

Лабораторна робота № 17 Створення проєктів в C++ Builder

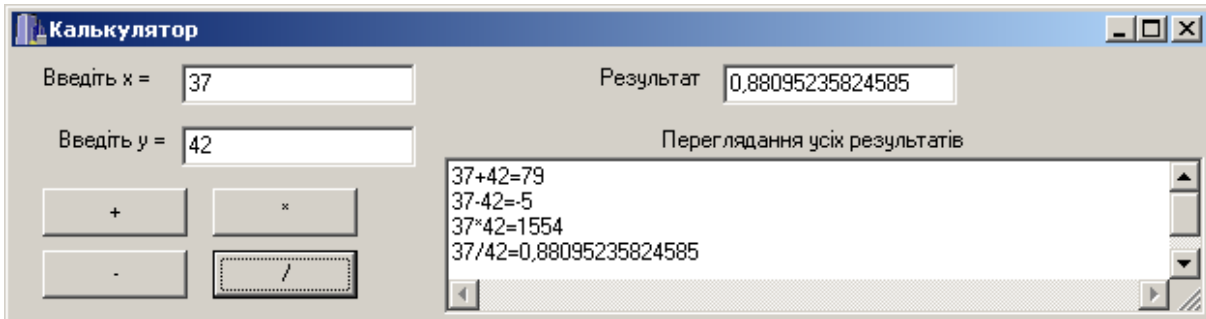
Мета роботи: освоїти принципи візуального програмування, вивчити основні функціональні можливості C++ Builder, навчитися працювати з головним візуальними компонентами C++ Builder.

Завдання 1. Простий калькулятор

1. Постановка задачі

Створити програму – прототип калькулятора – для обчислення простих арифметичних дій над двома числами.

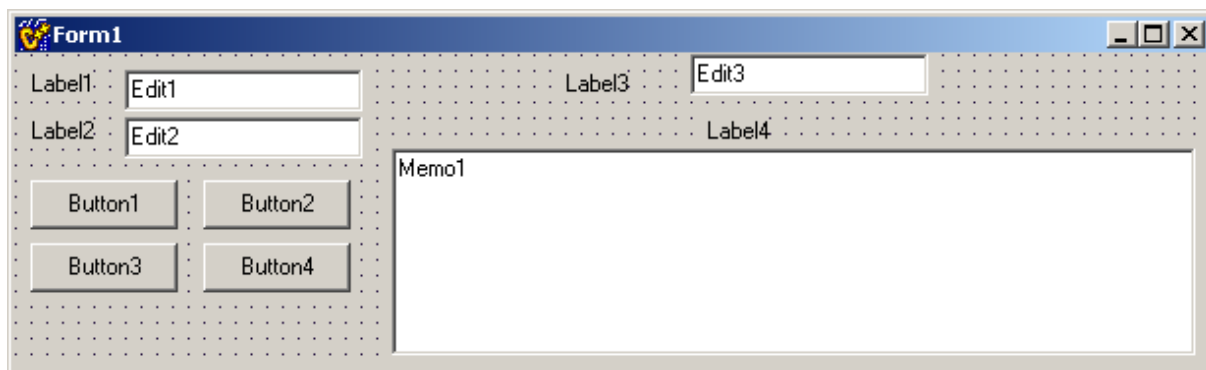
Робочий вигляд форми проєкту:



2. Створювання форми

На вікно форми треба встановити такі компоненти: три компоненти Edit, чотири надписи Label, чотири кнопки Button, один компонент Мемо. Усі ці компоненти розташовано на палітрі компонентів на вкладці «Standard».

Вигляд форми після розташування всіх компонентів:



Задамо властивості компонентів, наведені в табл. 1.

Таблиця 1 – Властивості компонентів

Компонент	Властивість	Значення
Form1	Caption	Калькулятор
Label 1	Caption	Введіть x =
Label 2	Caption	Введіть y =
Label 3	Caption	Результат
Label 4	Caption	Переглядання усіх результатів
Button 1	Caption	+
Button 2	Caption	-

Button 3	Caption	*
Button 4	Caption	/
Memo1	ScrollBars	ssBoth

3. Створювання програм

Задамо для командної кнопки Button1 шаблон для відгуку на подію OnClick (подія виникає при клацанні мишею по кнопці). Для цього можна виконати чотири кроки, подані у прикладі 1 чи просто *двічі* швидко клацнути по Button1. Внаслідок цього матимемо такий шаблон у вікні редактора коду:

```
void __fastcall TForm1::Button1Click(TObject *Sender)
{
}
```

Впишемо поміж операторних дужок такий програмний код:

```
float x, y, s;
x = StrToFloat(Edit1->Text);
y = StrToFloat(Edit2->Text); s = x + y;
Edit3->Text = FloatToStr(s);
Memo1->Lines->Add(FloatToStr(x)+" "+FloatToStr(y)+"="+FloatToStr(s));
```

Після цього слід знову перейти до форми проєкту (це можна здійснити за допомогою миші чи то клавіші **F12**) і повторити дії щодо створювання шаблонів і програмного коду для решти командних кнопок Button2, Button3 та Button4. Оскільки текст коду для всіх компонентів **Button** є вельми схожий, то доцільно використовувати копіювання.

Для очищення компонентів Edit1, Edit2, Edit3 та Memo1 при завантажуванні форми створимо шаблон події FormCreate подвійним клацанням по формі:

```
void __fastcall TForm1::FormCreate(TObject *Sender)
{
}
```

Наведемо повний текст програм у вікні редактора коду C++ Builder. Деякі фрагменти прокоментуємо (коментар (пояснювальний текст) не виконується компілятором і розміщується в один рядок після символів // чи в кілька рядків поміж символами /* */).

Текст програмного коду:

```
//
#include <vcl.h>
#pragma hdrstop #include "Unit1.h"
//
#pragma package(smart_init)
#pragma resource "*.dfm" TForm1 *Form1;
//
__fastcall TForm1::TForm1(TComponent* Owner)
: TForm(Owner)
{
}
//
void __fastcall TForm1::FormCreate(TObject *Sender)
{ // очищення компонентів Edit1, Edit2, Edit3 і Memo1 відповідно
Edit1->Clear(); Edit2->Clear(); Edit3->Clear(); Memo1->Clear();
}
//
```


```

void__fastcall TForm1::Button1Click(TObject *Sender) // Кнопка «+»
{ float x, y, s;
x = StrToFloat(Edit1->Text); y = StrToFloat(Edit2->Text); s = x + y;
Edit3->Text = FloatToStr(s);
Memo1->Lines->Add(FloatToStr(x)+" "+FloatToStr(y)+"="+FloatToStr(s));
}
//
void__fastcall TForm1::Button2Click(TObject *Sender) // Кнопка «-»
{ float x, y, s;
x = StrToFloat(Edit1->Text);
y = StrToFloat(Edit2->Text);
s = x - y;
Edit3->Text = FloatToStr(s);
Memo1->Lines ->Add( FloatToStr(x)+"-"+FloatToStr(y) +"="+FloatToStr(s));
}
//
void__fastcall TForm1::Button3Click(TObject *Sender) // Кнопка «*»
{ float x, y, s;
x = StrToFloat(Edit1->Text);
y = StrToFloat(Edit2->Text);
s = x * y;
Edit3->Text = FloatToStr(s);
Memo1->Lines ->Add( FloatToStr(x)+"*"+FloatToStr(y) +"="+FloatToStr(s));
}
//
void__fastcall TForm1::Button4Click(TObject *Sender) // Кнопка «/»
{ float y, x, s;
x = StrToFloat(Edit1->Text);
y = StrToFloat(Edit2->Text);
s = x / y;
Edit3->Text = FloatToStr(s);
Memo1->Lines ->Add( FloatToStr(x)"/"+FloatToStr(y)+"="+FloatToStr(s));
}

```

4. Запуск програми (проекту)

Для запуску проекту на виконання слід виконати команду меню **Run->Run** або натиснути на клавіатурі кнопку **F9**.

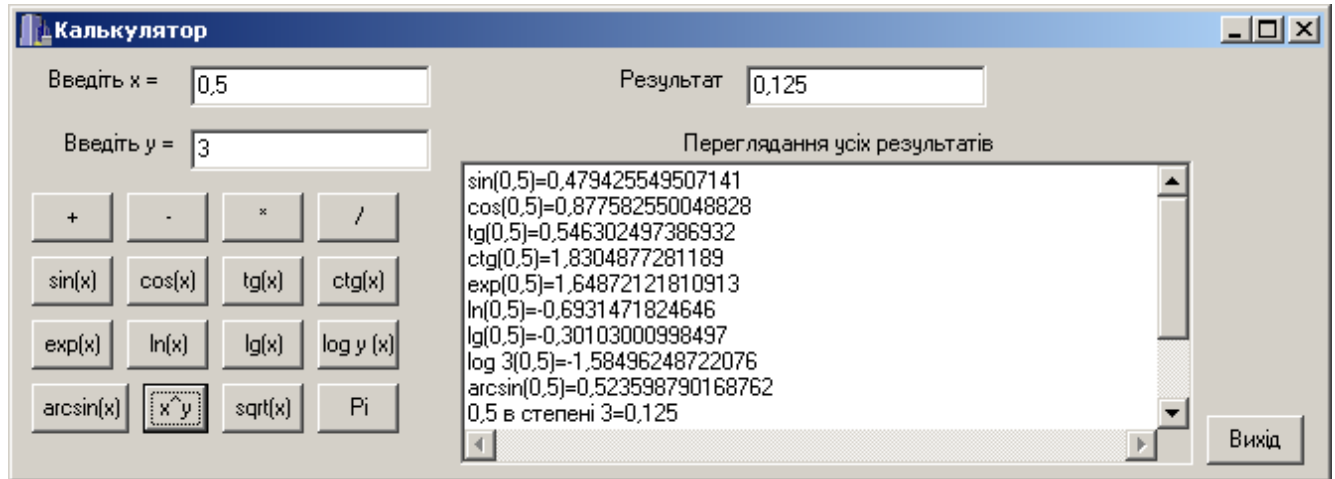
Увага! Перед запуском проекту його обов'язково слід зберегти в окремій теці – команда **File->SaveAll** ()

Завдання 2. Інженерний калькулятор

1. Постановка задачі

Створити програму – прототип калькулятора – для обчислення арифметичних дій та математичних функцій.

Робочий вигляд форми проекту:



2. Створення форми

Продовжуємо виконувати приклад №1, вдосконалимо його, додавши нові можливості обчислення математичних функцій. Додамо на форму ще 13 кнопок Button і змінимо їх надписи (властивість Label).

C++ Builder підтримує роботу з великою кількістю стандартних математичних функцій (див. додаток А). Ці функції містяться в системних математичних бібліотеках (окремих файлах) **math.h** та **Math.hpp**, які треба долучити директивою:

```
#include <ім`я бібліотеки>
```

Остаточний текст програм у вікні редактора коду, за винятком програм для чотирьох кнопок, які вже було наведено у завданні 1:

```
#include <vcl.h> #pragma hdrstop #include "Unit1.h"  
#include <math.h> // в цьому місці слід долучити математичні бібліотеки  
#include <Math.hpp>  
#pragma package(smart_init)  
#pragma resource "*.dfm"  
TForm1 *Form1;  
//  
__fastcall TForm1::TForm1(TComponent* Owner)  
: TForm(Owner)  
{  
}  
void __fastcall TForm1::Button5Click(TObject *Sender) // Кнопка "sin(x)"  
{ float x, s;  
x = StrToFloat(Edit1->Text);  
s = sin(x);  
Edit3->Text =FloatToStr(s);  
Memo1->Lines ->Add( "sin("+ FloatToStr(x) +"")="+ FloatToStr(s));
```

```

}
//
void __fastcall TForm1::Button6Click(TObject *Sender) // Кнопка "cos(x)"
{ float x, s; x=StrToFloat(Edit1->Text);
s=cos(x);
Edit3->Text = FloatToStr(s);
Memol->Lines->Add( "cos("+ FloatToStr(x) +")=" + FloatToStr(s));
}
//
void __fastcall TForm1::Button7Click(TObject *Sender) // Кнопка "tg(x)"
{ float x, s;
x = StrToFloat(Edit1->Text);
s = tan(x);
Edit3->Text = FloatToStr(s);
Memol->Lines->Add( "tg("+ FloatToStr(x) +")=" + FloatToStr(s));
}
//
void __fastcall TForm1::Button8Click(TObject *Sender) // Кнопка "ctg(x)"
{ float x,s; x=StrToFloat(Edit1->Text);
s=Cotan(x);
Edit3->Text =FloatToStr(s);
Memol->Lines ->Add( "ctg("+ FloatToStr(x) +")=" + FloatToStr(s));
}
//
void __fastcall TForm1::Button9Click(TObject *Sender) // Кнопка "exp(x)"
{ float x, s;
x = StrToFloat(Edit1->Text);
s = exp(x);
Edit3->Text = FloatToStr(s);
Memol->Lines->Add( "exp("+ FloatToStr(x) +")=" + FloatToStr(s));
}
//
void __fastcall TForm1::Button10Click(TObject *Sender) // Кнопка "ln(x)"
{ float x, s;
x = StrToFloat(Edit1->Text);
s = log(x);
Edit3->Text = FloatToStr(s);
Memol->Lines->Add( "ln("+ FloatToStr(x) +")=" + FloatToStr(s));
}
//
void __fastcall TForm1::Button11Click(TObject *Sender) // Кнопка "lg(x)"
{ float x, s;
x = StrToFloat(Edit1->Text);
s = log10(x);
Edit3->Text = FloatToStr(s);
Memol->Lines->Add( "lg("+ FloatToStr(x) +")=" + FloatToStr(s));
}
//
void __fastcall TForm1::Button12Click(TObject *Sender) // Кнопка "log y(x)"
{ float x, y, s;
x = StrToFloat(Edit1->Text);

```

```

y = StrToFloat(Edit2->Text);
s = LogN(x, y);
Edit3->Text = FloatToStr(s);
Mem1->Lines->Add("log"+FloatToStr(y)+" (" +FloatToStr(x)+") =" +
FloatToStr(s));
}
//
void__fastcall TForm1::Button13Click(TObject *Sender) // Кнопка "arcsin(x)"
{ float x, s;
x = StrToFloat(Edit1->Text);
s = asin(x);
Edit3->Text = FloatToStr(s);
Mem1->Lines->Add( "arcsin(" + FloatToStr(x) + ") =" + FloatToStr(s));
}
//
void__fastcall TForm1::Button14Click(TObject *Sender) // Кнопка "x^y"
{float x, y, s;
x = StrToFloat(Edit1->Text);
y = StrToFloat(Edit2->Text);
s = pow(x, y);
Edit3->Text = FloatToStr(s);
Mem1->Lines->Add(FloatToStr(x) + " в степені " + FloatToStr(y) + "=" +
FloatToStr(s));
}
//
void__fastcall TForm1::Button15Click(TObject *Sender) // Кнопка "sqrt(x)"
{ float x, s;
x = StrToFloat(Edit1->Text);
s = sqrt(x);
Edit3->Text = FloatToStr(s);
Mem1->Lines->Add( "sqrt(" + FloatToStr(x) + ") =" + FloatToStr(s));
}
//
void__fastcall TForm1::Button16Click(TObject *Sender) // Кнопка "Pi"
{ Edit3->Text = FloatToStr(M_PI);
}
//
void__fastcall TForm1::Button17Click(TObject *Sender) // Кнопка "Вихід"
{ ShowMessage("Програма припиняє свою роботу");
Close();
}

```

Контрольні запитання

- 1 Для чого потрібна форма проекту?
- 2 Для чого потрібна палітра компонентів?
- 3 Для чого потрібен інспектор об'єктів?
- 4 Перелічіть компоненти, які ви вже знаєте.
- 5 Які властивості компонентів ви змінювали і в який спосіб це зробили?
- 6 Де записують тексти програм?
- 7 Для чого потрібні бібліотеки математичних функцій і в який спосіб вони добуваються?
- 8 В яких бібліотеках містяться математичні функції?
- 9 В який спосіб можна зберегти проект?
- 10 В який спосіб можна запустити проект на виконання?

Додаток А

Основні математичні функції

Функція	Опис	Бібліотека
int abs (int <i>i</i>)	модуль цілого числа $x - x $	stdlib.h
double fabs (double <i>x</i>)	модуль дійсного числа $x - x $	math.h
double sqrt (double <i>x</i>)	корінь квадратний – \sqrt{x}	math.h
double pow (double <i>x</i> , double <i>y</i>)	x^y – піднесення x до степеня $y - x^y$	math.h
Extended IntPower (Extended <i>Base</i> , int <i>x</i>)	піднесення <i>Base</i> до цілого степеня x	Math.hpp
double exp (double <i>x</i>)	експонента e^x	math.h
double log (double <i>x</i>)	натуральний логарифм – $\ln(x)$	math.h
double log10 (double <i>x</i>)	десятковий логарифм – $\lg(x)$	math.h
Extended LogN (Extended <i>N</i> , Extended <i>X</i>)	логарифм X на основі $N - \log_N(X)$	Math.hpp
double cos (double <i>x</i>)	косинус – $\cos(x)$	math.h
double sin (double <i>x</i>)	синус – $\sin(x)$	math.h
double tan (double <i>x</i>)	тангенс – $\text{tg}(x)$	math.h
double Cotan (double <i>x</i>)	котангенс – $\text{ctg}(x)$	Math.hpp
double acos (double <i>x</i>)	арккосинус – $\arccos(x)$	math.h
double asin (double <i>x</i>)	арксинус – $\arcsin(x)$	math.h
double atan (double <i>x</i>)	арктангенс – $\arctg(x)$	math.h
double ceil (double <i>x</i>)	округлення доверху: найменше ціле, не менше за x	math.h
double floor (double <i>x</i>)	округлення донизу: найбільше ціле, не більше за x	math.h
M_PI	константа $\pi = 3.14159$	math.h
M_PI_2	константа $\pi/2$	math.h