Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное

учреждение высшего образования

ИРКУТСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ

ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

Институт Информационных технологий и анализа данных

**«РАБОТА С ГРАФИКОЙ В С++ BUILDER»**

Отчет по лабораторной работе № 9

Вариант 22

по дисциплине Программирование на языке высокого уровня

Выполнилcтудент ИСТБ-23-1 А.О. Чебоксарова

номер группы подпись И. О. Фамилия

дата

Принял

должность подпись И. О. Фамилия

дата

Иркутск 2024

**Вариант 22**

**Задание 1.1**

Построить графики четырёх функций на PaintBox или Image. Все графики построить различными цветами. Предусмотреть масштабирование. Шаг изменения и цвет графика должны задаваться пользователем.

**Задание 1.2**

Построить графики тех же функций с использованием Chart с несколькими сериями и выбором цвета линий.

**Задание 2**

Построить график кардиоиды PaintBox или Image. Все графики построить различными цветами. Предусмотреть масштабирование. Шаг изменения, параметры и цвет графика должны задаваться пользователем.

**Задание 3**

Изобразить на экране квадрат, вращающийся в плоскости экрана вокруг своего центра.

**Задание 4**

Используя компоненты Image и Timer организовать смену картинок, сформировать анимационный ролик.

Постановка задачи

1. Построить графики четырёх функций на PaintBox или Image.
2. Построить графики тех же функций с использованием Chart.
3. Построить график кардиоиды PaintBox или Image. Oчистка Canvas в PaintBox.
4. Изобразить на экране квадрат
5. Отрисовка графика кардиоиды.
6. Используя компоненты Image и Timer организовать смену картинок.

Вх. данные:

Form2: Выбор всех характеристик графиков

Form4: Выбор всех характеристик кардиоиды

Вых.данные:

* Form2: Вывод графиков и анимация квадрата происходит на компоненте PaintBox или Chart
* Form4: Вывод графика кардиоиды на компоненте PaintBox.
* Form5: смена картинок на компоненте Image.

Схема иерархии функций

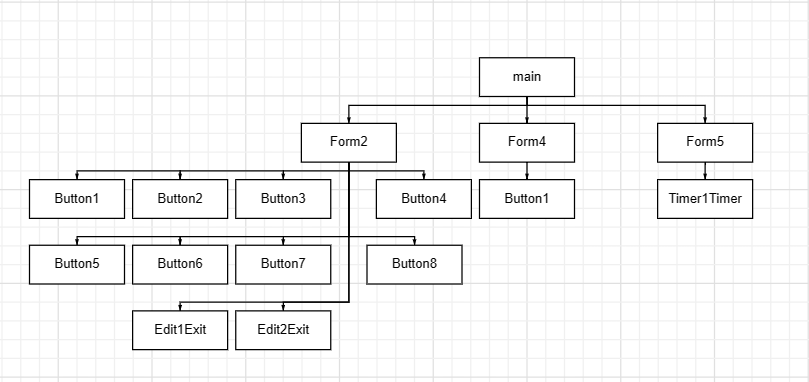


Рис 1.- Cхема иерархии функций

Таблица 1- Формирование спецификаций функций

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Имя | Назначение | Тип | | ОДЗ |
| **Button1Click, Button2Click, Button3Click, Button4Click Form2** | | | | | |
| 1 | centerX | Координата центра по оси X | int | От −2 147 483 648 до 2 147 483 647 | |
| 2 | centerY | Координата центра по оси Y | int | От −2 147 483 648 до 2 147 483 647 | |
| 3 | Len1 | Длина единичного отрезка | int | От −2 147 483 648 до 2 147 483 647 | |
| 4 | dopY | Вспомогательная переменная для координаты Y | int | От −2 147 483 648 до 2 147 483 647 | |
| 5 | dopX | Вспомогательная переменная для координаты X | int | От −2 147 483 648 до 2 147 483 647 | |
| 6 | stepY | Шаг по оси Y | int | От −2 147 483 648 до 2 147 483 647 | |
| 7 | labelY | Координаты по оси Y | int | От −2 147 483 648 до 2 147 483 647 | |
| 8 | stepX | Шаг по оси X | int | От −2 147 483 648 до 2 147 483 647 | |
| 9 | labelX | Координаты по X | int | От −2 147 483 648 до 2 147 483 647 | |
| 10 | xmin | Минимальное значение по оси X | double | 1,7E +/- 308 (15 знаков) | |
| 11 | xmax | Максимальное значение по оси Y | double | 1,7E +/- 308 (15 знаков) | |
| 12 | prevX | Предыдущее значение по оси X | double | 1,7E +/- 308 (15 знаков) | |
| 13 | prexY | Предыдущее значение по оси Y | double | 1,7E +/- 308 (15 знаков) | |
| 14 | x | Текущее значение по оси X | double | 1,7E +/- 308 (15 знаков) | |
| 15 | y | Текущее значение по оси Y | double | 1,7E +/- 308 (15 знаков) | |
| **Button7Click Form2** | | | | | |
| 16 | xmin | Минимальное значение по оси X | double | 1,7E +/- 308 (15 знаков) | |
| 17 | xmax | Максимальное значение по оси Y | double | 1,7E +/- 308 (15 знаков) | |
| 18 | step | Шаг для построения графика | double | 1,7E +/- 308 (15 знаков) | |
| 19 | x | Текущее значение аргумента для построения графика | double | 1,7E +/- 308 (15 знаков) | |
| **Button6Click Form2** | | | | | |
| 20 | paintBoxWidth | Ширина компонента PaintBox1 | int | От −2 147 483 648 до 2 147 483 647 | |
| 21 | paintBoxHeight | Высота компонента PaintBox1 | int | От −2 147 483 648 до 2 147 483 647 | |
| 22 | x | Значение координаты X для квадрата | int | От −2 147 483 648 до 2 147 483 647 | |
| 23 | y | Значение координаты Y для квадрата | int | От −2 147 483 648 до 2 147 483 647 | |
| 24 | centerX | Координата X центра квадрата | int | От −2 147 483 648 до 2 147 483 647 | |
| 25 | centerY | Координата Y центра квадрата | int | От −2 147 483 648 до 2 147 483 647 | |
| **Timer1Timer Form2** | | | | | |
| 26 | rotationAngle | Угол поворота квадрата | int | От −2 147 483 648 до 2 147 483 647 | |
| 27 | sideLength | Длина стороны квадрата | int | От −2 147 483 648 до 2 147 483 647 | |
| 28 | points | Массив точек для рисования квадрата | TPoint | - | |
| 29 | angleRad | Угол поворота в радианах | double | 1,7E +/- 308 (15 знаков) | |
| **Button1Click Form4** | | | | | |
| 30 | a | Параметр «а» кардиоиды | int | От −2 147 483 648 до 2 147 483 647 | |
| 31 | tStep | Шаг «t» для построения | double | 1,7E +/- 308 (15 знаков) | |
| 32 | x | Координата X | double | 1,7E +/- 308 (15 знаков) | |
| 33 | y | Координата Y | double | 1,7E +/- 308 (15 знаков) | |
| 34 | t | Параметр «t» для кардиоиды | double | 1,7E +/- 308 (15 знаков) | |
| 35 | prevX | Предыдущая координата X | int | От −2 147 483 648 до 2 147 483 647 | |
| 36 | prevY | Предыдущая координата Y | int | От −2 147 483 648 до 2 147 483 647 | |
| 37 | scaledX | Масштабированная координата X | int | От −2 147 483 648 до 2 147 483 647 | |
| 38 | scaledY | Масштабированная координата Y | int | От −2 147 483 648 до 2 147 483 647 | |
| 39 | firstPoint | Флаг первой точки | bool |  | |
| **Timer1Timer Form 5** | | | | | |
| 40 | ImageIndex | Индекс изображения | int | От −2 147 483 648 до 2 147 483 647 | |

Описание функций

Таблица 2- Описание функций

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Название функции | Назначение функции | Тип | Параметры |
| 1 | Button1Click Form2 | Отрисовка графика x3 | void | TObject \*Sender |
| 2 | Button2Click  Form2 | Отрисовка графика 2x3 | void | TObject \*Sender |
| 3 | Button3Click  Form2 | Отрисовка графика x3/2 | void | TObject \*Sender |
| 4 | Button4Click  Form2 | Отрисовка графика (x/2)3 | void | TObject \*Sender |
| 5 | Button7Click  Form2 | Отображение графиков в Chart | void | TObject \*Sender |
| 6 | Button6Click  Form2 | Анимация квадрата | void | TObject \*Sender |
| 7 | Button5Click  Form2 | Вызов кардиоиды | void | TObject \*Sender |
| 8 | Button1Click Form4 | Отрисовка графика кардиоиды | void | TObject \*Sender |
| 9 | Button8Click  Form2 | Запуск смены картинок | void | TObject \*Sender |
| 10 | Timer1Timer  Form5 | Смена картинок | void | TObject \*Sender |

Тесты программы

Таблица 3- Тесты программы

|  |  |  |  |
| --- | --- | --- | --- |
| № | Назначение теста | Исходные данные | Ожидаемый результат |
| 1 | Создать графики различных цветов на компоненте PaintBox | xmin: -10  xmax:10 | Графики четырёх функций (x3, 2x3, x3/2, (x/2)3) на компоненте PaintBox. |
| 2 | Создать графики различных цветов на компоненте Chart | xmin: -10  xmax:10 | Графики четырёх функций (x3, 2x3, x3/2, (x/2)3) на компоненте Chart. |
| 3 | Создать график кардиоиды на компоненте PaintBox | а: 10  шаг: 0,001 | График кардиоиды на компоненте Chart. |
| 4 | Создать график кардиоиды на компоненте PaintBox | а: 10  шаг: 1 | График кардиоиды на компоненте Chart. |
| 5 | Анимация вращения квадрата вокруг своей оси | - | Анимация вращения квадрата вокруг своей оси на компоненте PaintBox. |
| 6 | Создание анимационного ролика с помощью смены картинок | - | Создание анимационного ролика на компоненте Image |

Текст программы

Form2

#include <vcl.h>

#pragma hdrstop

#include "Unit2.h"

#include "Unit4.h"

#include "Unit5.h"

//---------------------------------------------------------------------------

#pragma package(smart\_init)

#pragma resource "\*.dfm"

TForm2 \*Form2;

//---------------------------------------------------------------------------

\_\_fastcall TForm2::TForm2(TComponent\* Owner)

: TForm(Owner)

{

}

//---------------------------------------------------------------------------

/\*Функция для отрисовки графика x^3

\*/

void \_\_fastcall TForm2::Button1Click(TObject \*Sender)

{

Timer1->Enabled = false;

Chart1->Visible = false;

PaintBox1->Visible=true;

PaintBox1->Canvas->Brush->Color = clWhite;

PaintBox1->Canvas->FillRect(PaintBox1->ClientRect);

int TextOut = PaintBox1->Width / 2, centerY = PaintBox1->Height / 2;

int Len1 = 25;

PaintBox1->Canvas->Pen->Color = clBlack;

PaintBox1->Canvas->MoveTo(0, centerY);

PaintBox1->Canvas->LineTo(PaintBox1->Width, centerY);

PaintBox1->Canvas->MoveTo(centerX, 0);

PaintBox1->Canvas->LineTo(centerX, PaintBox1->Height);

PaintBox1->Canvas->Pen->Style = psDot;

int dopY = centerY - Len1;

while (dopY > 0) {

PaintBox1->Canvas->MoveTo(0, dopY);

PaintBox1->Canvas->LineTo(PaintBox1->Width, dopY);

dopY = dopY - Len1;

}

dopY = centerY + Len1;

while (dopY < PaintBox1->Height) {

PaintBox1->Canvas->MoveTo(0, dopY);

PaintBox1->Canvas->LineTo(PaintBox1->Width, dopY);

dopY = dopY + Len1;

}

int dopX = centerX - Len1;

while (dopX > 0) {

PaintBox1->Canvas->MoveTo(dopX, 0);

PaintBox1->Canvas->LineTo(dopX, PaintBox1->Height);

dopX = dopX - Len1;

}

dopX = centerX + Len1;

while (dopX < PaintBox1->Width) {

PaintBox1->Canvas->MoveTo(dopX, 0);

PaintBox1->Canvas->LineTo(dopX, PaintBox1->Height);

dopX = dopX + Len1;

}

PaintBox1->Canvas->Pen->Style = psSolid;

PaintBox1->Canvas->Font->Size = 8;

PaintBox1->Canvas->Font->Color = clBlack;

int stepY = Len1;

int labelY = 1;

while (centerY - stepY > 0) {

PaintBox1->Canvas->TextOut(centerX + 5, centerY - stepY - 10, IntToStr(labelY));

PaintBox1->Canvas->TextOut(centerX + 5, centerY + stepY - 10, IntToStr(labelY));

stepY += Len1;

labelY += 1;

}

int stepX = Len1;

int labelX = 1;

while (centerX - stepX > 0) {

PaintBox1->Canvas->TextOut(centerX - stepX - 10, centerY + 5, IntToStr(labelX));

PaintBox1->Canvas->TextOut(centerX + stepX - 10, centerY + 5, IntToStr(labelX));

stepX += Len1;

labelX += 1;

}

Form2->Caption = "График функции";

double xmin = StrToFloat(Edit1->Text);

double xmax = StrToFloat(Edit2->Text);

double prevX = xmin, prevY = pow(prevX, 3); for (double x = xmin; x <= xmax; x += 0.01) {

double y; y = pow(x, 3);

PaintBox1->Canvas->Pen->Color = clBlue; PaintBox1->Canvas->Pen->Color = ColorBox1->Selected;

PaintBox1->Canvas->MoveTo(centerX + prevX \* Len1, centerY - prevY \* Len1); PaintBox1->Canvas->LineTo(centerX + x \* Len1, centerY - y \* Len1);

prevX = x; prevY = y;

} PaintBox1->Visible = true;

}

/\*Функция для отрисовки графика 2x^3

\*/

void \_\_fastcall TForm2::Button2Click(TObject \*Sender)

{

Timer1->Enabled = false;

Chart1->Visible = false;

PaintBox1->Visible=true;

PaintBox1->Canvas->Brush->Color = clWhite;

PaintBox1->Canvas->FillRect(PaintBox1->ClientRect);

int centerX = PaintBox1->Width / 2, centerY = PaintBox1->Height / 2;

int Len1 = 25;

PaintBox1->Canvas->Pen->Color = clBlack;

PaintBox1->Canvas->MoveTo(0, centerY);

PaintBox1->Canvas->LineTo(PaintBox1->Width, centerY);

PaintBox1->Canvas->MoveTo(centerX, 0);

PaintBox1->Canvas->LineTo(centerX, PaintBox1->Height);

PaintBox1->Canvas->Pen->Style = psDot;

int dopY = centerY - Len1;

while (dopY > 0) {

PaintBox1->Canvas->MoveTo(0, dopY);

PaintBox1->Canvas->LineTo(PaintBox1->Width, dopY);

dopY = dopY - Len1;

}

dopY = centerY + Len1;

while (dopY < PaintBox1->Height) {

PaintBox1->Canvas->MoveTo(0, dopY);

PaintBox1->Canvas->LineTo(PaintBox1->Width, dopY);

dopY = dopY + Len1;

}

int dopX = centerX - Len1;

while (dopX > 0) {

PaintBox1->Canvas->MoveTo(dopX, 0);

PaintBox1->Canvas->LineTo(dopX, PaintBox1->Height);

dopX = dopX - Len1;

}

dopX = centerX + Len1;

while (dopX < PaintBox1->Width) {

PaintBox1->Canvas->MoveTo(dopX, 0);

PaintBox1->Canvas->LineTo(dopX, PaintBox1->Height);

dopX = dopX + Len1;

}

PaintBox1->Canvas->Pen->Style = psSolid;

PaintBox1->Canvas->Font->Size = 8;

PaintBox1->Canvas->Font->Color = clBlack;

int stepY = Len1;

int labelY = 1;

while (centerY - stepY > 0) {

PaintBox1->Canvas->TextOut(centerX + 5, centerY - stepY - 10, IntToStr(labelY));

PaintBox1->Canvas->TextOut(centerX + 5, centerY + stepY - 10, IntToStr(labelY));

stepY += Len1;

labelY += 1;

}

int stepX = Len1;

int labelX = 1;

while (centerX - stepX > 0) {

PaintBox1->Canvas->TextOut(centerX - stepX - 10, centerY + 5, IntToStr(labelX));

PaintBox1->Canvas->TextOut(centerX + stepX - 10, centerY + 5, IntToStr(labelX));

stepX += Len1;

labelX += 1;

}

Form2->Caption = "График функции";

double xmin = StrToFloat(Edit1->Text);

double xmax = StrToFloat(Edit2->Text);

double prevX = xmin, prevY = pow(prevX, 3) + 2;

for (double x = xmin; x <= xmax; x += 0.01) {

double y =2\*pow(x,3);

PaintBox1->Canvas->Pen->Color = ColorBox2->Selected;

PaintBox1->Canvas->MoveTo(centerX + prevX \* Len1, centerY - prevY \* Len1);

PaintBox1->Canvas->LineTo(centerX + x \* Len1, centerY - y \* Len1);

prevX = x;

prevY = y;

}

PaintBox1->Visible = true;

}

//---------------------------------------------------------------------------

/\*Функция для отрисовки графика x^(3/2)

\*/

void \_\_fastcall TForm2::Button3Click(TObject \*Sender)

{

Timer1->Enabled = false;

Chart1->Visible = false;

PaintBox1->Visible=true;

PaintBox1->Canvas->Brush->Color = clWhite;

PaintBox1->Canvas->FillRect(PaintBox1->ClientRect);

int centerX = PaintBox1->Width / 2, centerY = PaintBox1->Height / 2;

int Len1 = 25;

PaintBox1->Canvas->Pen->Color = clBlack;

PaintBox1->Canvas->MoveTo(0, centerY);

PaintBox1->Canvas->LineTo(PaintBox1->Width, centerY);

PaintBox1->Canvas->MoveTo(centerX, 0);

PaintBox1->Canvas->LineTo(centerX, PaintBox1->Height);

PaintBox1->Canvas->Pen->Style = psDot;

int dopY = centerY - Len1;

while (dopY > 0) {

PaintBox1->Canvas->MoveTo(0, dopY);

PaintBox1->Canvas->LineTo(PaintBox1->Width, dopY);

dopY = dopY - Len1;

}

dopY = centerY + Len1;

while (dopY < PaintBox1->Height) {

PaintBox1->Canvas->MoveTo(0, dopY);

PaintBox1->Canvas->LineTo(PaintBox1->Width, dopY);

dopY = dopY + Len1;

}

int dopX = centerX - Len1;

while (dopX > 0) {

PaintBox1->Canvas->MoveTo(dopX, 0);

PaintBox1->Canvas->LineTo(dopX, PaintBox1->Height);

dopX = dopX - Len1;

}

dopX = centerX + Len1;

while (dopX < PaintBox1->Width) {

PaintBox1->Canvas->MoveTo(dopX, 0);

PaintBox1->Canvas->LineTo(dopX, PaintBox1->Height);

dopX = dopX + Len1;

}

PaintBox1->Canvas->Pen->Style = psSolid;

PaintBox1->Canvas->Font->Size = 8;

PaintBox1->Canvas->Font->Color = clBlack;

int stepY = Len1;

int labelY = 1;

while (centerY - stepY > 0) {

PaintBox1->Canvas->TextOut(centerX + 5, centerY - stepY - 10, IntToStr(labelY));

PaintBox1->Canvas->TextOut(centerX + 5, centerY + stepY - 10, IntToStr(labelY));

stepY += Len1;

labelY += 1;

}

int stepX = Len1;

int labelX = 1;

while (centerX - stepX > 0) {

PaintBox1->Canvas->TextOut(centerX - stepX - 10, centerY + 5, IntToStr(labelX));

PaintBox1->Canvas->TextOut(centerX + stepX - 10, centerY + 5, IntToStr(labelX));

stepX += Len1;

labelX += 1;

}

Form2->Caption = "График функции";

double xmin = StrToFloat(Edit1->Text);

double xmax = StrToFloat(Edit2->Text);

double prevX = xmin, prevY = pow(prevX, 3) + 2;

for (double x = xmin; x <= xmax; x += 0.01) {

if(x>=0){

double y =pow(x,1.5);

PaintBox1->Canvas->Pen->Color = ColorBox3->Selected;

PaintBox1->Canvas->MoveTo(centerX + prevX \* Len1, centerY - prevY \* Len1);

PaintBox1->Canvas->LineTo(centerX + x \* Len1, centerY - y \* Len1);

prevX = x;

prevY = y;

}

}

PaintBox1->Visible = true;

}

//---------------------------------------------------------------------------

/\*Функция для отрисовки графика (x/2)^3

\*/

void \_\_fastcall TForm2::Button4Click(TObject \*Sender)

{

Timer1->Enabled = false;

Chart1->Visible = false;

PaintBox1->Visible=true;

PaintBox1->Canvas->Brush->Color = clWhite;

PaintBox1->Canvas->FillRect(PaintBox1->ClientRect);

int centerX = PaintBox1->Width / 2, centerY = PaintBox1->Height / 2;

int Len1 =25;

PaintBox1->Canvas->Pen->Color = clBlack;

PaintBox1->Canvas->MoveTo(0, centerY);

PaintBox1->Canvas->LineTo(PaintBox1->Width, centerY);

PaintBox1->Canvas->MoveTo(centerX, 0);

PaintBox1->Canvas->LineTo(centerX, PaintBox1->Height);

PaintBox1->Canvas->Pen->Style = psDot;

int dopY = centerY - Len1;

while (dopY > 0) {

PaintBox1->Canvas->MoveTo(0, dopY);

PaintBox1->Canvas->LineTo(PaintBox1->Width, dopY);

dopY = dopY - Len1;

}

dopY = centerY + Len1;

while (dopY < PaintBox1->Height) {

PaintBox1->Canvas->MoveTo(0, dopY);

PaintBox1->Canvas->LineTo(PaintBox1->Width, dopY);

dopY = dopY + Len1;

}

int dopX = centerX - Len1;

while (dopX > 0) {

PaintBox1->Canvas->MoveTo(dopX, 0);

PaintBox1->Canvas->LineTo(dopX, PaintBox1->Height);

dopX = dopX - Len1;

}

dopX = centerX + Len1;

while (dopX < PaintBox1->Width) {

PaintBox1->Canvas->MoveTo(dopX, 0);

PaintBox1->Canvas->LineTo(dopX, PaintBox1->Height);

dopX = dopX + Len1;

}

PaintBox1->Canvas->Pen->Style = psSolid;

PaintBox1->Canvas->Font->Size = 8;

PaintBox1->Canvas->Font->Color = clBlack;

int stepY = Len1;

int labelY = 1;

while (centerY - stepY > 0) {

PaintBox1->Canvas->TextOut(centerX + 5, centerY - stepY - 10, IntToStr(labelY));

PaintBox1->Canvas->TextOut(centerX + 5, centerY + stepY - 10, IntToStr(labelY));

stepY += Len1;

labelY += 1;

}

int stepX = Len1;

int labelX = 1;

while (centerX - stepX > 0) {

PaintBox1->Canvas->TextOut(centerX - stepX - 10, centerY + 5, IntToStr(labelX));

PaintBox1->Canvas->TextOut(centerX + stepX - 10, centerY + 5, IntToStr(labelX));

stepX += Len1;

labelX += 1;

}

Form2->Caption = "График функции";

double xmin = StrToFloat(Edit1->Text);

double xmax = StrToFloat(Edit2->Text);

double prevX = xmin, prevY = pow(prevX, 3) + 2;

for (double x = xmin; x <= xmax; x += 0.01) {

double y =pow(x/2,3);

PaintBox1->Canvas->Pen->Color = ColorBox3->Selected;

PaintBox1->Canvas->MoveTo(centerX + prevX \* Len1, centerY - prevY \* Len1);

PaintBox1->Canvas->LineTo(centerX + x \* Len1, centerY - y \* Len1);

prevX = x;

prevY = y;

}

PaintBox1->Visible = true;

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm2::N4Click(TObject \*Sender)

{

if (SaveDialog1->Execute())

{

TFileStream \*fileStream = new TFileStream(SaveDialog1->FileName, fmCreate);

try

{

}

\_\_finally

{

delete fileStream;

}

}

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm2::N6Click(TObject \*Sender)

{

Application->Terminate();

}

//---------------------------------------------------------------------------

/\* Отображение графиков в Chart

\*/

void \_\_fastcall TForm2::Button7Click(TObject \*Sender)

{

Timer1->Enabled = false;

PaintBox1->Visible = false;

// Получить значения начального и конечного аргумента для графика из полей ввода

double xmin = StrToFloat(Edit1->Text);

double xmax = StrToFloat(Edit2->Text);

// Создать серии TFastLineSeries для каждой функции и установить цвет и название

TFastLineSeries \*series1 = new TFastLineSeries(Chart1);

series1->Color=clBlue;

series1->Title = "y = x^3";

TFastLineSeries \*series2 = new TFastLineSeries(Chart1);

series2->Color=clRed;

series2->Title = "y = 2x^3";

TFastLineSeries \*series3 = nullptr;

if (ColorBox3->Visible) {

series3 = new TFastLineSeries(Chart1);

series3->Color=clGreen;

series3->Title = "y = x^1.5";

}

TFastLineSeries \*series4 = new TFastLineSeries(Chart1);

series4->Color=clBlack;

series4->Title = "y = (x/2)^3";

// Отрисовать графики в TChart

Chart1->Visible = true;

Chart1->Series[0]->Clear(); // Удалить все серии графика, кроме основной серии

while (Chart1->SeriesCount() > 1) {

delete Chart1->Series[1];

}

// Добавить серии в график

Chart1->AddSeries(series1);

Chart1->AddSeries(series2);

if (series3 != nullptr) {

Chart1->AddSeries(series3);

}

Chart1->AddSeries(series4);

// Шаг для построения графика

double step = 0.1;

// Построить графики для каждой функции для каждого значения аргумента от xmin до xmax

for (double x = xmin; x <= xmax; x += step) {

double y1 = pow(x, 3);

double y2 = 2 \* pow(x, 3);

double y4 = pow(x / 2, 3);

// Добавить точки на графики

series1->AddXY(x, y1);

series2->AddXY(x, y2);

series4->AddXY(x, y4);

if (series3 != nullptr && x >= 0) {

double y3 = pow(x, 1.5);

series3->AddXY(x, y3);

}

}

}

//---------------------------------------------------------------------------

/\*функция для анимации квадрата

\*/

// Объявление переменных для хранения центра квадрата и угла поворота

int centerX = 0;

int centerY = 0;

int rotationAngle = 0;

void \_\_fastcall TForm2::Button6Click(TObject \*Sender)

{

Chart1->Visible=false;

Timer2->Enabled =false;

Timer1->Enabled =true;

// Получаем размеры PaintBox

int paintBoxWidth = PaintBox1->Width;

int paintBoxHeight = PaintBox1->Height;

int x = StrToIntDef(Edit1->Text, 0);

int y = StrToIntDef(Edit2->Text, 0);

// Вычисляем координаты центра квадрата

centerX = paintBoxWidth / 2;

centerY = paintBoxHeight / 2;

PaintBox1->Invalidate();

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm2::Timer1Timer(TObject \*Sender)

{

// Увеличение угла поворота

rotationAngle += 5; // увеличиваем на 5 градусов каждый тик таймера

// Рисование квадрата с учетом поворота

int sideLength = 150; // Длина стороны квадрата

TPoint points[4]; // Массив точек для рисования квадрата

// Вычисление координат вершин квадрата после поворота

double angleRad = rotationAngle \* M\_PI / 180.0; // Перевод угла из градусов в радианы

points[0] = TPoint(centerX + cos(angleRad) \* sideLength / 2, centerY + sin(angleRad) \* sideLength / 2);

points[1] = TPoint(centerX + cos(angleRad + M\_PI / 2) \* sideLength / 2, centerY + sin(angleRad + M\_PI / 2) \* sideLength / 2);

points[2] = TPoint(centerX + cos(angleRad + M\_PI) \* sideLength / 2, centerY + sin(angleRad + M\_PI) \* sideLength / 2);

points[3] = TPoint(centerX + cos(angleRad + 3 \* M\_PI / 2) \* sideLength / 2, centerY + sin(angleRad + 3 \* M\_PI / 2) \* sideLength / 2);

// Очистка канвы

PaintBox1->Canvas->Brush->Color = clWhite;

PaintBox1->Canvas->FillRect(PaintBox1->ClientRect);

// Рисование квадрата на канве

PaintBox1->Canvas->Pen->Color = clBlack;

PaintBox1->Canvas->Polygon(points, 3);

}

//---------------------------------------------------------------------------

/\*Вызов кардиоиды

\*/

void \_\_fastcall TForm2::Button5Click(TObject \*Sender)

{

Form4->Show();

}

/\*Запуск смены картинок

\*/

void \_\_fastcall TForm2::Button8Click(TObject \*Sender)

{

Form5->Show();

}

Form4

#include <vcl.h>

#pragma hdrstop

#include "Unit4.h"

//---------------------------------------------------------------------------

#pragma package(smart\_init)

#pragma resource "\*.dfm"

TForm4 \*Form4;

//---------------------------------------------------------------------------

\_\_fastcall TForm4::TForm4(TComponent\* Owner)

: TForm(Owner)

{

}

//---------------------------------------------------------------------------

/\* Функция для отрисовки графика кардиоиды

\*/

void \_\_fastcall TForm4::Button1Click(TObject \*Sender)

{

PaintBox1->Canvas->Brush->Color = clWhite;

PaintBox1->Canvas->FillRect(PaintBox1->ClientRect);

int centerX = PaintBox1->Width / 2, centerY = PaintBox1->Height / 2;

int Len1 =25;

PaintBox1->Canvas->Pen->Color = clBlack;

PaintBox1->Canvas->MoveTo(0, centerY);

PaintBox1->Canvas->LineTo(PaintBox1->Width, centerY);

PaintBox1->Canvas->MoveTo(centerX, 0);

PaintBox1->Canvas->LineTo(centerX, PaintBox1->Height);

PaintBox1->Canvas->Pen->Style = psDot;

int dopY = centerY - Len1;

while (dopY > 0) {

PaintBox1->Canvas->MoveTo(0, dopY);

PaintBox1->Canvas->LineTo(PaintBox1->Width, dopY);

dopY = dopY - Len1;

}

dopY = centerY + Len1;

while (dopY < PaintBox1->Height) {

PaintBox1->Canvas->MoveTo(0, dopY);

PaintBox1->Canvas->LineTo(PaintBox1->Width, dopY);

dopY = dopY + Len1;

}

int dopX = centerX - Len1;

while (dopX > 0) {

PaintBox1->Canvas->MoveTo(dopX, 0);

PaintBox1->Canvas->LineTo(dopX, PaintBox1->Height);

dopX = dopX - Len1;

}

dopX = centerX + Len1;

while (dopX < PaintBox1->Width) {

PaintBox1->Canvas->MoveTo(dopX, 0);

PaintBox1->Canvas->LineTo(dopX, PaintBox1->Height);

dopX = dopX + Len1;

}

PaintBox1->Canvas->Pen->Style = psSolid;

PaintBox1->Canvas->Font->Size = 8;

PaintBox1->Canvas->Font->Color = clBlack;

int stepY = Len1;

int labelY = 1;

while (centerY - stepY > 0) {

PaintBox1->Canvas->TextOut(centerX + 5, centerY - stepY - 10, IntToStr(labelY));

PaintBox1->Canvas->TextOut(centerX + 5, centerY + stepY - 10, IntToStr(labelY));

stepY += Len1;

labelY += 1;

}

int stepX = Len1;

int labelX = 1;

while (centerX - stepX > 0) {

PaintBox1->Canvas->TextOut(centerX - stepX - 10, centerY + 5, IntToStr(labelX));

PaintBox1->Canvas->TextOut(centerX + stepX - 10, centerY + 5, IntToStr(labelX));

stepX += Len1;

labelX += 1;

}

int a = StrToInt(Edit2->Text); // Получить значение параметра "a" из Edit3 и преобразовать в целое число

double tStep = StrToFloat(Edit3->Text); // Получить значение шага "t" из Edit5 и преобразовать в вещественное число

double x, y, t;

int prevX = 0, prevY = 0;

bool firstPoint = true;

// Нарисовать кардиоид в PaintBox1

for (t = 0; t < 2 \* M\_PI; t += tStep) {

// Вычислить координаты кардиоиды для данного значения "t" с помощью добавленных функций

double x = a \* cos(t) \* (1 + cos(t));

double y = a \* sin(t) \* (1 + cos(t));

int scaledX = round(x \* 10) + PaintBox1->Width / 2; // Масштабировать координаты

int scaledY = round(y \* 10) + PaintBox1->Height / 2;

PaintBox1->Canvas->Pixels[scaledX][scaledY] = clBlack; // Нарисовать точку

PaintBox1->Canvas->Pen->Color = ColorBox1->Selected; // Установить цвет пера

if (!firstPoint) {

PaintBox1->Canvas->MoveTo(prevX, prevY);

PaintBox1->Canvas->LineTo(scaledX, scaledY); // Нарисовать линию

} else {

firstPoint = false;

}

prevX = scaledX;

prevY = scaledY;

}

}

void \_\_fastcall TForm4::ПостроитьClick(TObject \*Sender)

{

PaintBox1->Canvas->Brush->Color = clWhite;

PaintBox1->Canvas->FillRect(PaintBox1->ClientRect);

int centerX = PaintBox1->Width / 2, centerY = PaintBox1->Height / 2;

int Len1 =25;

PaintBox1->Canvas->Pen->Color = clBlack;

PaintBox1->Canvas->MoveTo(0, centerY);

PaintBox1->Canvas->LineTo(PaintBox1->Width, centerY);

PaintBox1->Canvas->MoveTo(centerX, 0);

PaintBox1->Canvas->LineTo(centerX, PaintBox1->Height);

PaintBox1->Canvas->Pen->Style = psDot;

int dopY = centerY - Len1;

while (dopY > 0) {

PaintBox1->Canvas->MoveTo(0, dopY);

PaintBox1->Canvas->LineTo(PaintBox1->Width, dopY);

dopY = dopY - Len1;

}

dopY = centerY + Len1;

while (dopY < PaintBox1->Height) {

PaintBox1->Canvas->MoveTo(0, dopY);

PaintBox1->Canvas->LineTo(PaintBox1->Width, dopY);

dopY = dopY + Len1;

}

int dopX = centerX - Len1;

while (dopX > 0) {

PaintBox1->Canvas->MoveTo(dopX, 0);

PaintBox1->Canvas->LineTo(dopX, PaintBox1->Height);

dopX = dopX - Len1;

}

dopX = centerX + Len1;

while (dopX < PaintBox1->Width) {

PaintBox1->Canvas->MoveTo(dopX, 0);

PaintBox1->Canvas->LineTo(dopX, PaintBox1->Height);

dopX = dopX + Len1;

}

PaintBox1->Canvas->Pen->Style = psSolid;

PaintBox1->Canvas->Font->Size = 8;

PaintBox1->Canvas->Font->Color = clBlack;

int stepY = Len1;

int labelY = 1;

while (centerY - stepY > 0) {

PaintBox1->Canvas->TextOut(centerX + 5, centerY - stepY - 10, IntToStr(labelY));

PaintBox1->Canvas->TextOut(centerX + 5, centerY + stepY - 10, IntToStr(labelY));

stepY += Len1;

labelY += 1;

}

int stepX = Len1;

int labelX = 1;

while (centerX - stepX > 0) {

PaintBox1->Canvas->TextOut(centerX - stepX - 10, centerY + 5, IntToStr(labelX));

PaintBox1->Canvas->TextOut(centerX + stepX - 10, centerY + 5, IntToStr(labelX));

stepX += Len1;

labelX += 1;

}

int a = StrToInt(Edit2->Text); // Получить значение параметра "a" из Edit3 и преобразовать в целое число

double tStep = StrToFloat(Edit3->Text); // Получить значение шага "t" из Edit5 и преобразовать в вещественное число

double x, y, t;

int prevX = 0, prevY = 0;

bool firstPoint = true;

// Нарисовать кардиоид в PaintBox1

for (t = 0; t < 2 \* M\_PI; t += tStep) {

// Вычислить координаты кардиоиды для данного значения "t" с помощью добавленных функций

double x = a \* cos(t) \* (1 + cos(t));

double y = a \* sin(t) \* (1 + cos(t));

int scaledX = round(x \* 10) + PaintBox1->Width / 2; // Масштабировать координаты

int scaledY = round(y \* 10) + PaintBox1->Height / 2;

PaintBox1->Canvas->Pixels[scaledX][scaledY] = clBlack; // Нарисовать точку

PaintBox1->Canvas->Pen->Color = ColorBox1->Selected; // Установить цвет пера

if (!firstPoint) {

PaintBox1->Canvas->MoveTo(prevX, prevY);

PaintBox1->Canvas->LineTo(scaledX, scaledY); // Нарисовать линию

} else {

firstPoint = false;

}

prevX = scaledX;

prevY = scaledY;

}

Form5

#include <vcl.h>

#pragma hdrstop

#include "Unit5.h"

//---------------------------------------------------------------------------

#pragma package(smart\_init)

#pragma resource "\*.dfm"

TForm5 \*Form5;

//---------------------------------------------------------------------------

\_\_fastcall TForm5::TForm5(TComponent\* Owner)

: TForm(Owner)

{

}

//---------------------------------------------------------------------------

/\* функция для смены картинок

\*/

int ImageIndex=1;

void \_\_fastcall TForm5::Timer1Timer(TObject \*Sender)

{

Timer1->Interval = 250;

if (ImageIndex == 1)

{

Image1->Picture->LoadFromFile("C:\\Users\\User\\Desktop\\Image1\\1.bmp");

++ImageIndex;

}

else if (ImageIndex == 2)

{

Image1->Picture->LoadFromFile("C:\\Users\\User\\Desktop\\Image1\\2.bmp");

++ImageIndex;

}

else if (ImageIndex == 3)

{

Image1->Picture->LoadFromFile("C:\\Users\\User\\Desktop\\Image1\\3.bmp");

++ImageIndex;

}

else if (ImageIndex == 4)

{

Image1->Picture->LoadFromFile("C:\\Users\\User\\Desktop\\Image1\\4.bmp");

++ImageIndex;

}

else

{

ImageIndex = 1; // Сбрасываем индекс обратно на 1, если достигли конца списка

Image1->Picture->LoadFromFile("C:\\Users\\User\\Desktop\\Image1\\1.bmp");

}

}

Тестирование программы

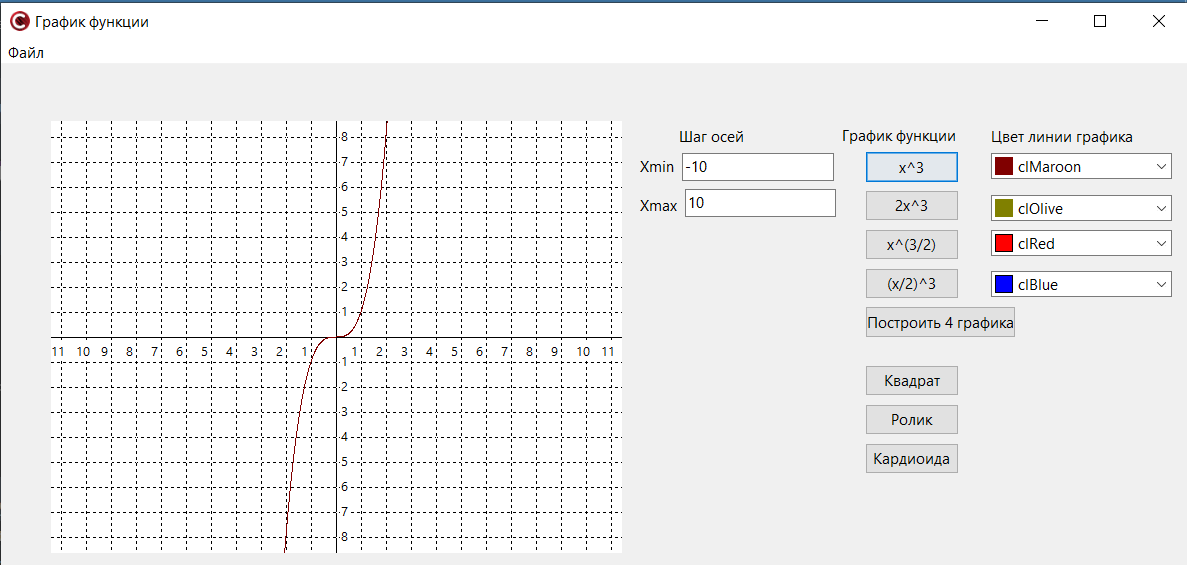


Рис 2.- Результат теста №1

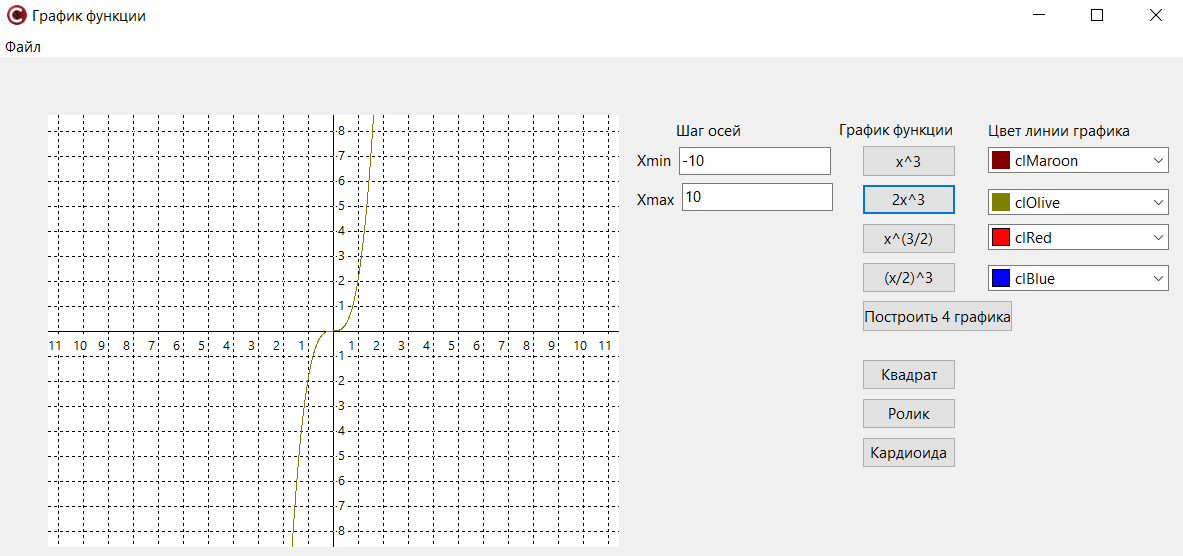


Рис 3.- Результат теста №1

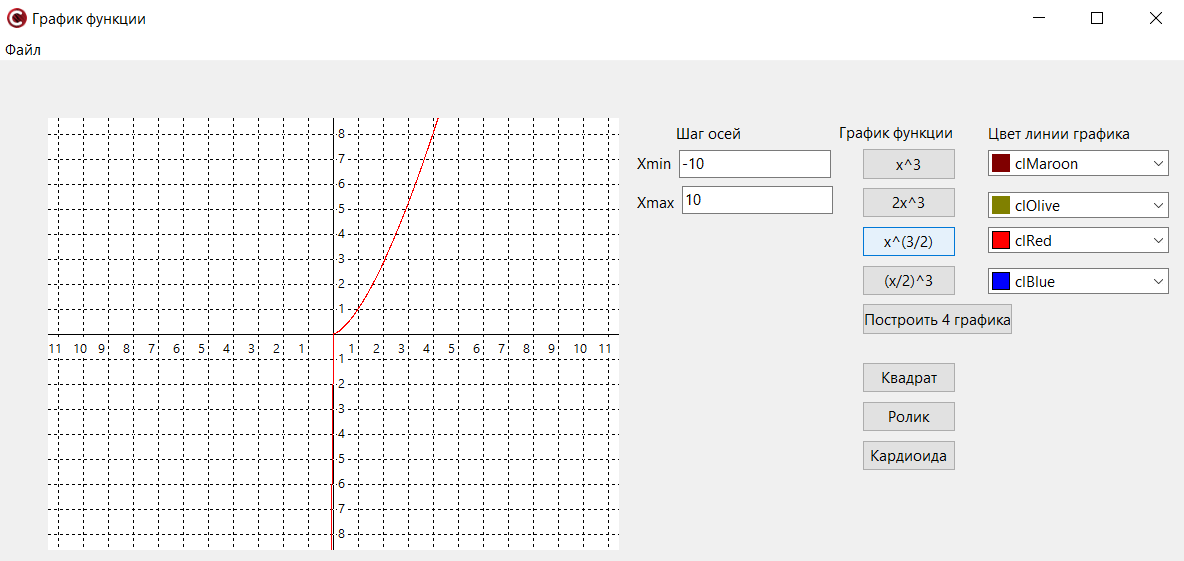


Рис 4.- Результат теста №1

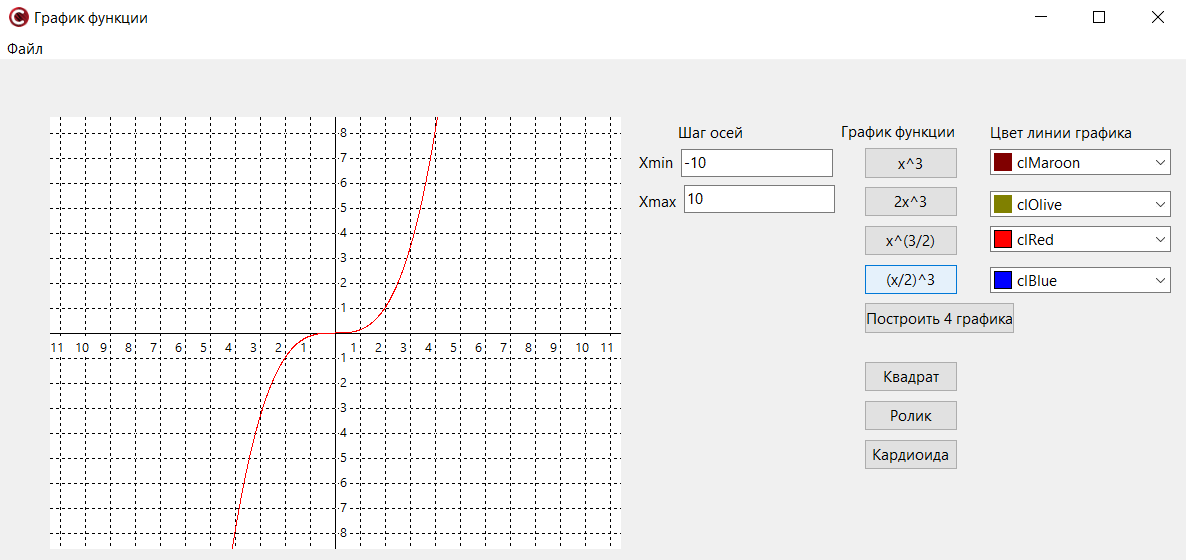


Рис 5.- Результат теста №1

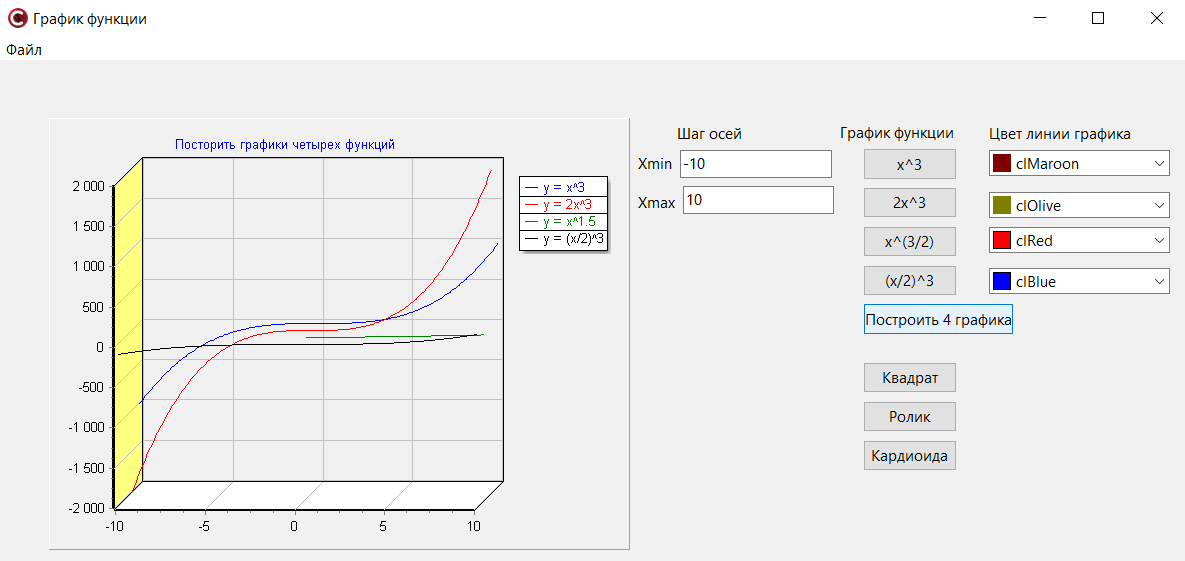


Рис 6.- Результат теста №2

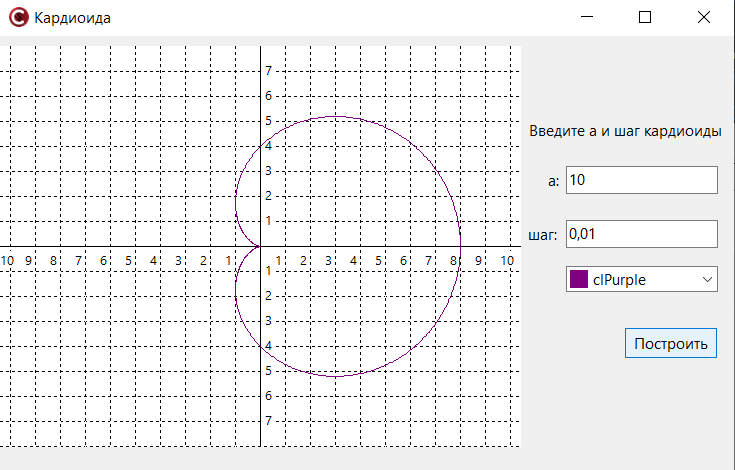


Рис 7.- Результат теста №3

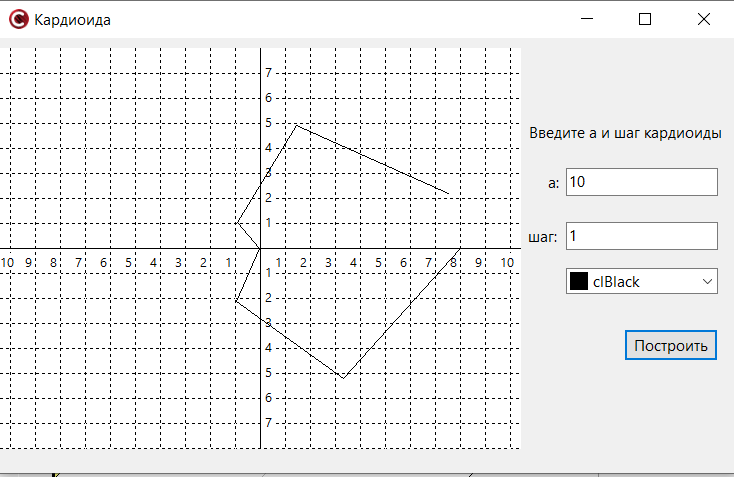


Рис 8.- Результат теста №4

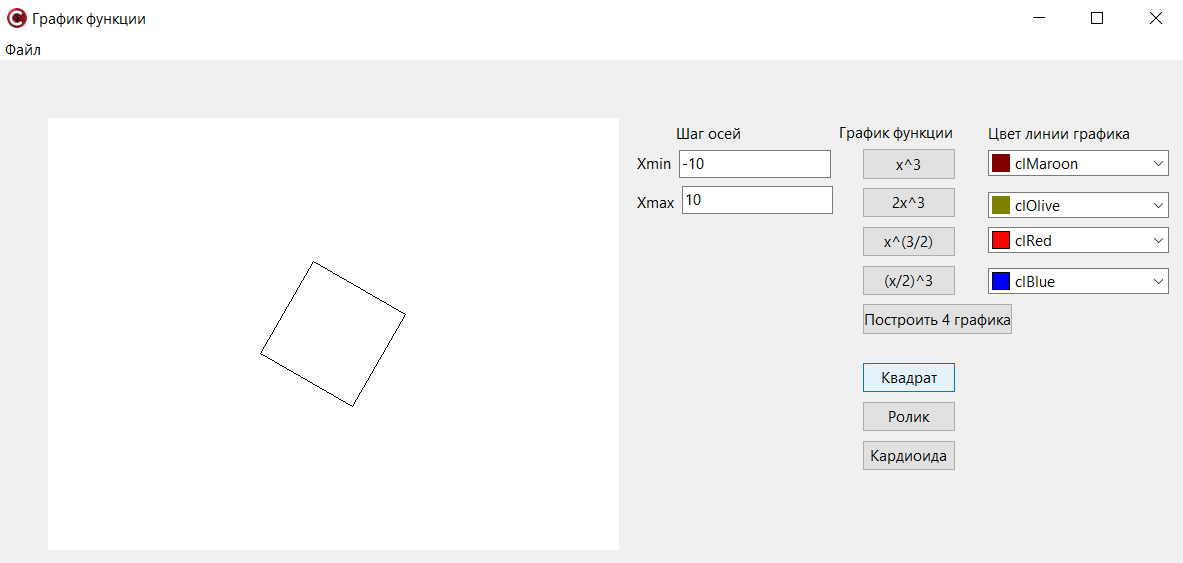


Рис 9.- Результат теста №5

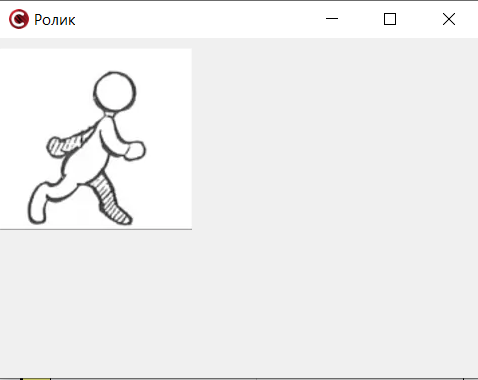


Рис 10.- Результат теста №6