у вікні Form1. Він за промовчанням отримає ім'я Button1. Змініть його розміри.
7. Встановити властивістьCaption об'єкта Button1 до значення "Збільшення".
8. Написати код для подіїClick на об'єкті Button1: Двічі клацнути по об'єкту Button1 у формі Між словами Begin та End написати наступний код: `Label1.Font.Size := Label1.Font.Size +2;`
9. Виконати програму. Зверніть увагу на те, що відбувається при натисканні кнопки з написом "Збільшення".
10. Зберегти форму та проект на диску: МенюFile, Save.
11. Створити об'єкт "командна кнопка", щоб зменшити розмір шрифту в тексті.
12. Створити об'єкт "командна кнопка", щоб рухати текст.
Код: `Label1.Left :=Label1.Left+ 10;`
`Label1.Top :=Label1.Top+ 10;`
13. Створити об'єкт "командна кнопка", щоб зробити текст невидимим.
Код: `Label1.visible:= false;`
14. Створити об'єкт "командна кнопка" для виходу з програми. Код:`Close;`
15. Зберегти форму та проект.

Лабораторна робота №3

Найпростіша математична програма

Цілі роботи: Метою роботи є практичне освоєння методології та принципів створення базових стандартних елементів інтерфейсу Windows-програми серед візуального проектування.

У наведеному нижче проекті використовуємо наступний мінімальний набір компонентів.

Button – стандартна кнопка, зазвичай кнопка використовується для запуску дії, при цьому задіють лише метод OnEvent (реакція на натискання).

Властивість Default=True асоціює компонент, що вводиться, з кнопкою Enter,

Cancel = True - з кнопкою Esc.

Властивості Color для оформлення напису (Caption) у кнопки немає.

Амперсант, вміщений у тексті написи, вказує швидку Alt-клавішу запуску, наприклад, Caption=A&Ppend викликає спрацьовування кнопки при натисканні Alt-P.

Властивість ModalResult=true визначить обов'язковість натискання закриття дочірнього вікна.

Label - мітка, використовується як напис або область виведення інформації для читання. Як і для кнопки, для мітки можна визначити клавішу швидкого доступу, але вона запускатиме пов'язаний з міткою компонент (по FocusControl).

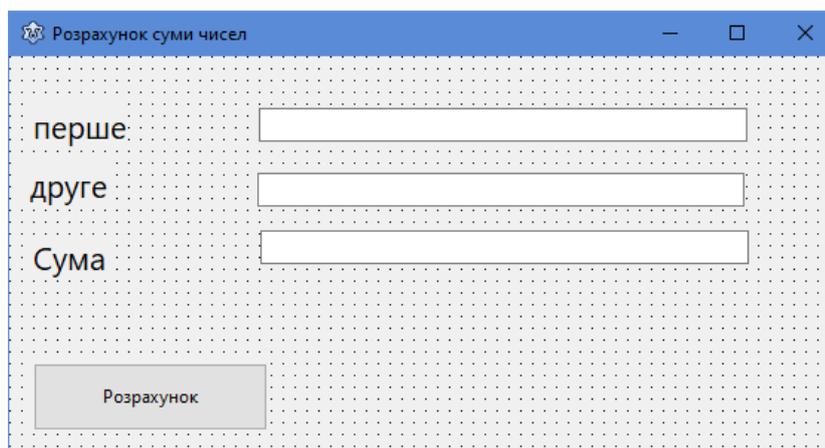
Властивість AutoSize=True визначить мінімізацію розміру мітки під текст напису, Aligment - центрування цього тексту,

WordWrap - можливість розташування тексту кілька рядків, Transparent - прозорість при накладення інші елементи.

Edit – рядок введення. Заголовка (Caption) цього компонента немає, але є властивість Text як вміст рядка. Цю властивість можна як зчитувати, і присвоювати (при необхідності з обмеженням довжини призначенням якості MaxLength).

При введенні конфіденційної інформації вказують символи, що відображаються (зазвичай "*"), при цьому потрібно перевизначити властивість PasswordChar, задаючи його відмінним від #0.

Складемо проект для підсумовування двох чисел, які вводяться з клавіатури.



При цьому на формі потрібно розмістити чотири написи (з властивостями Caption, що задаються) і п'ятий напис з порожнім Caption - для відображення суми. Визначити два рядки введення для сумованих чисел (проти позначок "перше" і "друге") та одну кнопку "Розрахунок" для запуску процедури підсумовування після введення чисел.

Після подвійного клацання на кнопці можна заповнити шаблон процедури реакції на натискання цієї кнопки (рамкою виділено текст, що вводиться).

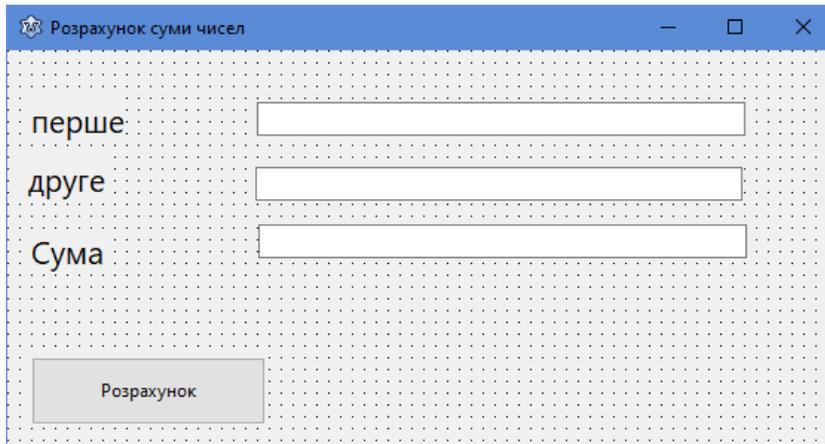
```
procedure TForm1.Button1Click(Sender: TObject);
var a, b, c: real;
s: string; code: integer;
begin
  {введення даних із полів редагування}
  val(edit1.text,a,code);
  val(edit2.text,b,code); c:=a+b;
  str(c:-10:4,s); {переведення числа в рядок}
  label5.Caption:=s;
end;
```

Лабораторна робота №4

Проста обробка подій

Цілі роботи: Метою роботи є практичне освоєння методології та принципів створення та оформлення елементів інтерфейсу Windows-програми.

Необхідні елементи оформлення проекту



Наведений у раб. № 3 варіант програми цілком працездатний. Але в подібних програмах обов'язкова вимога щодо їх оформлення - передбачити реакції на введення символів у полях редагування, наприклад, захист від введення букв або другої десяткової точки.

При натисканні Enter природно переносити курсор у наступне поле редагування або виконувати інші дії, якщо введення даних завершено.

У обробниках подій (закладка Events інспектора подій Lazarus) для полів введення визначимо методи OnKeyPress, задавши їм імена, наприклад, e1 та e2.

Потім після подвійного натискання заповнимо шаблони процедур.

```
procedure TForm1.e1(Sender: TObject; varKey: Char); begin
  {захист поля редагування на введення числа}
  case key of
    '0'..'9',chr(8);
    '!': if pos('.',edit1.text)>0 then key:=chr(0);
    '!': iflength( edit1.text)>0 then key:=chr(0);
  chr(13): edit2.SetFocus;
  else key:=chr(0);
  end;
end;
```

Друга процедура відрізняється від першої лише реакцією на натискання клавіші Enter

```
procedure TForm1.e2(Sender: TObject; varKey: Char);
begin
... edit2.text...
  chr(13): edit2.font.color:=clRed;
...
end;
```

Текст процедури TForm1.Button1Click бажано оформити як самостійну процедуру, наприклад,

```
procedure Summa(edit1,edit2: tEdit; label5: TLabel);
```

і викликати її як усередині TForm1.Button1Click, так і реакції на Enter в процедурі TForm1.e2, при цьому закінчення введення даних відразу запустить обчислення.

Введемо кнопку очищення полів введення та виведення результату для нового розрахунку. Заголовок кнопки визначимо як Caption = "нове", задаємо реакцію OnClick (подвійним клацанням на кнопці).

```
procedure TForm1.Button2Click(Sender: TObject);  
begin  
  {очищення полів введення}  
  edit1.text:="";  
  edit2.text:="";  
  label5.caption:="";  
  edit1.SetFocus;  
end;
```

Введемо кнопку виходу procedure TForm1.Button3Click(Sender: TObject);

```
Begin  
  
form1.close; { завершення програми }  
  
end;
```

Лабораторна робота №5

Компоненти вибору та налаштування параметрів

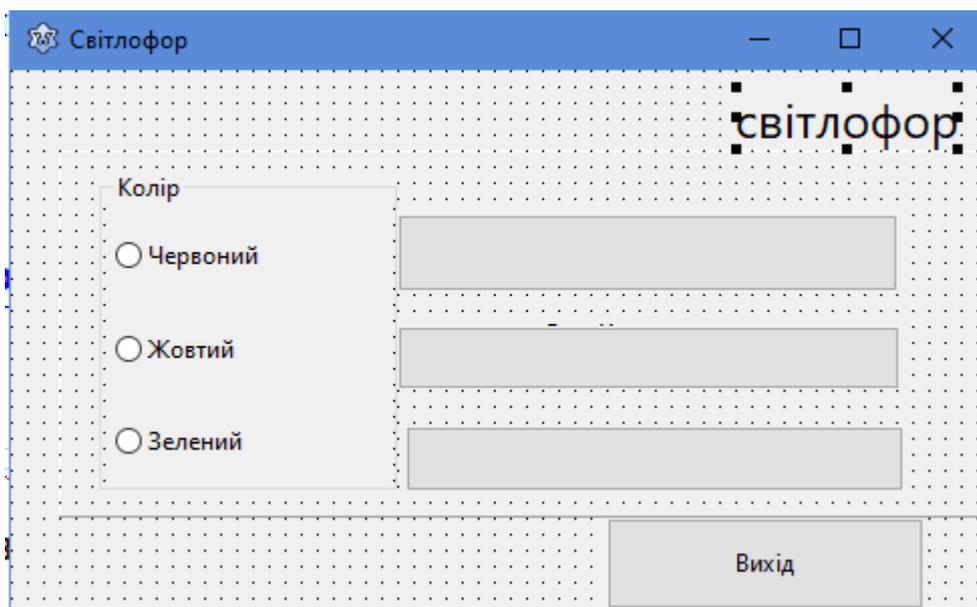
Цілі роботи: Створення Windows-програми, в якій при натисканні на радіо-кнопці з назвою кольору на світлофорі загоряється відповідний колір

Вибір та налаштування параметрів під час роботи з програмним додатком вважається стандартною частиною роботи користувача з будь-яким серйозним додатком. Це може бути як налаштування самої програми, так і визначення параметрів процесів і явищ, що відображаються або моделюються в додатку. Елементи інтерфейсу Windows-програми для основних операцій такої роботи зараз практично стандартизовані. Розглянемо створення цих елементів з прикладу роботи з компонентами бібліотеки VCL (Visual Component Library) серед Lazarus.

Базові елементи вибору та налаштування параметрів розташовані на сторінці Standard панелі компонентів Lazarus. У наведеному нижче проекті використовуємо наступний класичний набір компонентів:

- GroupBox - група, яка візуально та логічно поєднує набори компонентів, визначає порядок переміщення компонентами на формі (при натисканні клавіші TAB). При розміщенні групи новий компонент отримує властивості ParentColor, ParentShowHint, ParentFont, ParentCtl3D цієї групи. Властивості Left та Top згрупованих об'єктів визначаються по верхньому кутку групи, а не форми;
- RadioGroup – група для об'єктів RadioButton (див. нижче);
- RadioButton – перемикачі або радіокнопки, служать для вибору однієї можливості з набору взаємовиключних можливостей. Термін відображає схожість із набором кнопок вибору каналів радіоприймача. Ці кнопки зазвичай поєднують групою RadioGroup. Вибір кнопки відображає властивість Checked, властивість Alignment визначає положення пояснювального напису щодо кнопки;
- CheckBox – вимикач, виглядає як рядок тексту з віконцем для встановлення позначки про вибір. Вимикачі працюють незалежно, але їх зазвичай групують. При визначенні реакції на вибір можна використовувати подію OnClick, але зазвичай встановлюють як індикатор властивість State за трьома станами - cbChecked (є), cbUnchecked (ні), cbGrayed (невизначено) усередині програми. Для блокування ручної зміни цієї властивості потрібно встановити DragMode=Automatic.

Приклад проекту із вибором параметрів



1. Помістити компоненти Label, Panel, GroupBox, RadioButton (сторінка Standard) у форму.

2. Встановити такі властивості об'єктів, використовуючи Інспектор об'єктів:

Label 1	Caption	Світлофор
Panel1,2,3	Caption	
	Caption	Колір
RadioButton1	Caption	Червоний
RadioButton2	Caption	Жовтий
RadioButton3	Caption	Зелений

3. Записати код для процедури обробки подіїClick (клацання миші) на об'єкті RadioButton1:
- ```
procedure TForm1.RadioButton1Click(Sender: TObject);
begin
 Panel1.Color := clRed;
 Panel2.Color := clWhite;
 Panel3.Color := clWhite;
end;
```

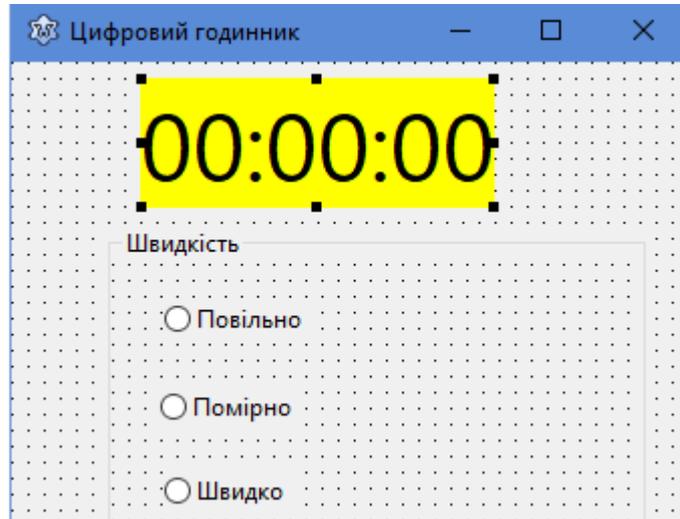
Самостійно записати код для процедур: TForm1.RadioButton2Click та TForm1.RadioButton3Click

4. Додати друк інформації "Стійте", "Увага", "Ідіть" на панелі з відповідним сигналом білим кольором шрифту жирним зображенням12п.

# Лабораторна робота №6

## Цифровий годинник

**Цілі роботи:** Створення Windows-програми, в якій працює цифровий годинник з різною швидкістю



1. Помістити компоненти Label (вкладка Standard) та Timer (System) у форму Form1.
2. Встановити такі властивості об'єктів

| Об'єкт  | Властивість | Значення |
|---------|-------------|----------|
| Form1   | Name        | Clock    |
| Label 1 | Caption     | 00:00:00 |
| Label 1 | Color       | clYellow |
| Label 1 | Font.Size   | 24       |
| Label 1 | Font.Color  | Червоний |

2. Записати код оновлення часу для процедури TClock.Timer1Timer: Label 1.Caption:=TimeToStr(Time);
4. Додавання кнопки регулювання швидкості оновлення часу.
  - 4.1. Додати до форми компоненти GroupBox та RadioButton:
  - 4.2. Встановити такі властивості об'єктів:

|              |         |           |
|--------------|---------|-----------|
| GroupBox1    | Caption | Швидкість |
| RadioButton1 | Caption | Повільно  |
| RadioButton2 | Caption | Помірно   |
| RadioButton3 | Caption | Швидко    |

- 4.3. Записати код для процедури TForm1.RadioButton3Click: Timer1.Interval := 1000; Самостійно записати код для процедур: TForm1.RadioButton1Click (3000) та TForm1.RadioButton2Click (2000)

# Лабораторна робота №7

## Використання списків

**Цілі роботи:** Створення програми Windows з типовим компонентом ListBox.

**Вивчити його можливості**

ListBox – звичайний список, цей компонент призначений для роботи з переліком текстових елементів (з обмеженням за кількістю до 5000 шт). Список можна створювати (у тому числі завантажувати як рядки з текстового файлу), перетворювати та вивантажувати у файл. Елементи списку можна вибрати за допомогою клавіатури або миші. Класичний приклад використання ListBox у середовищі Windows - вибір файлу зі списку пункті меню File/Open багатьох додатків.

Основна властивість списку - Items (масив рядків), вона аналогічна до властивості Lines для компонента Мемо. Індекс вибраного елемента списку зберігається у змінній ItemIndex. Методи Add, Delete, Insert використовуються для додавання, видалення та вставки рядків.

Властивість Sorted=True упорядковує список зростання коду символів рядків. ItemHeight – вертикальний розмір елементів, Columns – число колонок у списку, ExtendedSelect – можливість множинного вибору елементів (при утриманні Shift), при цьому для вибраних елементів властивість Selected [номер] дорівнює True. ComboBox – комбінований список, додатково до звичайного включає рядок введення. З кількох типів ComboBox найбільш популярний спадаючий вниз (dropdown combo box).

Створимо типовий проект із компонентом ListBox



На формі (малюнок) крім списку розмістимо ряд кнопок (або пунктів меню), а також два рядки введення Edit1, Edit2 та дві мітки Label1, Label2.

На вибір пунктів організуємо наступні операції зі списком:

Завантаження рядків із файлу, ім'я якого попередньо набирається у рядку введення (пункт "файл")

```
Listbox1.Sorted:=false;
```

```
Listbox1.Items.LoadFromFile(Edit1.Text);
```

Сортування списку (пункт "сортувати"):

```
Listbox1.Sorted:=true;
```

Запис списку файл, ім'я якого попередньо набирається в рядку введення (пункт "зберегти як"):

```
Listbox1.Items.SaveToFile(Edit2.Text);
```

```
MessageDlg('Створено файл'+Edit2.Text,mtInformation,[mbOK],0)
```

Завантаження списку екранних шрифтів (пункт "шрифт"):

```
Listbox1.Items:=Screen.Fonts;
```

Додавання випадкового числа до списку з дотриманням сортування, якщо воно задане (пункт "додати")

```
var s:string;
```

```
begin
```

```
str(random:10:8,s); {генерація випадкового числа}
```

```
ListBox1.Items.Add(s) end;
```

Додавання числа до потрібного місця списку (пункт "вставити")

```
var s:string;
```

```
begin
```

```
str(random:10:8,s); {генерація випадкового числа}
```

```
ListBox1.Items.Insert(ListBox1.ItemIndex,s);
```

```
end;
```

Видалення вибраного елемента списку (пункт "Видалити")

```
ListBox1.Items.Delete(ListBox1.ItemIndex)
```

Виведемо деякі характеристики вибраного елемента на мітках:

```
var code: integer; a: real;
```

```
begin
```

```
Label2.Caption:=ListBox1.Items[ListBox1.ItemIndex];
```

```
Val(Label2.Caption,a,code);
```

```
If code=0 then Label1.Caption:='число'
```

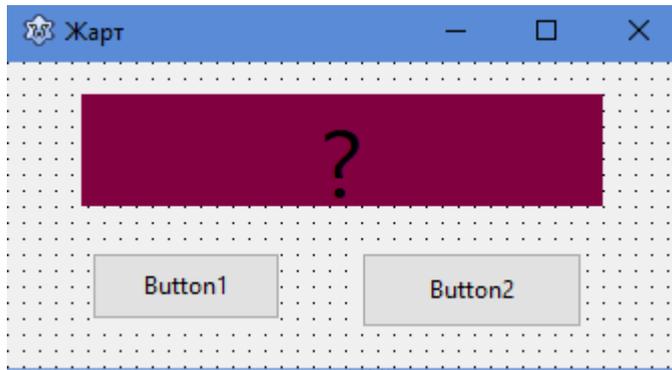
```
else Label1.Caption:='рядок';
```

```
end;
```

# Лабораторна робота №8

## Програма-жарт

**Цілі роботи:** Створення програми Windows, в якій змінюється розташування об'єктів.



1. Помістити компоненти Label та Button у форму відповідно до малюнку.

2. Встановити такі властивості об'єктів

| Об'єкт  | Властивість | Значення |
|---------|-------------|----------|
| Form1   | Caption     | Жарт     |
| Label 1 | Caption     | ?        |
| Label1  | Color       | clMaroon |
| Label1  | Font.Size   | 18       |
| Label1  | Font.Color  | Синій    |
| Label1  | Alignment   | taCenter |

3. Встановити властивість об'єкта Button2: DragMode dniAutomatic

4. Записати код для обробки події MouseMove на об'єкті Button2: procedure TForm1.Button2MouseMove(Sender: TObject; Shift: TShiftState; X, Y: Integer);  
begin  
    Button2.Left :=Button2.Left+10;  
    Button2.Top :=Button2.Top+10;  
end;

5. Записати код для обробки події Click на об'єкті Button1: procedure TForm1.Button1Click(Sender: TObject);  
begin  
    Label1.Caption := 'Ми були в цьому впевнені!';  
end;

5. Виконати програму.

6. Розвиток задачі:

Змінити програму так, щоб при підведенні курсору миші до кнопки Button2 кнопка зникала, а при відведенні курсору з'являлася.