Практична робота
№ 12

Тема: **«Основні функції базової графіки в C++ Builder».**

Виконав: **Устич Максим**

Група : **alk43**

Київ 2025р.

Мета роботи:

Детальне знайомство з можливостями графічних функцій в C++ на конкретному прикладі програми, яка буде керувати кольором і шаблонами заливки, відображати текстову інформацію, малювати за допомогою графічних примітивів елементи в середовищі програмування C++ Builder.

Хід роботи:

Елемент	Тип	Призначення
PageControl1	TPageControl	Вкладки програми
TabSheet1–TabSheet6	TTabSheet	Розділи (вкладки) програми
Image1–Image7	TImage	Полотна для малювання
Panel1–Panel7	TPanel	Панелі з кнопками
Button1–Button14	TButton	Кнопки керування
RadioGroup1–RadioGroup4	TRadioGroup	Вибір режимів малювання
Timer1	TTimer	Таймер для мультиплікації

Опис вкладок:

1. **Малювання пікселями і пером**
   Малювання кривої за допомогою пікселів і пера, очищення та вихід.

2. **Фігури**
   Демонстрація графічних примітивів: Arc, Chord, Ellipse, Pie, Polygon, Rectangle.

3. **Стилі ліній**
   Відображення різних стилів пера (суцільна, штрихова, крапкова тощо).

4. **Синусоїди**

   Побудова двох синусоїд різної частоти.

5. **Мультиплікація**

   Малювання “чоловічка”, який рухається і крокує за допомогою таймера.

6. **Перо і пензель**

   Демонстрація використання пензля (Brush) з різними стилями і кольорами.

Код програми Unit1.cpp:

Unit1.cpp

Search for a type | Search for a method

```
//---------------------------------------------------------------------------

#include <vcl.h>
#pragma hdrstop
#include <math.h>
#include "Unit1.h"

//---------------------------------------------------------------------------
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm1 *Form1;

//---------------------------------------------------------------------------
int posX = 0; // для мультиплікації

//---------------------------------------------------------------------------
__fastcall TForm1::TForm1(TComponent* Owner)
    : TForm(Owner)
{
    // Білий фон для всіх Image при запуску
    TImage* images[] = { Image1, Image2, Image3, Image4, Image5, Image6, Image7 };
    for (TImage* img : images)
    {
        img->Canvas->Brush->Color = clWhite;
        img->Canvas->FillRect(Rect(0, 0, img->Width, img->Height));
    }
}

//---------------------------------------------------------------------------
// Малювання пікселями
void __fastcall TForm1::Button1Click(TObject *Sender)
{
    int a = 100, px, py, cx, cy;
    const float pi = 3.141593;
    cx = Image1->Width / 2;
    cy = Image1->Height / 2;

    for (int i = 0; i <= 180; i++)
    {
        px = (int)(a * pow(cos(2 * pi * i / 180), 3));
        py = (int)(a * pow(sin(2 * pi * i / 180), 3));
        Image1->Canvas->Pixels[cx + px][cy + py] = clBlue;
    }
}
```

```
  // Малювання пером
void __fastcall TForm1::Button2Click(TObject *Sender)
{
    int a = 100, px, py, cx, cy;
    const float pi = 3.141593;
    cx = Image1->Width / 2;
    cy = Image1->Height / 2;

    Image1->Canvas->MoveTo(cx, cy);
    Image1->Canvas->Pen->Color = clRed;

    for (int i = 0; i <= 180; i++)
    {
        px = (int)(a * pow(cos(2 * pi * i / 180), 3));
        py = (int)(a * pow(sin(2 * pi * i / 180), 3));
        Image1->Canvas->LineTo(cx + px, cy + py);
    }
}

//---------------------------------------------------------------------------
// Очищення
void __fastcall TForm1::Button3Click(TObject *Sender)
{
    Image1->Canvas->Brush->Color = clWhite;
    Image1->Canvas->FillRect(Rect(0, 0, Image1->Width, Image1->Height));
}

//---------------------------------------------------------------------------
void __fastcall TForm1::Image1MouseDown(TObject *Sender,
    TMouseButton Button, TShiftState Shift, int X, int Y)
{
    if (RadioGroup1->ItemIndex == 0)
    {
        for (int i = 0; i < 300; i += 3)
            Image1->Canvas->Pixels[i][i] = clBlue;
    }
    else
    {
        Image1->Canvas->Pen->Color = clRed;
        Image1->Canvas->LineTo(X, Y);
    }
}


//---------------------------------------------------------------------------
void __fastcall TForm1::RadioGroup1Click(TObject *Sender)
{
    // не використовується
}

//---------------------------------------------------------------------------
void __fastcall TForm1::Button4Click(TObject *Sender)
{

}

//---------------------------------------------------------------------------
// Вкладка "Фігури"
void __fastcall TForm1::Button5Click(TObject *Sender)
{
    Image2->Canvas->Font->Style = TFontStyles() << fsBold;

    Image2->Canvas->Arc(10,10,90,90,10,50,50,10);
    Image2->Canvas->TextOut(40,60,"Arc");

    Image2->Canvas->Chord(110,10,190,90,50,110,50,90);
    Image2->Canvas->TextOut(135,60,"Chord");

    Image2->Canvas->Ellipse(210,10,290,90);
    Image2->Canvas->TextOut(230,60,"Ellipse");

    Image2->Canvas->Pie(310,10,390,90,390,30,310,30);
    Image2->Canvas->TextOut(340,60,"Pie");

    TPoint points1[3] = { Point(10,120), Point(50,180), Point(90,120) };
    Image2->Canvas->Polygon(points1, 2);
    Image2->Canvas->TextOut(35,150,"Polygon");
```

```
    TPoint points2[4] = { Point(110,120), Point(130,180), Point(170,180), Point(190,120) };
    Image2->Canvas->Polyline(points2, 3);
    Image2->Canvas->TextOut(135,150,"Polyline");

    Image2->Canvas->Rectangle(210,120,290,180);
    Image2->Canvas->TextOut(225,150,"Rectangle");

    Image2->Canvas->RoundRect(310,120,390,180,20,20);
    Image2->Canvas->TextOut(325,150,"RoundRect");
}

//---------------------------------------------------------------------------
void __fastcall TForm1::Button6Click(TObject *Sender)
{
    Close();
}

//---------------------------------------------------------------------------
// Вкладка "Стилі ліній"
void __fastcall TForm1::Button7Click(TObject *Sender)
{
    Image3->Canvas->Brush->Color = clWhite;
    Image3->Canvas->FillRect(Rect(0, 0, Image3->Width, Image3->Height));

    for (int i = 1; i < 8; i++)
    {
        Image3->Canvas->Pen->Style = TPenStyle(i - 1);
        Image3->Canvas->MoveTo(10, i * 25);
        Image3->Canvas->LineTo(Image3->Width - 10, i * 25);

        switch (i)
        {
            case 1: Image3->Canvas->TextOut(20, i * 25 + 2, "psSolid"); break;
            case 2: Image3->Canvas->TextOut(20, i * 25 + 2, "psDash"); break;
            case 3: Image3->Canvas->TextOut(20, i * 25 + 2, "psDot"); break;
            case 4: Image3->Canvas->TextOut(20, i * 25 + 2, "psDashDot"); break;
            case 5: Image3->Canvas->TextOut(20, i * 25 + 2, "psDashDotDot"); break;
            case 6: Image3->Canvas->TextOut(20, i * 25 + 2, "psClear"); break;
            case 7: Image3->Canvas->TextOut(20, i * 25 + 2, "psInsideFrame"); break;
        }
    }
}

//---------------------------------------------------------------------------
void __fastcall TForm1::Button8Click(TObject *Sender)
{
    Close();
}

//---------------------------------------------------------------------------
// Вкладка "Синусоїди"
void __fastcall TForm1::Button9Click(TObject *Sender)
{
    int x0 = Image4->Width / 2;
    int y0 = Image4->Height / 2;
    Image4->Canvas->Pen->Color = clBlue;
    Image4->Canvas->Brush->Color = clWhite;
    Image4->Canvas->FillRect(Rect(0, 0, Image4->Width, Image4->Height));

    for (int x = 0; x < Image4->Width; x++)
    {
        int y = y0 - int(50 * sin(x * 0.05));
        Image4->Canvas->Pixels[x][y] = clBlue;
    }

    x0 = Image5->Width / 2;
    y0 = Image5->Height / 2;
    Image5->Canvas->Pen->Color = clRed;
    Image5->Canvas->Brush->Color = clWhite;
    Image5->Canvas->FillRect(Rect(0, 0, Image5->Width, Image5->Height));

    for (int x = 0; x < Image5->Width; x++)
    {
        int y = y0 - int(50 * sin(x * 0.1));
        Image5->Canvas->Pixels[x][y] = clRed;
    }
}
```

Calculating...

```
//---------------------------------------------------------------------------
void __fastcall TForm1::Button10Click(TObject *Sender)
{
    Close();
}

//---------------------------------------------------------------------------
// Вкладка "Мультиплікація"
void __fastcall TForm1::Button11Click(TObject *Sender)
{
    if (!Timer1->Enabled)
    {
        Timer1->Enabled = true;
        Button11->Caption = "Стоп";
    }
    else
    {
        Timer1->Enabled = false;
        Button11->Caption = "Пуск";
    }
}

//---------------------------------------------------------------------------
void __fastcall TForm1::Timer1Timer(TObject *Sender)
{
    static int step = 0;
    Image6->Canvas->Brush->Color = clWhite;
    Image6->Canvas->FillRect(Rect(0, 0, Image6->Width, Image6->Height));

    int y = 150;
    int x = 50 + posX;
    int tilt = (step % 20 < 10) ? -3 : 3;

    Image6->Canvas->Pen->Width = 2;
    Image6->Canvas->Pen->Color = clBlack;

    Image6->Canvas->Ellipse(x, y - 40, x + 20, y - 20);
    Image6->Canvas->MoveTo(x + 10 + tilt, y - 20);
    Image6->Canvas->LineTo(x + 10 - tilt, y + 20);

    int armOffset = (step % 20 < 10) ? 10 : -10;
    Image6->Canvas->MoveTo(x + 10, y - 10);
    Image6->Canvas->LineTo(x - 20 + armOffset, y + 10);
    Image6->Canvas->MoveTo(x + 10, y - 10);
    Image6->Canvas->LineTo(x + 40 - armOffset, y + 10);

    int legOffset = (step % 20 < 10) ? 8 : -8;
    Image6->Canvas->MoveTo(x + 10, y + 20);
    Image6->Canvas->LineTo(x - 5, y + 50 + legOffset);
    Image6->Canvas->MoveTo(x + 10, y + 20);
    Image6->Canvas->LineTo(x + 25, y + 50 - legOffset);

    posX += 5;
    if (x > Image6->Width - 50)
        posX = 0;

    step++;
}

//---------------------------------------------------------------------------
void __fastcall TForm1::Button12Click(TObject *Sender)
{
    Close();
}

//---------------------------------------------------------------------------
// Вкладка "Перо і пензель"
void __fastcall TForm1::RadioGroup2Click(TObject *Sender)
{
    Image7->Canvas->Brush->Style = bsSolid;
    Image7->Canvas->Brush->Color = clWhite;
    Image7->Canvas->FillRect(Rect(0, 0, Image7->Width, Image7->Height));

    Image7->Canvas->Pen->Color = clBlack;
    Image7->Canvas->Pen->Width = 2;
```

Calculating...

```
    switch (RadioGroup2->ItemIndex)
    {
        case 0: Image7->Canvas->Rectangle(50, 100, 150, 180); break;
        case 1: Image7->Canvas->Ellipse(200, 100, 300, 180); break;
        case 2: Image7->Canvas->TextOut(80, 60, "Графіка в C++ Builder"); break;
        case 3:
            Image7->Canvas->Brush->Color = clYellow;
            Image7->Canvas->Ellipse(200, 100, 300, 180);
            break;
    }
}

//---------------------------------------------------------------------------
void __fastcall TForm1::RadioGroup3Click(TObject *Sender)
{
    TBrushStyle style;
    switch (RadioGroup3->ItemIndex)
    {
        case 0: style = bsSolid; break;
        case 1: style = bsCross; break;
        case 2: style = bsDiagCross; break;
        case 3: style = bsVertical; break;
        default: style = bsSolid;
    }

    I        as->Brush->Style = style;
    Image7->Canvas->Brush->Color = clYellow;
    Image7->Canvas->Rectangle(50, 100, 150, 180);
    Image7->Canvas->Rectangle(200, 100, 300, 180);
}

//---------------------------------------------------------------------------
void __fastcall TForm1::RadioGroup4Click(TObject *Sender)
{
    switch (RadioGroup4->ItemIndex)
    {
        case 0: Image7->Canvas->Pen->Mode = pmCopy; break;
        case 1: Image7->Canvas->Pen->Mode = pmXor; break;
        case 2: Image7->Canvas->Pen->Mode = pmNotXor; break;
    }
}

//---------------------------------------------------------------------------
void __fastcall TForm1::Button13Click(TObject *Sender)
{
    Image7->Canvas->Brush->Color = clWhite;
    Image7->Canvas->FillRect(Rect(0,0,Image7->Width,Image7->Height));
}

//---------------------------------------------------------------------------
void __fastcall TForm1::Button14Click(TObject *Sender)
{
    Close();
}
```

Рис.1 - Повний код програми **Unit1.cpp.**

Результати виконання:

Після запуску програми користувач може перейти між шістьма вкладкамита переглянути результати малювання, побудови фігур і анімації.

Вкладка «Малювання пікселями і пером».

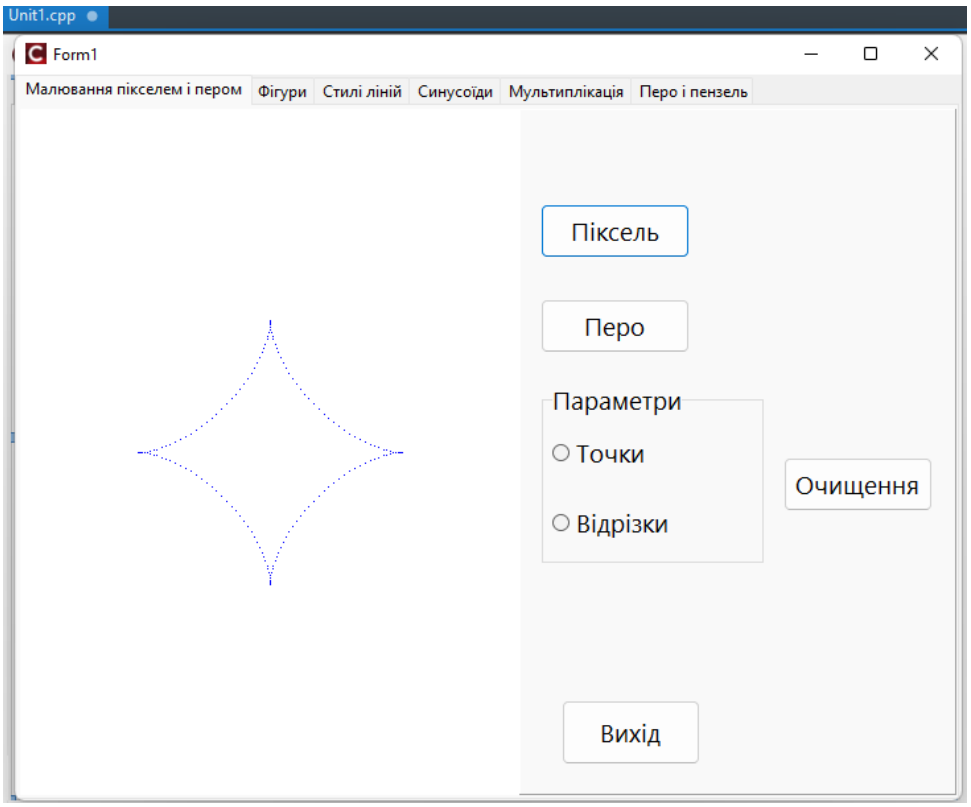

Рис. 2 – Вкладка «Малювання пікселями і пером».

Вкладка «Фігури».

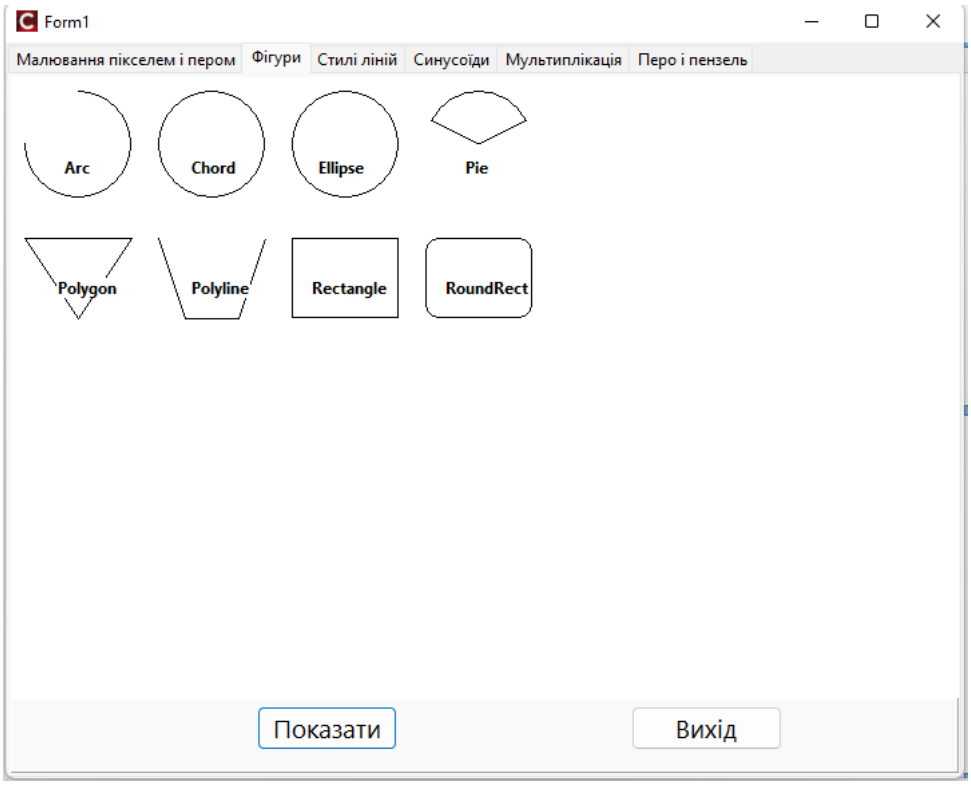

Рис. 3 – Вкладка «Фігури».

Вкладка «Стилі ліній».

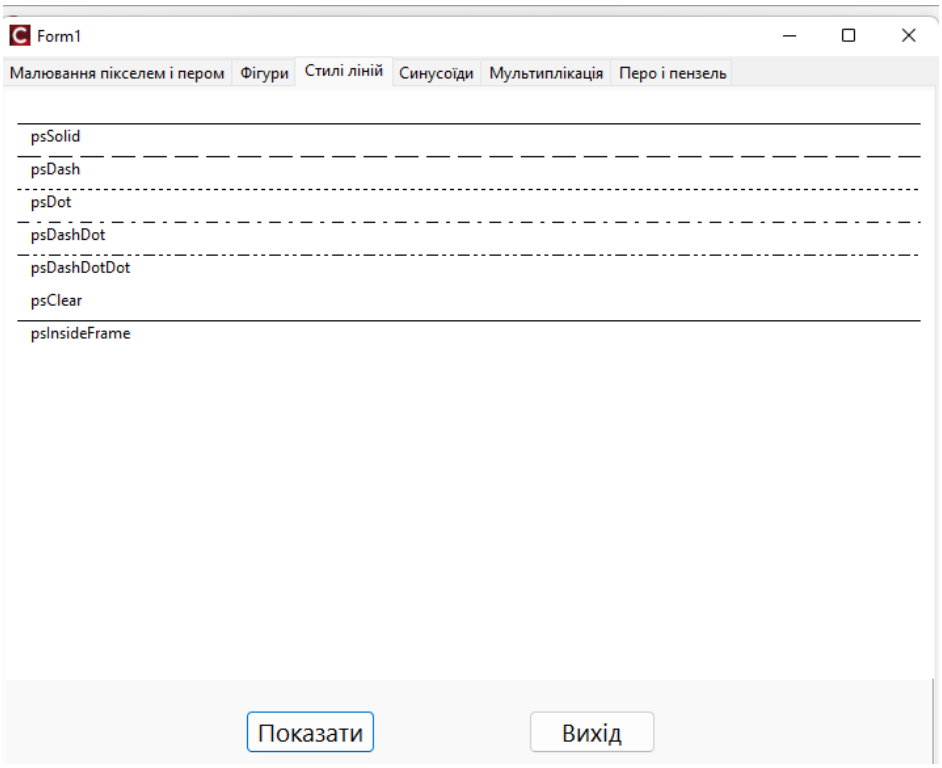

Рис. 4 – Вкладка «Стилі ліній».

Вкладка «Синусоїди».

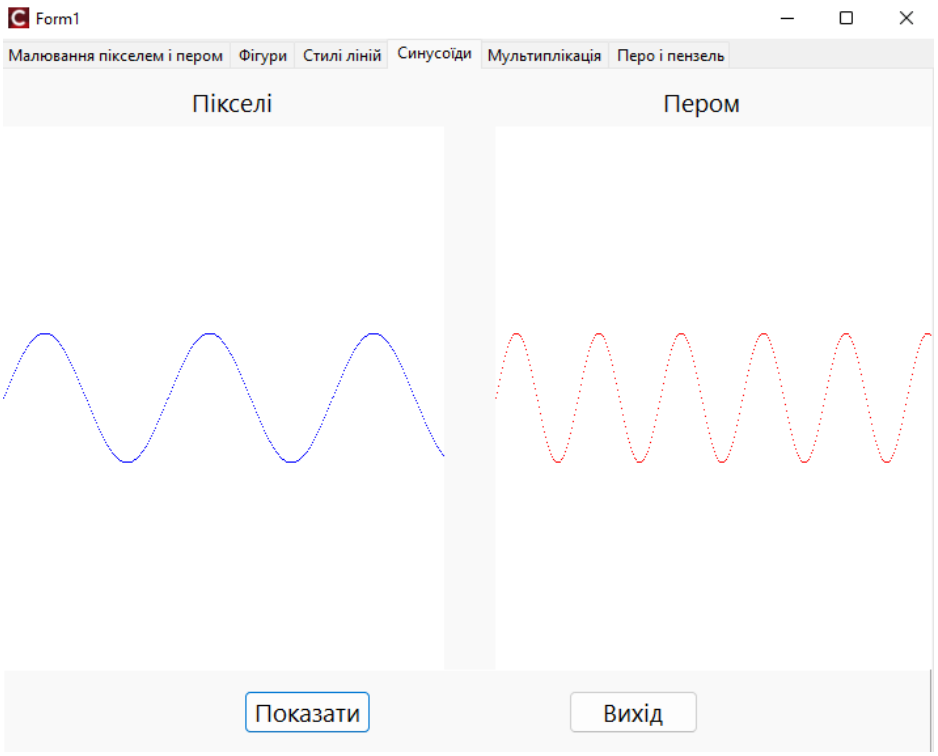

Рис. 5 – Вкладка «Синусоїди».

Вкладка «Мультиплікація».

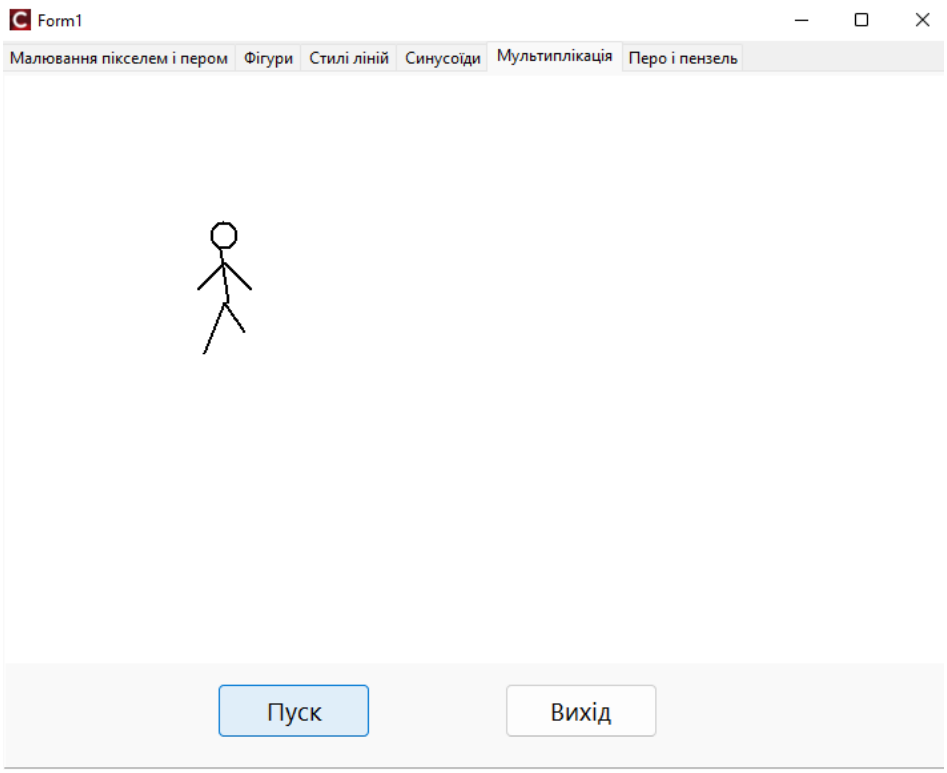

Рис. 6 – Вкладка «Мультиплікація».

Вкладка «Перо і пензель».

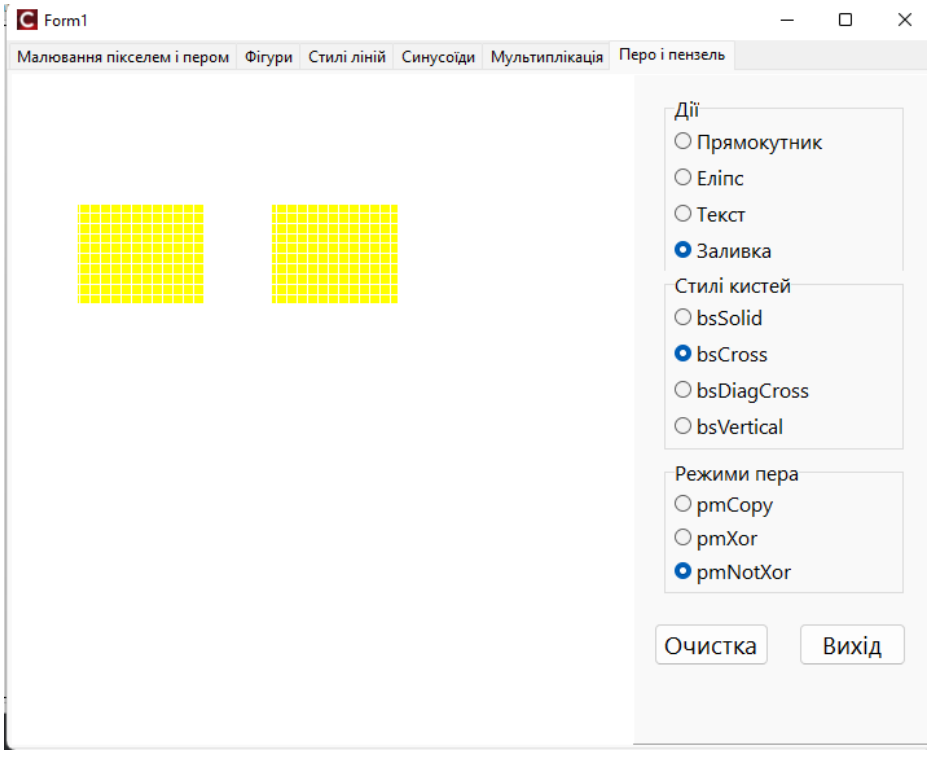

Рис. 7 – Вкладка «Перо і пензель».

Додаткове завдання

Побудовано графік функції та креслення за індивідуальним завданням. У програмі додано **сьому вкладку "Додаткове завдання"**, яка містить два приклади:

1. **Графік функції:** побудова параболи
   Формула:

$$y = x^2$$

Побудова виконується з використанням циклу та методів `MoveTo` і `LineTo`.

2. **Креслення:** стилізоване зображення будинку із вікнами, дахом і дверима,
   створене за допомогою графічних примітивів `Rectangle`, `Polygon`, `Ellipse`.

Код програми Додаткового завдання:

Unit1.cpp

Search for a type | Search for a method

```
}

//---------------------------------------------------------------------------
// Вкладка "Додаткове завдання"
void __fastcall TForm1::Button15Click(TObject *Sender)
{
    Image8->Canvas->Brush->Color = clWhite;
    Image8->Canvas->FillRect(Rect(0, 0, Image8->Width, Image8->Height));

    Image8->Canvas->Pen->Color = clBlue;

    for (int x = 0; x < Image8->Width; x++)
    {
        double y = 0.02 * pow(x - 100, 2) - 2 * (x - 100) + 100;
        int screenY = Image8->Height - int(y);
        if (screenY >= 0 && screenY < Image8->Height)
            Image8->Canvas->Pixels[x][screenY] = clBlue;
    }

    Image8->Canvas->TextOut(10, 10, "y = 0.02x^2 - 2x + 100");
}
```

```
//---------------------------------------------------------------------------
void __fastcall TForm1::Button16Click(TObject *Sender)
{
    Image8->Canvas->Brush->Color = clWhite;
    Image8->Canvas->FillRect(Rect(0, 0, Image8->Width, Image8->Height));

    // Дім
    Image8->Canvas->Brush->Color = clYellow;
    Image8->Canvas->Rectangle(100, 150, 250, 250);

    // Дах
    Image8->Canvas->Brush->Color = clRed;
    TPoint roof[3] = { Point(100,150), Point(175,80), Point(250,150) };
    Image8->Canvas->Polygon(roof, 2);

    // Вікно
    Image8->Canvas->Brush->Color = clAqua;
    Image8->Canvas->Rectangle(130,180,170,220);

    // Двері
Image8->Canvas->Brush->Color = (TColor)0x996633; // темно-бежевий
    Image8->Canvas->Rectangle(190,200,230,250);

    Image8->Canvas->TextOut(120, 260, "Мій дім");
}
```

Вкладка “Додаткове завдання” (графік функції):

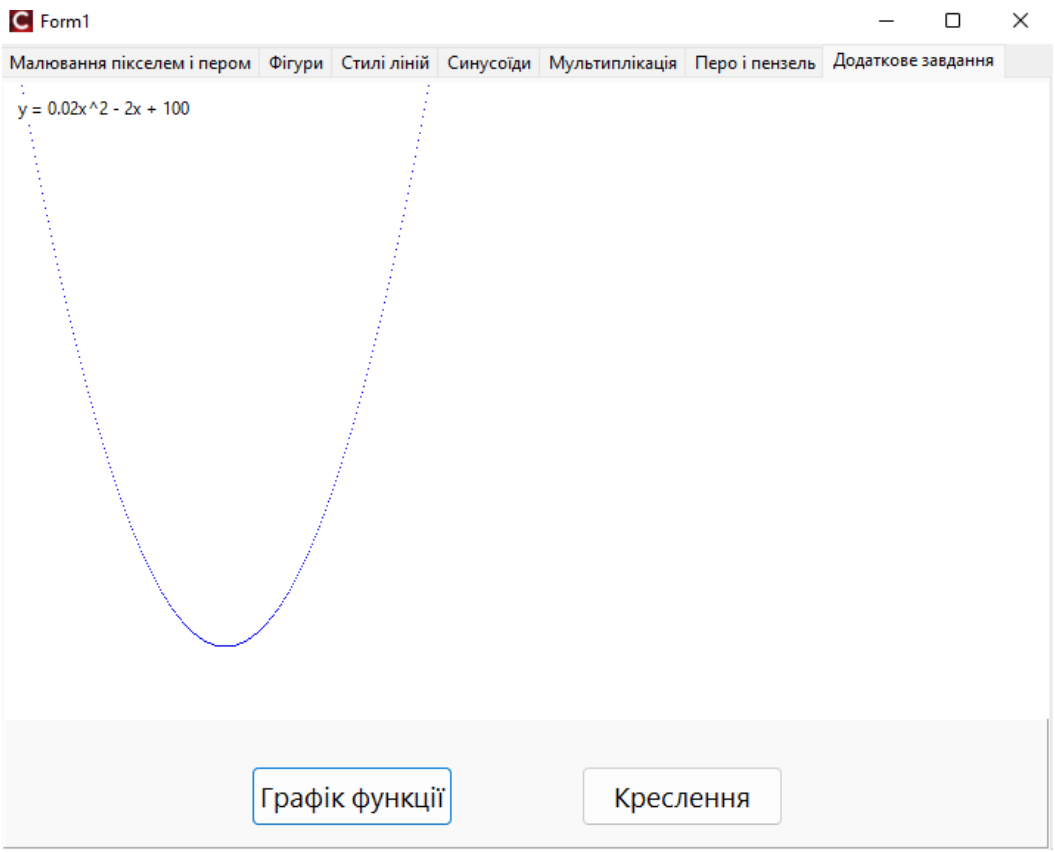

Рис. 8 – Вкладка “Додаткове завдання” (графік функції).

Вкладка “Додаткове завдання” (креслення будинку):

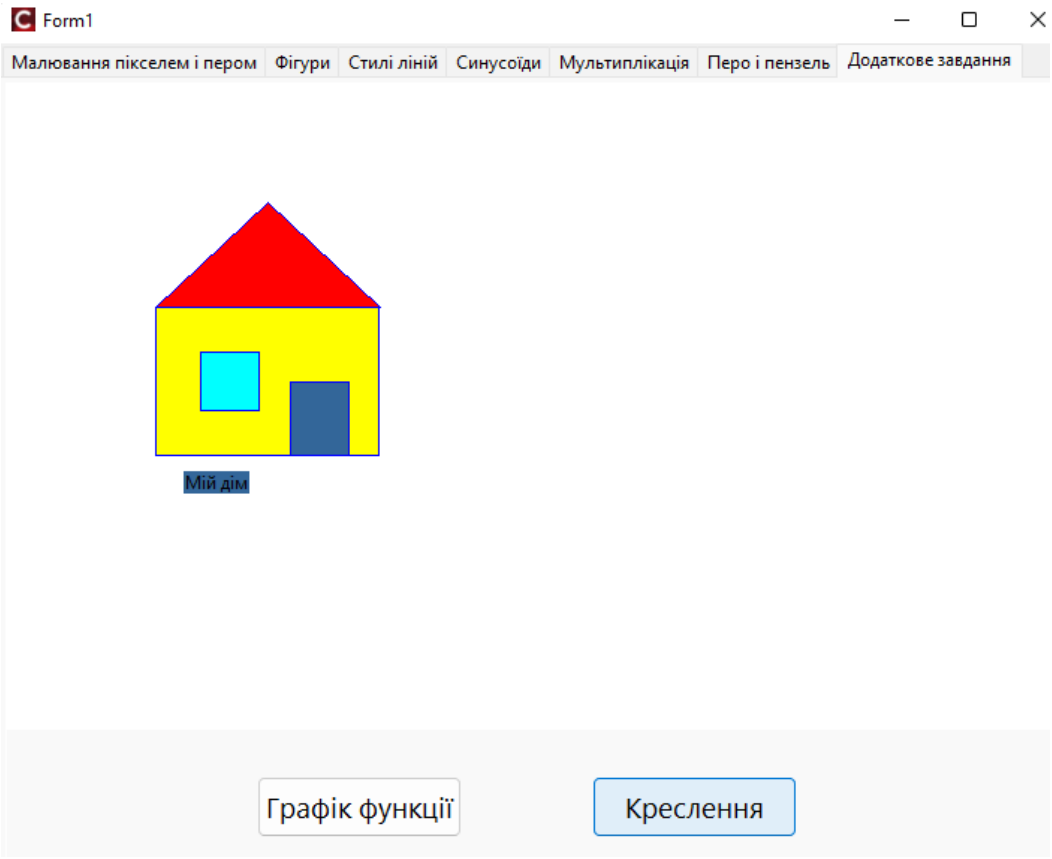

Рис. 9 – Вкладка “Додаткове завдання” (креслення будинку).

Відповіді на контрольні питання

1. В якому файлі бібліотеки зберігаються графічні функції і прототипи (оголошення) цих функцій?

Усі графічні функції та їх прототипи зберігаються в **заголовному файлі graphics.hpp**, який підключається автоматично через `#include <vcl.h>` у середовищі **C++ Builder**.

Саме цей файл містить оголошення класу `TCanvas` і методів для малювання.

2. Який метод викреслює прямокутник з округленими кутами?

Метод `RoundRect`.

Використовується так:

```
Canvas->RoundRect(Left, Top, Right, Bottom, XRadius, YRadius);
```

де `XRadius`, `YRadius` – радіуси заокруглення.

3. Для чого застосовується функція Polyline, і які вона має параметри?

Функція `Polyline` малює ламану лінію, що з'єднує послідовність точок.

Форма виклику:

Canvas->Polyline(Points, Count);

де

- `Points` – масив точок (`TPoint`),
- `Count` – кількість точок мінус 1.

4. Для чого застосовується метод Canvas, і які графічні функції він містить?

Метод `Canvas` — це полотно (контекст малювання) компонента. Він забезпечує доступ до графічних властивостей і методів, таких як:

- `MoveTo`, `LineTo`, `Ellipse`, `Rectangle`, `Polygon`,
- `Brush`, `Pen`, `Font` (для налаштування кольору, стилю, товщини),
- `TextOut`, `FloodFill`, `FillRect`, `FrameRect` тощо.

5. Для чого застосовується функція FloodFill і які вона має параметри?

Функція `FloodFill` виконує заливку області кольором.

Форма виклику:

Canvas->FloodFill(X, Y, BorderColor, FillStyle);

де

- `X, Y` — координати точки початку заливки,
- `BorderColor` — колір межі,
- `FillStyle` — спосіб заливки (`fsSurface` або `fsBorder`).

6. Яку фігуру викреслює метод Polygon? Запишіть інструкцію виклику цього методу.

Метод `Polygon` малює замкнутий багатокутник.

Приклад виклику:

```
TPoint points[3] = { Point(10, 100), Point(60, 50),
Point(110, 100) };

Canvas->Polygon(points, 2);
```

7. Яку функцію виконують методи TextWidth і TextHeight?

- `TextWidth()` — повертає ширину тексту у пікселях.
- `TextHeight()` — повертає висоту тексту у пікселях.

  Використовуються для точного вирівнювання написів на зображенні.

8. Які функції виконують методи FillRect і FrameRect?

- `FillRect(Rect)` — заливає прямокутну область кольором, визначеним у `Brush`.

- `FrameRect(Rect)` — малює рамку навколо прямокутника кольором пера (`Pen`).