

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение высшего
профессионального образования
«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ АЭРОКОСМИЧЕСКОГО
ПРИБОРОСТРОЕНИЯ»

КАФЕДРА КОМПЬЮТЕРНЫХ ТЕХНОЛОГИЙ И ПРОГРАММНОЙ ИНЖЕНЕРИИ

ОТЧЕТ
ЗАЩИЩЕН С ОЦЕНКОЙ
РУКОВОДИТЕЛЬ

кандидат технических наук, доцент
должность, уч. степень, звание

подпись, дата

С.В.Щекин
инициалы, фамилия

ОТЧЕТ О ВЫПОЛНЕНИИ КОНТРОЛЬНОЙ РАБОТЫ

по дисциплине: КОМПЬЮТЕРНАЯ ГРАФИКА

РАБОТУ ВЫПОЛНИЛ
СТУДЕНТ ГР. Z7431

20.12.2019
подпись, дата

М.Д.Семочкин
инициалы, фамилия

Студ. билет 2014/1054

Санкт-Петербург
2019


Содержание

1. Титульный лист
2. Содержание
3. Задание
4. Используемые инструменты разработки
5. Описание программы
6. Текст программы
7. Результат работы программы
8. Вывод

Задание

Целью работы является освоение практических навыков реализации графических приложений с двумерным интерфейсом с использованием стандартных вызовов библиотек и реализации простых геометрических преобразований, используемых в компьютерной графике.

Используя библиотеку GDI+ реализовать построение геометрической фигуры и аффинные преобразования на плоскости в соответствии с заданным вариантом.

8		1 – поворот по часовой стрелке 2 – поворот против часовой стрелки 3 – перенос влево 4 – перенос вправо
---	---	---

Используемые инструменты разработки

IDE – Microsoft Visual C# 2010 Express

ОС – Windows 7 x64

Описание программы

Написанная программа рисует полигон в форме стрелки, точки этого полигона хранятся в свойстве класса. Добавляются обработчики событий нажатия на клавиши клавиатуры, которыми можно вызывать функции, выполняющие преобразования для точек полигона и вызывающие его перерисовку:

- 1) Поворот по часовой стрелке
- 2) Поворот против часовой стрелки
- 3) Перенос влево
- 4) Перенос вправо

Текст программы

// Программа рисует стрелочку и позволяет ее двигать и поворачивать

```
using System;
using System.Windows.Forms;
using System.Drawing;
using System.Drawing.Drawing2D;

public class Draw : Form
{
    private Point[] points =
    {
        new Point(200, 300), new Point(300, 300), new Point(300, 250),
        new Point(400, 325), new Point(300, 400), new Point(300, 350),
        new Point(200, 350), new Point(200, 300)
    };

    public Draw()
    {
        this.Text = "Рисуем фигуру";
        this.Size = new Size(600, 600);
        this.Paint += new PaintEventHandler(Draw_Graphics);

        this.KeyPress += new KeyPressEventHandler(keyPressHandler);

        MessageBox.Show("Нажмите:\n 1 - для поворота по часовой стрелке\n 2 - для поворота
против часовой " +
            "стрелки\n 3 - для переноса влево\n 4 - для переноса вправо");
    }

    public void Draw_Graphics(object sender, PaintEventArgs e)
    {
        // Получаем объект Graphics через идентификатор окна
        Graphics g = Graphics.FromHwnd(this.Handle);
        // А также ширину и высоту области для рисования
        int w = ClientSize.Width;
        int h = ClientSize.Height;
        // Очистка формы с помощью закрашенного прямоугольника
        g.FillRectangle(Brushes.White, 0, 0, w, h);

        // Стрелка
        g.DrawPolygon(Pens.Green, points);
    }

    private void keyPressHandler(object sender, KeyPressEventArgs e)
    {
        if (e.KeyChar == 49) rotate("clockwise");
        if (e.KeyChar == 50) rotate("conterclockwise");
        if (e.KeyChar == 51) moveLeft();
        if (e.KeyChar == 52) moveRight();
    }
}
```

```

private void moveRight()
{
    for(int i = 0; i < points.Length; i++)
    {
        points[i].X += 10;
    }
    this.Invalidate();
}

private void moveLeft()
{
    for (int i = 0; i < points.Length; i++)
    {
        points[i].X -= 10;
    }
    this.Invalidate();
}

private void rotate(string direction)
{
    int rotationAngle = 15;
    if (direction == "clockwise") rotationAngle = -15;
    double rotationRadians = rotationAngle * (Math.PI / 180);

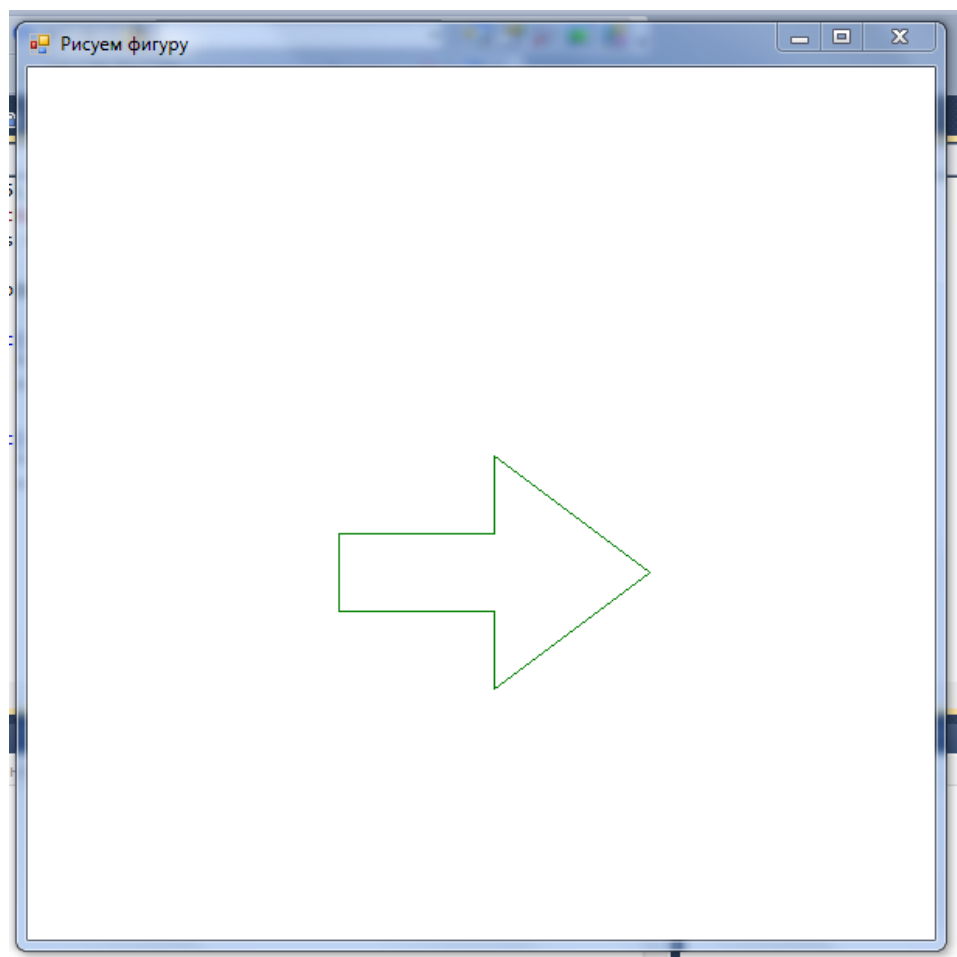
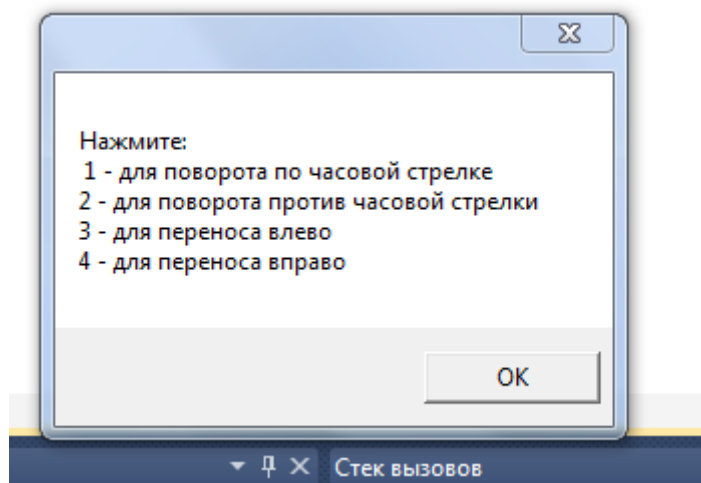
    for (int i = 0; i < points.Length; i++)
    {
        points[i].X = (int) Math.Round(
            ( points[i].X * Math.Cos(rotationRadians)) -
            ( points[i].Y * Math.Sin(rotationRadians))
        );
        points[i].Y = (int) Math.Round(
            ( points[i].X * Math.Sin(rotationRadians)) +
            ( points[i].Y * Math.Cos(rotationRadians))
        );
    }

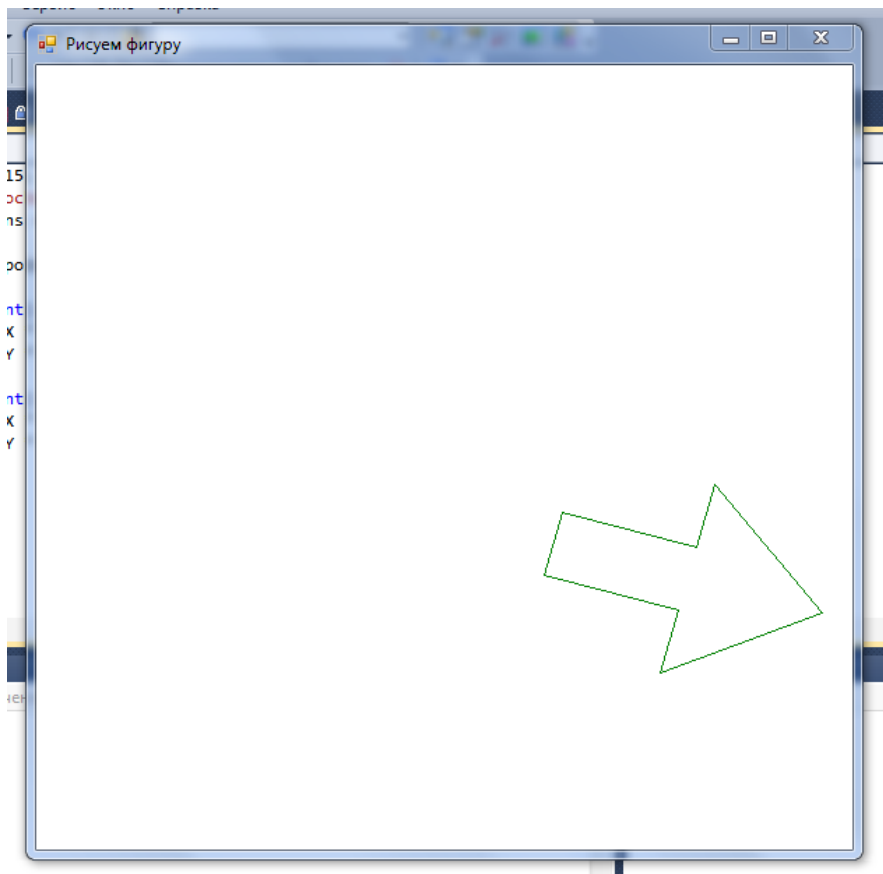
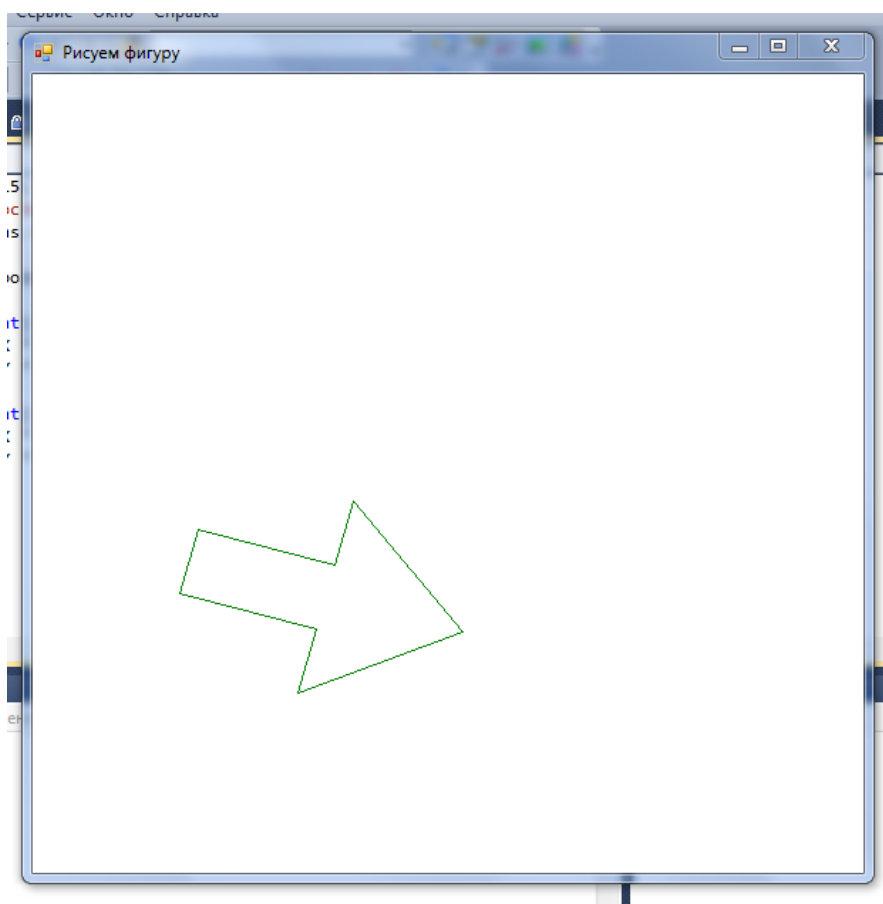
    this.Invalidate();
}

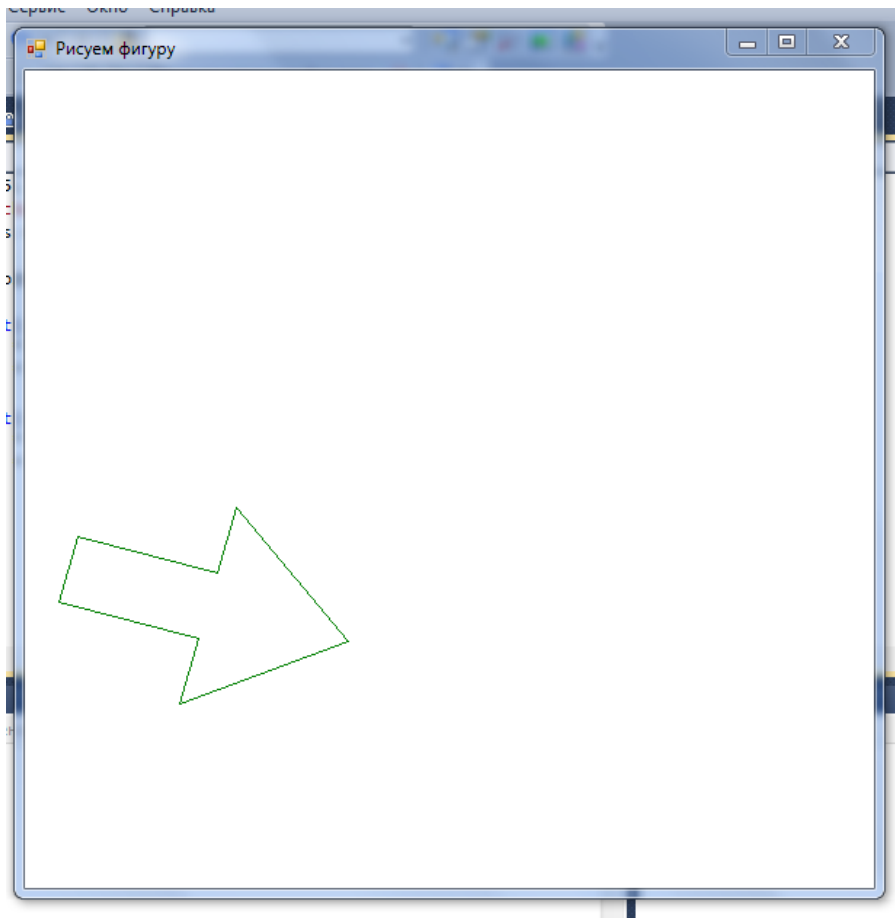
public static void Main()
{
    Application.Run(new Draw());
}
}

```

Результат работы программы







Вывод

Во время выполнения контрольной работы были получены навыки реализации графических приложений с двумерным интерфейсом с использованием стандартных вызовов библиотек и реализации простых геометрических преобразований, используя библиотеку GDI+.