Министерство транспорта Российской Федерации
Федеральное государственное автономное
образовательное учреждение высшего образования
«Российский университет транспорта (МИИТ)»

Институт транспортной техники и систем управления

Кафедра «Управление и защита информации»

Лабораторная работа №2

по дисциплине:

«Методы программирования»

на тему:
«Конструктор ГПИ. Фигуры»

Выполнил: ст. гр. ТКИ-341

Панаргин В.М.

Вариант №5

Проверил: к.т.н., доцент Сафронов А.И.

Москва – 2024 г.

1. Цель работы

«Закрепить навыки разработки визуального пользовательского интерфейса, освоить работу с текстовыми файлами и кодировкой в среде *Microsoft Visual Studio*, научиться реализовывать настройку множественных состояний объектов посредством управления компонентами со внутренней индексацией».

2. Формулировка задачи

«В интегрированной среде разработки *Microsoft Visual Studio* разработать программу в режиме *Windows Forms Application* на языке *Visual C#*, представляющую собой экранную форму, содержащую главное меню, позволяющее:

1. Начать работу с приложением.
2. Прервать работу приложения.
3. Предоставить пользователю справочную информацию о работе с приложением.

Сама программа должна реализовывать вывод в графический элемент управления (например, *PictureBox*) главной экранной формы плоскостную геометрическую фигуру, выбираемую пользователем из списка (вид списка * задаётся вариантом индивидуального задания). Список должен обязательно содержать следующие пункты:

1. «Квадрат»,
2. «Прямоугольный треугольник»,
3. «Эллипс»,
4. «Равнобедренный треугольник»,
5. «Круг»,
6. «Равносторонний треугольник»,
7. «Окружность»,
8. «Ромб»,

9. «Трапеция»,
10. «Параллелограмм»,
11. «Прямоугольник».

Согласно заданию, список должен быть организован в виде кнопок с иконками. Фигуры из списка должны быть расположены в следующем порядке: 5, 4, 8, 2, 3, 1, 10, 9, 6, 11, 7. Так же необходимо поместить сложное комбинированное изображение «Прицел» на 8 позицию. Таким образом, фигуры должны быть расположены в следующем порядке: круг, равнобедренный треугольник, ромб, прямоугольный треугольник, эллипс, квадрат, параллелограмм, прицел, трапеция, равносторонний треугольник, прямоугольник, окружность.

3. Составление диаграммы классов, входящих в состав решения.

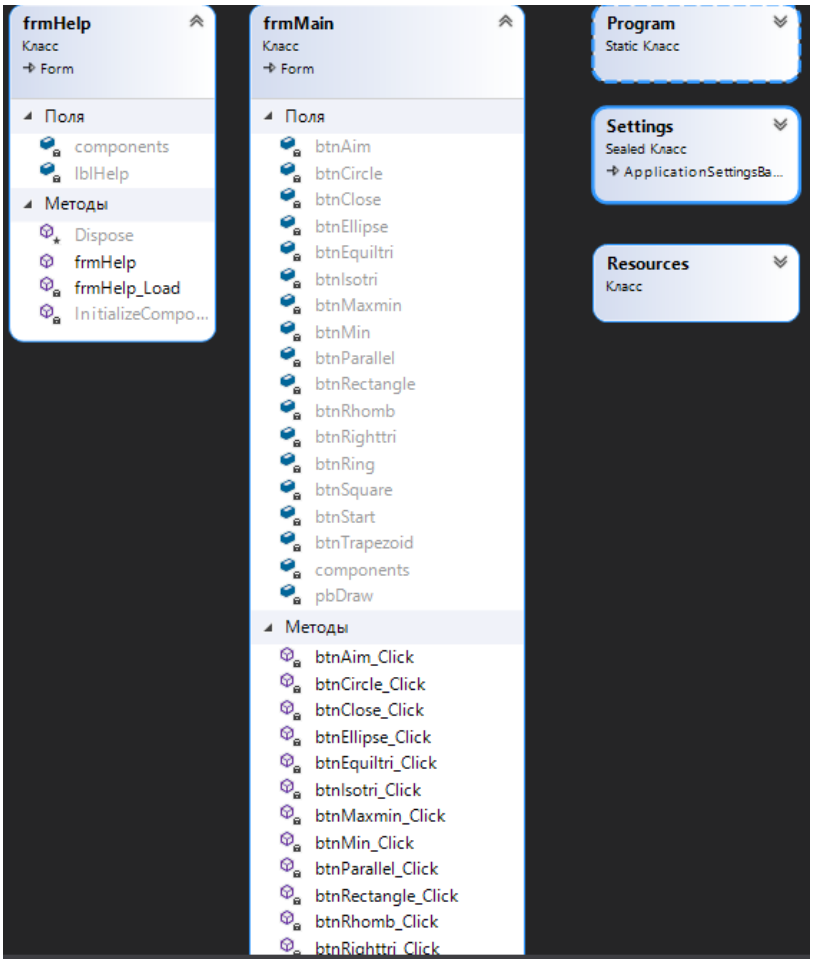

Рисунок 1 – Диаграмма классов

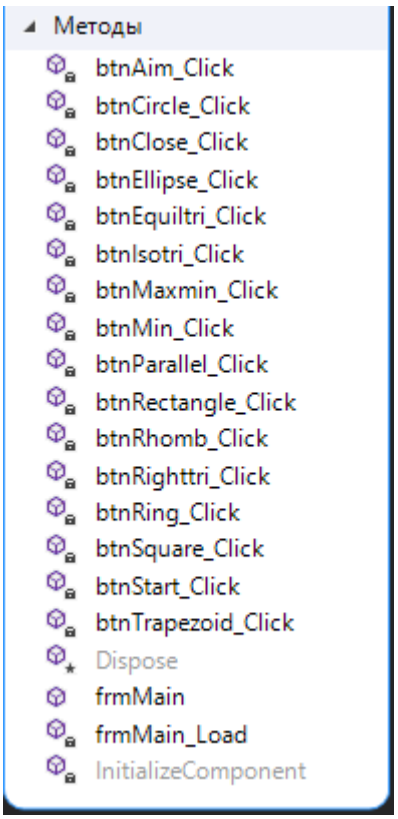

Рисунок 2 – Диаграмма классов (продолжение)

4. Составление сети Петри запрограммированного технологического процесса.

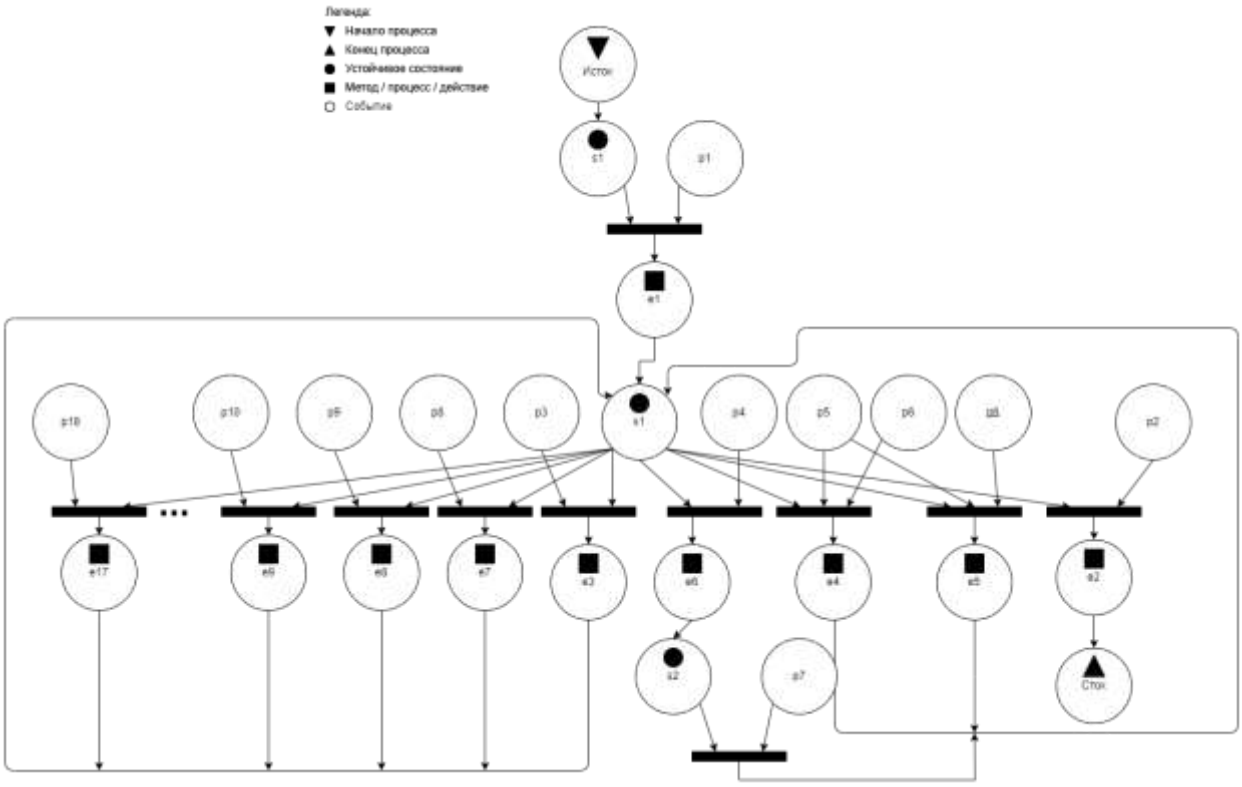

Рисунок 3 – Сеть Петри основной формы

Описание сети Петри

- состояния (states)

  s1 – форма ожидает действий пользователя

  s2 – форма находится в свернутом состоянии

- действия (effects)

  e1 – разблокируются кнопки взаимодействия;

  кнопка «Начало работы с приложением» деактивируется

e2 – закрытие формы

e3 – отрисовка фигуры круг

e4 – форма минимизируется

e5 – форма максимизируется

e6 – форма сворачивается

e7 – отрисовка фигуры равнобедренный треугольник

e8 – отрисовка фигуры ромб

e9 – отрисовка фигуры прямоугольный треугольник

…

e17 – отрисовка фигуры окружность

- события (prompts)

p1 – нажата кнопка «Начало работы с приложением»

p2 – нажата кнопка закрытия

p3 – нажата кнопка отрисовки фигуры круг

p4 – нажата кнопка «свернуть»

p5 – нажата кнопка «максимизировать/минимизировать»

p6 – форма максимизирована

p7 – пользователь разворачивает форму

p8 – нажата кнопка отрисовки фигуры равнобедренный треугольник

p9 – нажата кнопка отрисовки фигуры ромб

p10 – нажата кнопка отрисовки фигуры прямоугольный треугольник

…

p18 – нажата кнопка отрисовки фигуры окружность

5. Составление схем алгоритмов методов в составе решения, отмеченных на сети Петри в качестве «эффектов» (метка ■).

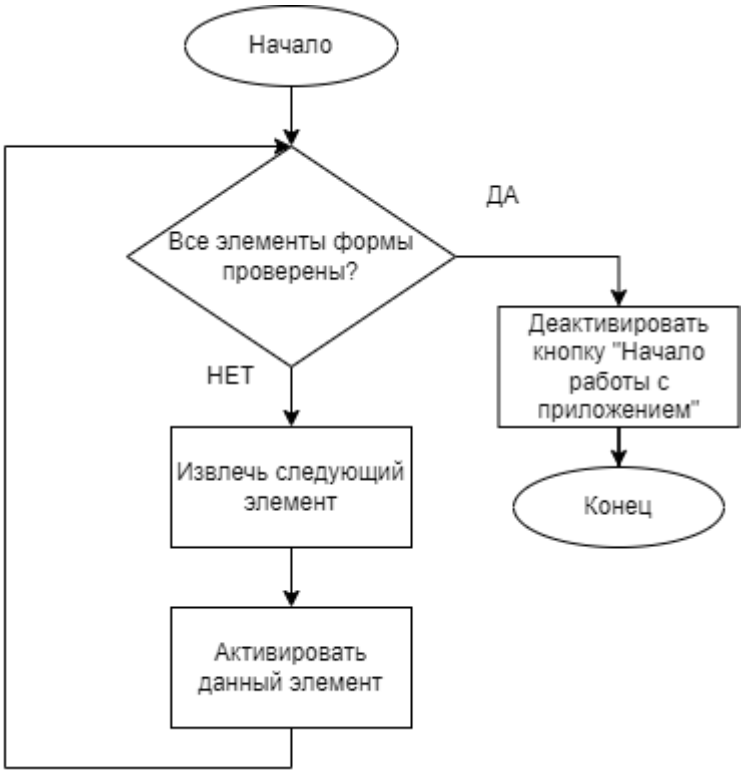

Рисунок 4 – Алгоритм кнопки "Начало работы с приложением"

Алгоритм отрисовки фигуры «Круг»:

- Создать поле 400x400
- Выбрать кисть с цветом красный
- Залить эллипс с координатой верхнего левого угла (100;100) и размерами (200;200)

Алгоритм отрисовки фигуры «Равнобедренный треугольник»:

- Создать поле 400x400
- Выбрать ручку с цветом красный и толщиной 3
- Нарисовать полигон с координатами точек (200, 100), (150, 300), (250, 300)

Алгоритм отрисовки фигуры «Ромб»:

- Создать поле 400x400
- Выбрать ручку с цветом красный и толщиной 3
- Нарисовать полигон с координатами точек (200, 100), (150, 200), (200, 300), (250, 200)

Алгоритм отрисовки фигуры «Прямоугольный треугольник»:

- Создать поле 400x400
- Выбрать ручку с цветом красный и толщиной 3
- Нарисовать полигон с координатами точек
  (100, 100), (100, 300), (300, 300)

Алгоритм отрисовки фигуры «Эллипс»:

- Создать поле 400x400
- Выбрать ручку с цветом красный и толщиной 3
- Нарисовать эллипс с координатой верхнего левого угла (150, 100) и размерами (100, 200)

Алгоритм отрисовки фигуры «Квадрат»:

- Создать поле 400x400
- Выбрать ручку с цветом красный и толщиной 3
- Нарисовать полигон с координатами точек
  (100, 100), (100, 300), (300, 300), (300, 100)

Алгоритм отрисовки фигуры «Окружность»:

- Создать поле 400x400
- Выбрать ручку с цветом красный и толщиной 3
- Нарисовать эллипс с координатой верхнего левого угла (100;100) и размерами (200;200)

Алгоритм отрисовки фигуры «Параллелограмм»:

- Создать поле 400x400
- Выбрать ручку с цветом красный и толщиной 3
- Нарисовать полигон с координатами точек
  (150, 150), (100, 250), (250, 250), (300, 150)

Алгоритм отрисовки фигуры «Трапеция»:

- Создать поле 400x400
- Выбрать ручку с цветом красный и толщиной 3
- Нарисовать полигон с координатами точек
  (150, 150), (250, 150), (300, 250), (100, 250)

Алгоритм отрисовки фигуры «Прицел»:

- Создать поле 400x400
- Выбрать ручку с цветом красный и толщиной 3
- Нарисовать эллипс с координатой верхнего левого угла (100;100) и размерами (200;200)
- Нарисовать эллипс с координатой верхнего левого угла (150;150) и размерами (100;100)
- Нарисовать линию, соединяющую точки (200; 100) и (200; 300)
- Нарисовать линию, соединяющую точки (100; 200) и (300; 200)
- Выбрать кисть с цветом красный
- Залить эллипс с координатой верхнего левого угла (195;195) и размерами (10;10)

Алгоритм отрисовки фигуры «Равносторонний треугольник»:

- Создать поле 400x400
- Выбрать ручку с цветом красный и толщиной 3
- Нарисовать полигон с координатами точек (200, 300-100*$\sqrt{3}$), (100, 300), (300, 300)

Алгоритм отрисовки фигуры «Прямоугольник»:

- Создать поле 400x400
- Выбрать ручку с цветом красный и толщиной 3
- Нарисовать полигон с координатами точек (100, 150), (100, 250), (300, 250), (300, 150)

6. Подбор тестовых примеров.

Тестовые примеры:

- Запустить программу (проверка открытия основной и дочерней формы)
- Нажать кнопку «Начало работы с приложением»
- Проверить корректность отрисовки каждой из фигур
- Проверить кнопку «максимизировать/минимизировать»

7. Листинг (код) составленного программного обеспечения.

Основная форма:

```
public partial class frmMain : Form
{
    public frmMain()
    {
        InitializeComponent();
    }

    private void frmMain_Load(object sender, EventArgs e)
    {
        frmHelp f1 = new frmHelp();
        f1.Show(this);
    }

    private void btnClose_Click(object sender, EventArgs e)
    {
        this.Close();
    }

    private void btnMaxmin_Click(object sender, EventArgs e)
    {
        if (this.WindowState == FormWindowState.Normal)
        {
            this.WindowState = FormWindowState.Maximized;
        }
        else
        {
            this.WindowState = FormWindowState.Normal;
        }
    }

    private void btnMin_Click(object sender, EventArgs e)
    {
        this.WindowState = FormWindowState.Minimized;
    }

    private void btnCircle_Click(object sender, EventArgs e)
    {
        Bitmap bitmap = new Bitmap(400, 400,
System.Drawing.Imaging.PixelFormat.Format32bppPArgb);
        Graphics graphics = Graphics.FromImage(bitmap);
        SolidBrush brush = new SolidBrush(Color.Red);
        graphics.FillEllipse(brush, new Rectangle(100, 100, 200, 200));
        pbDraw.Image = (Image)bitmap;
    }

    private void btnIsotri_Click(object sender, EventArgs e)
    {
        Bitmap bitmap = new Bitmap(400, 400,
System.Drawing.Imaging.PixelFormat.Format32bppPArgb);
        Graphics graphics = Graphics.FromImage(bitmap);
        Pen pen = new Pen(Color.Red, 3);
        graphics.DrawPolygon(pen, new PointF[3] {new PointF(200, 100), new PointF(150,
300), new PointF(250, 300)});
        pbDraw.Image = (Image)bitmap;
    }

    private void btnRhomb_Click(object sender, EventArgs e)
    {
        Bitmap bitmap = new Bitmap(400, 400,
System.Drawing.Imaging.PixelFormat.Format32bppPArgb);
```

```
        Graphics graphics = Graphics.FromImage(bitmap);
        Pen pen = new Pen(Color.Red, 3);
        graphics.DrawPolygon(pen, new PointF[4] { new PointF(200, 100), new PointF(150,
200), new PointF(200, 300), new PointF(250, 200)});
        pbDraw.Image = (Image)bitmap;
    }

    private void btnRighttri_Click(object sender, EventArgs e)
    {
        Bitmap bitmap = new Bitmap(400, 400,
System.Drawing.Imaging.PixelFormat.Format32bppPArgb);
        Graphics graphics = Graphics.FromImage(bitmap);
        Pen pen = new Pen(Color.Red, 3);
        graphics.DrawPolygon(pen, new PointF[3] { new PointF(100, 100), new PointF(100,
300), new PointF(300, 300) });
        pbDraw.Image = (Image)bitmap;
    }

    private void btnEllipse_Click(object sender, EventArgs e)
    {
        Bitmap bitmap = new Bitmap(400, 400,
System.Drawing.Imaging.PixelFormat.Format32bppPArgb);
        Graphics graphics = Graphics.FromImage(bitmap);
        Pen pen = new Pen(Color.Red, 3);
        graphics.DrawEllipse(pen, 150, 100, 100, 200);
        pbDraw.Image = (Image)bitmap;
    }

    private void btnSquare_Click(object sender, EventArgs e)
    {
        Bitmap bitmap = new Bitmap(400, 400,
System.Drawing.Imaging.PixelFormat.Format32bppPArgb);
        Graphics graphics = Graphics.FromImage(bitmap);
        Pen pen = new Pen(Color.Red, 3);
        graphics.DrawPolygon(pen, new PointF[4] { new PointF(100, 100), new PointF(100,
300), new PointF(300, 300), new PointF(300, 100) });
        pbDraw.Image = (Image)bitmap;
    }

    private void btnParallel_Click(object sender, EventArgs e)
    {
        Bitmap bitmap = new Bitmap(400, 400,
System.Drawing.Imaging.PixelFormat.Format32bppPArgb);
        Graphics graphics = Graphics.FromImage(bitmap);
        Pen pen = new Pen(Color.Red, 3);
        graphics.DrawPolygon(pen, new PointF[4] { new PointF(150, 150), new PointF(100,
250), new PointF(250, 250), new PointF(300, 150) });
        pbDraw.Image = (Image)bitmap;
    }

    private void btnTrapezoid_Click(object sender, EventArgs e)
    {
        Bitmap bitmap = new Bitmap(400, 400,
System.Drawing.Imaging.PixelFormat.Format32bppPArgb);
        Graphics graphics = Graphics.FromImage(bitmap);
        Pen pen = new Pen(Color.Red, 3);
        graphics.DrawPolygon(pen, new PointF[4] { new PointF(150, 150), new PointF(250,
150), new PointF(300, 250), new PointF(100, 250) });
        pbDraw.Image = (Image)bitmap;
    }

    private void btnAim_Click(object sender, EventArgs e)
    {
        Bitmap bitmap = new Bitmap(400, 400,
System.Drawing.Imaging.PixelFormat.Format32bppPArgb);
```

```
        Graphics graphics = Graphics.FromImage(bitmap);
        Pen pen = new Pen(Color.Red, 2);
        graphics.DrawEllipse(pen, new Rectangle(100, 100, 200, 200));
        graphics.DrawEllipse(pen, new Rectangle(150, 150, 100, 100));
        graphics.DrawLine(pen, 200, 100, 200, 300);
        graphics.DrawLine(pen, 100, 200, 300, 200);
        SolidBrush brush = new SolidBrush(Color.Red);
        graphics.FillEllipse(brush, new Rectangle(195, 195, 10, 10));
        pbDraw.Image = (Image)bitmap;
    }

    private void btnEquiltri_Click(object sender, EventArgs e)
    {
        Bitmap bitmap = new Bitmap(400, 400,
System.Drawing.Imaging.PixelFormat.Format32bppPArgb);
        Graphics graphics = Graphics.FromImage(bitmap);
        Pen pen = new Pen(Color.Red, 3);
        graphics.DrawPolygon(pen, new PointF[3] { new PointF(200, 300-
100*(float)Math.Sqrt(3)), new PointF(100, 300), new PointF(300, 300) });
        pbDraw.Image = (Image)bitmap;
    }

    private void btnRectangle_Click(object sender, EventArgs e)
    {
        Bitmap bitmap = new Bitmap(400, 400,
System.Drawing.Imaging.PixelFormat.Format32bppPArgb);
        Graphics graphics = Graphics.FromImage(bitmap);
        Pen pen = new Pen(Color.Red, 3);
        graphics.DrawPolygon(pen, new PointF[4] { new PointF(100, 150), new PointF(100,
250), new PointF(300, 250), new PointF(300, 150) });
        pbDraw.Image = (Image)bitmap;
    }

    private void btnRing_Click(object sender, EventArgs e)
    {
        Bitmap bitmap = new Bitmap(400, 400,
System.Drawing.Imaging.PixelFormat.Format32bppPArgb);
        Graphics graphics = Graphics.FromImage(bitmap);
        Pen pen = new Pen(Color.Red, 3);
        graphics.DrawEllipse(pen, new Rectangle(100, 100, 200, 200));
        pbDraw.Image = (Image)bitmap;
    }

    private void btnStart_Click(object sender, EventArgs e)
    {
        foreach (Control control in this.Controls)
        {
            control.Visible = true;
        }
        btnStart.Visible = false;
    }
}
```

Дополнительная форма:

```
public partial class frmHelp : Form
{
    public frmHelp()
    {
        InitializeComponent();
    }

    private void frmHelp_Load(object sender, EventArgs e)
    {
```

```
        StreamReader sr = new StreamReader("E:\\Методы программирования\\Задание
2\\Task2\\HelpInformation.txt");
        string info = sr.ReadToEnd();
        lblHelp.Text = info;
        sr.Close();
    }
}
```

8. Графический пользовательский интерфейс программного обеспечения и его описание.

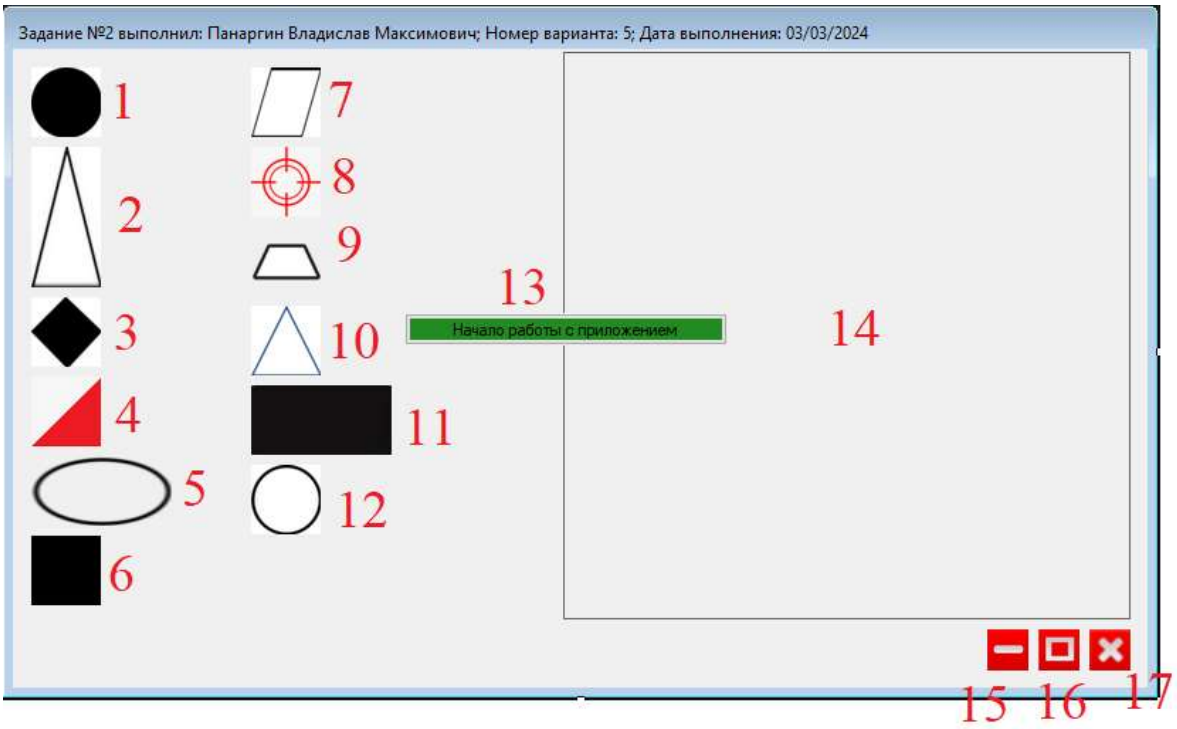

Рисунок 5 – Интерфейс основной формы

1. Кнопка отрисовки круга
2. Кнопка отрисовки равнобедренного треугольника
3. Кнопка отрисовки ромба
4. Кнопка отрисовки прямоугольного треугольника
5. Кнопка отрисовки эллипса
6. Кнопка отрисовки квадрата
7. Кнопка отрисовки параллелограмма
8. Кнопка отрисовки прицела
9. Кнопка отрисовки трапеции
10. Кнопка отрисовки равностороннего треугольника
11. Кнопка отрисовки прямоугольника
12. Кнопка отрисовки окружности
13. Кнопка «Начало работы с приложением»
14. Зона отрисовки
15. Кнопка «свернуть»

16. Кнопка «максимизировать/минимизировать»
17. Кнопка «закрыть»

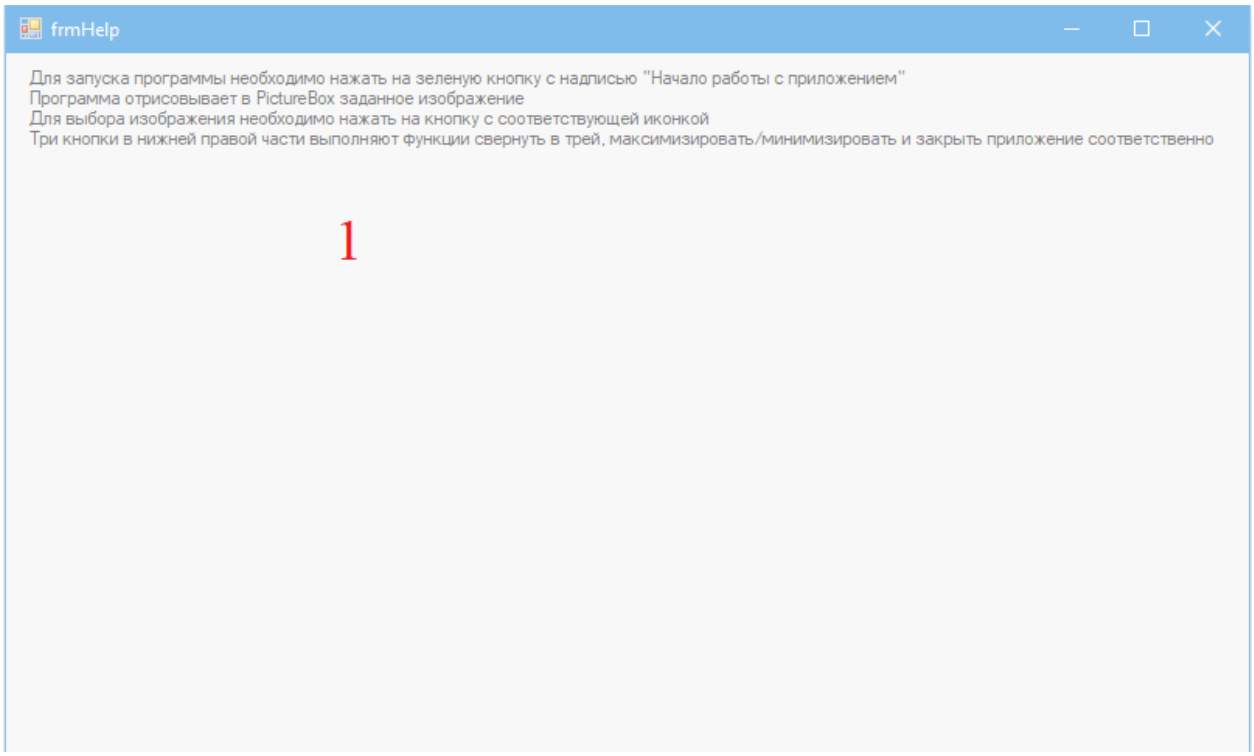

Рисунок 6 – Интерфейс вспомогательной формы

1. Зона для выгрузки текста из файла

9. Подтверждение соответствия графического пользовательского интерфейса требованиям к оформлению.

1. Заголовок экранной формы должен содержать надпись вида: «Задание №2 выполнил: [Фамилия И.О. авторов]; Номер варианта: [Номер]; Дата выполнения: [дд/мм/гггг]».

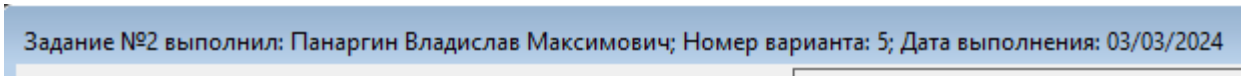

Рисунок 7 – Название формы

2. Дата выполнения проставляется в момент, когда программа считается законченной и по ней можно готовить итоговый отчёт о выполнении работы.

Рисунок 8 – Время на момент создания законченной программы

3. Нечётные варианты отключают стандартный блок управления экранной формой и создают авторские кнопки «Свернуть», «Развернуть», «Закрыть» внизу экранной формы.

Рисунок 9 – Авторские элементы управления

5. Справочная информация должна быть вызвана в дочерней экранной форме и считана в статический по размеру ярлык (Label) из текстового файла (нечётные варианты)

см. Рисунок 6; листинг дополнительной формы

6. Нечётные варианты обеспечивают полупрозрачность дочерней экранной формы.

см. Рисунок 6

8. Исходное состояние всех элементов, расположенных на главной экранной форме, должно быть настроено через перечень параметров этих элементов.

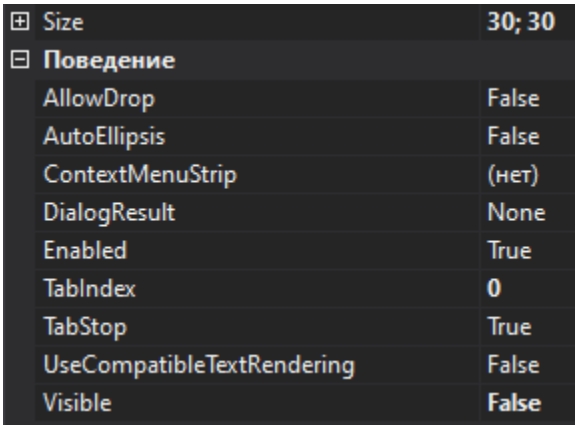

Рисунок 10 – Параметры кнопки "закрыть"

9. В качестве исходного состояния принимается заранее известная и заполненная элементами структура списков, все элементы экранной формы за исключением главного меню находятся либо в недоступном состоянии (.*Enabled = false*), либо в невидимом состоянии (.*Visible = false*).

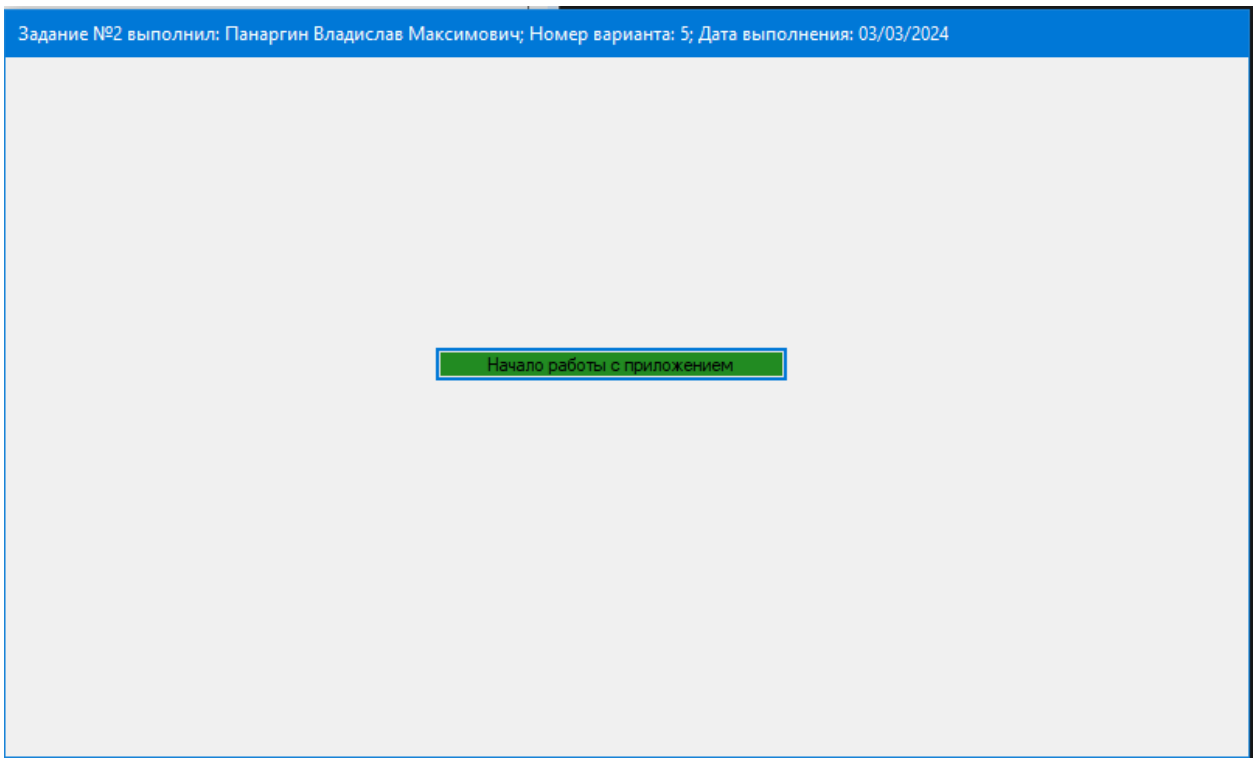

Рисунок 11 – Интерфейс до нажатия кнопки "Начало работы с приложением"

10. Пункт меню «Начало работы с приложением» должен реализовывать активацию доступа пользователя к элементам или отображение элементов на экранной форме для пользователя; список должен быть организован в виде кнопок с иконками; фигуры должны быть расположены в следующем порядке: круг, равнобедренный треугольник, ромб, прямоугольный треугольник, эллипс, квадрат, параллелограмм, прицел, трапеция, равносторонний треугольник, прямоугольник, окружность.

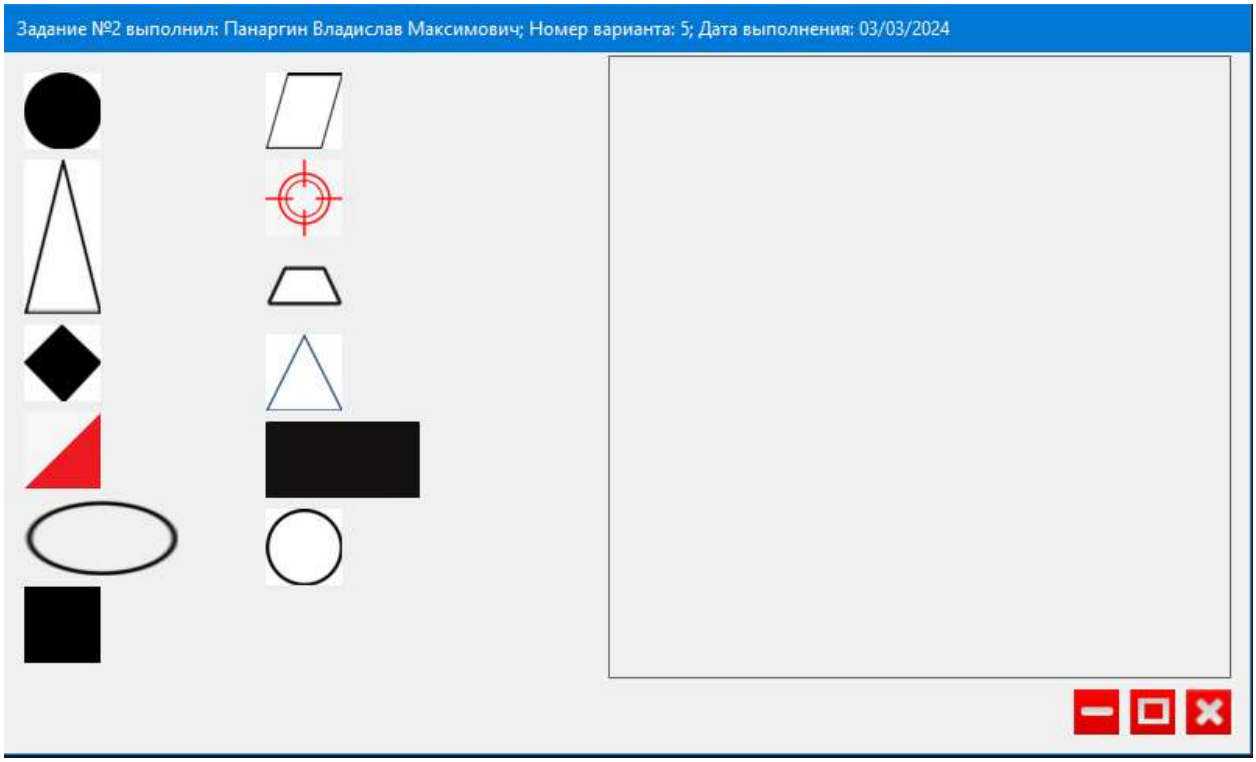

Рисунок 12 – Интерфейс после нажатия кнопки "Начало работы с приложением"

11. Все элементы программы должны носить значащие имена переменных, в которых отражено существо этих элементов, например, главная экранная форма – *frmMain*, ярлык – *lblHelp*, комбинированный список – *cmbFigures* и т.д.

см. Рисунки 1-2

10. Расчёт тестовых примеров с использованием составленного программного обеспечения.

Тестовые примеры:

- Запустить программу (проверка открытия основной и дочерней формы)

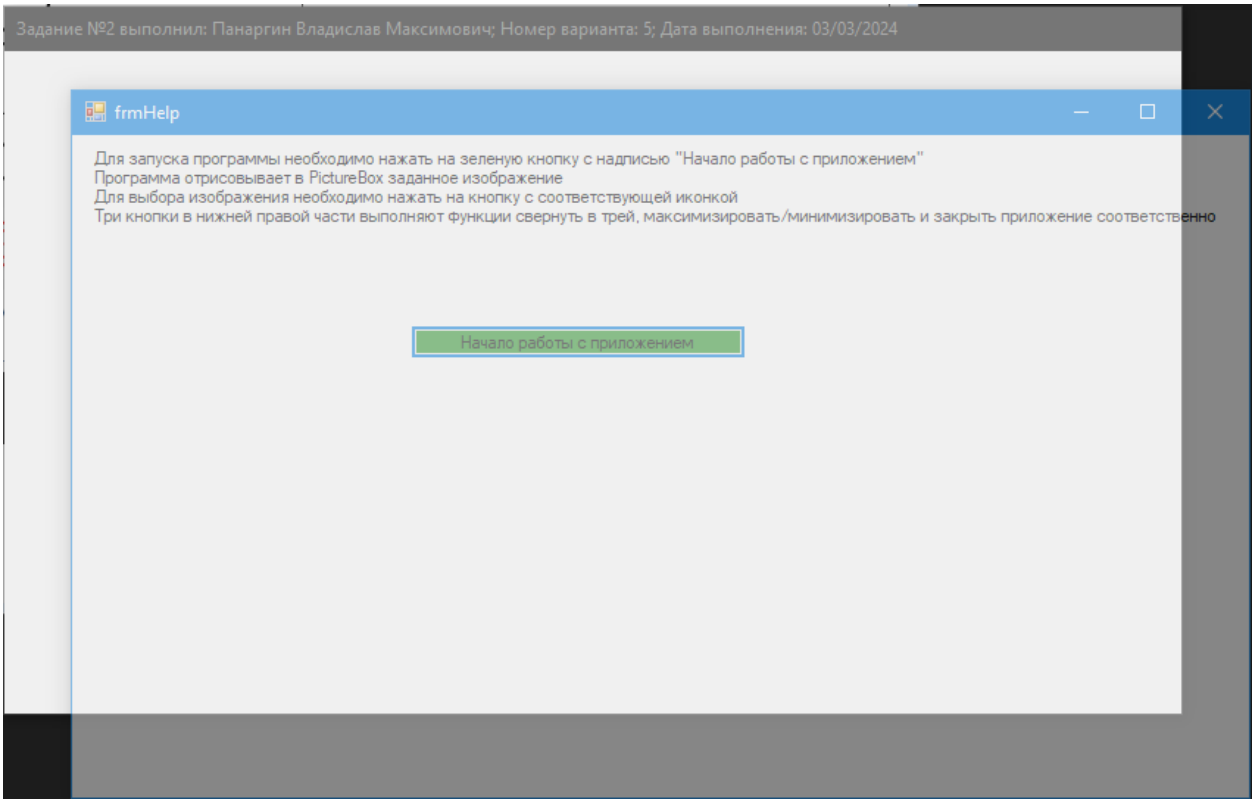

Рисунок 13 – Открытие двух форм

- Нажать кнопку «Начало работы с приложением»

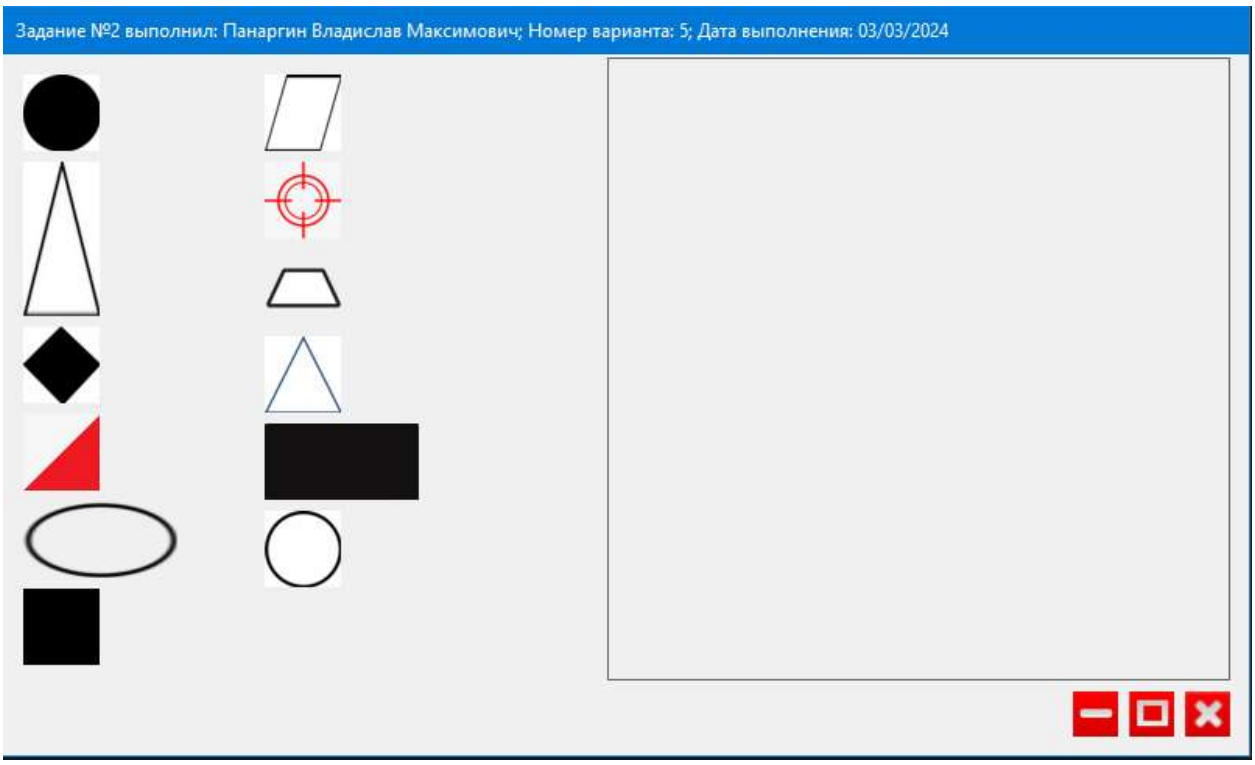

Рисунок 14 – Форма после начала работы

- Проверить корректность отрисовки каждой из фигур

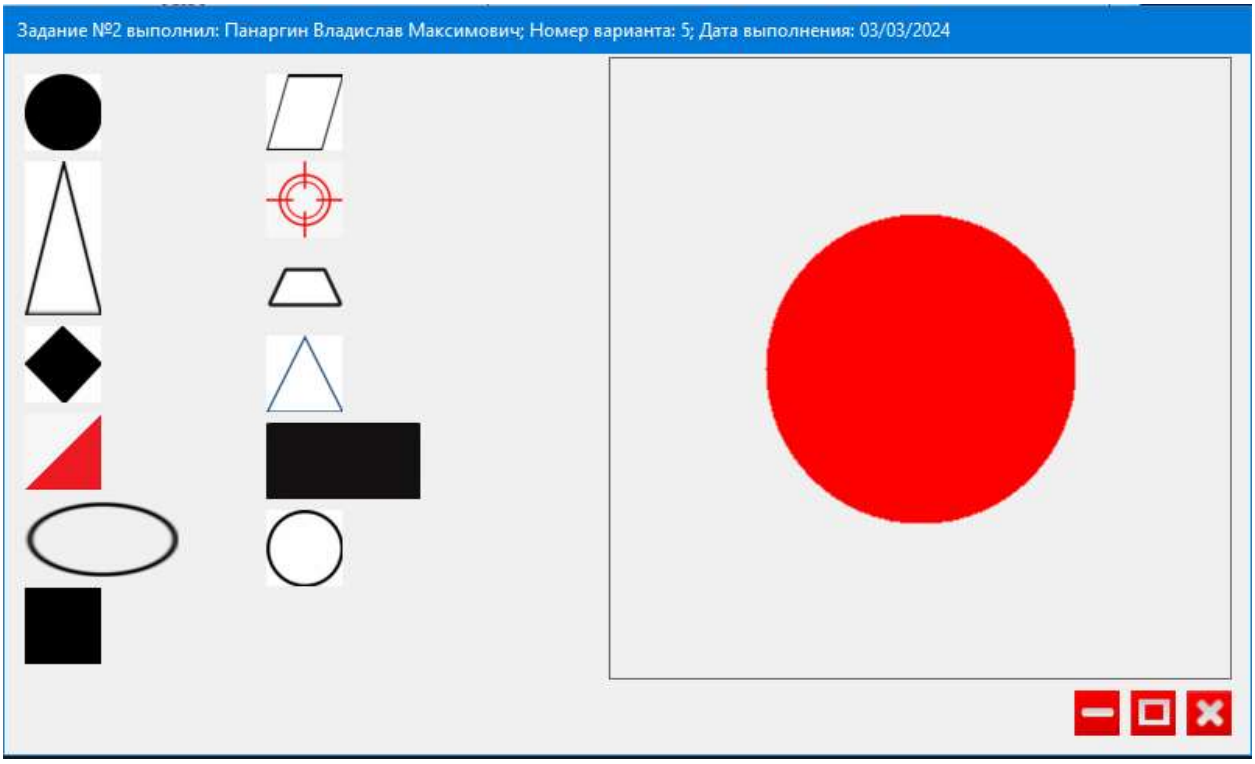

Рисунок 15 – Отрисовка окружности

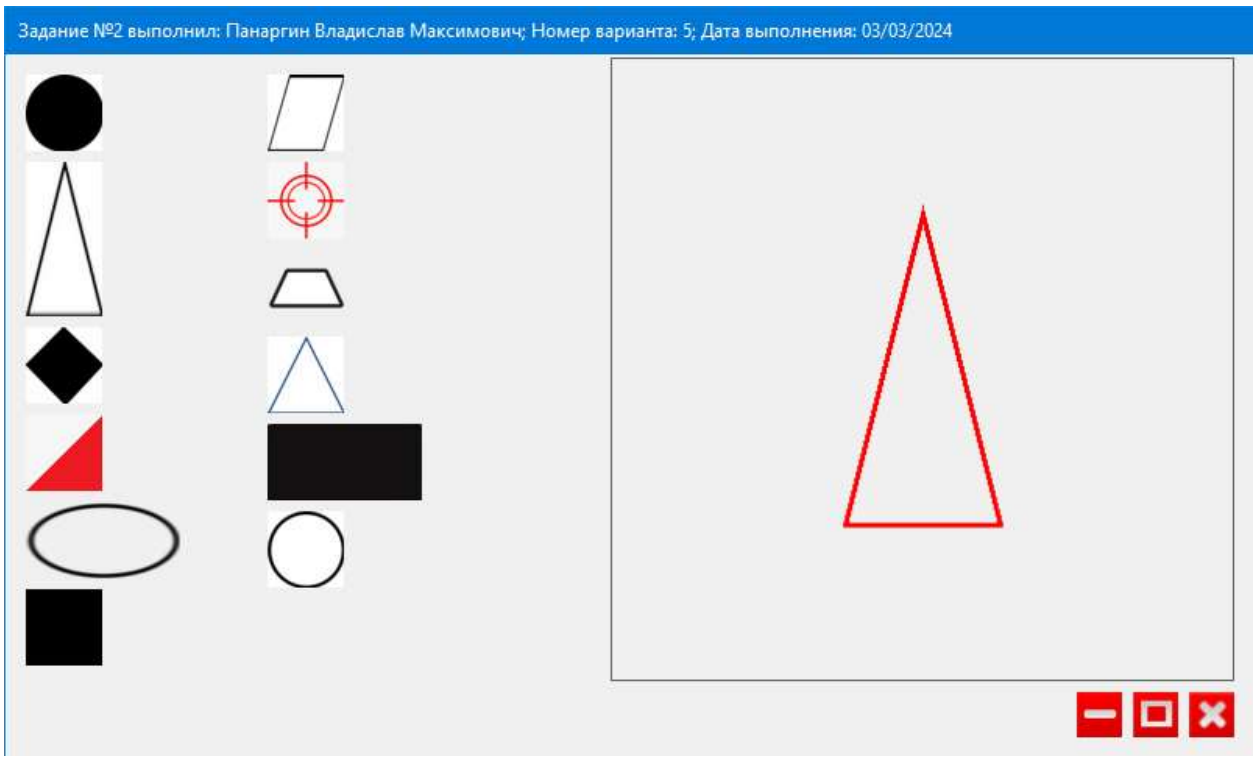

Рисунок 16 – Отрисовка равнобедренного треугольника

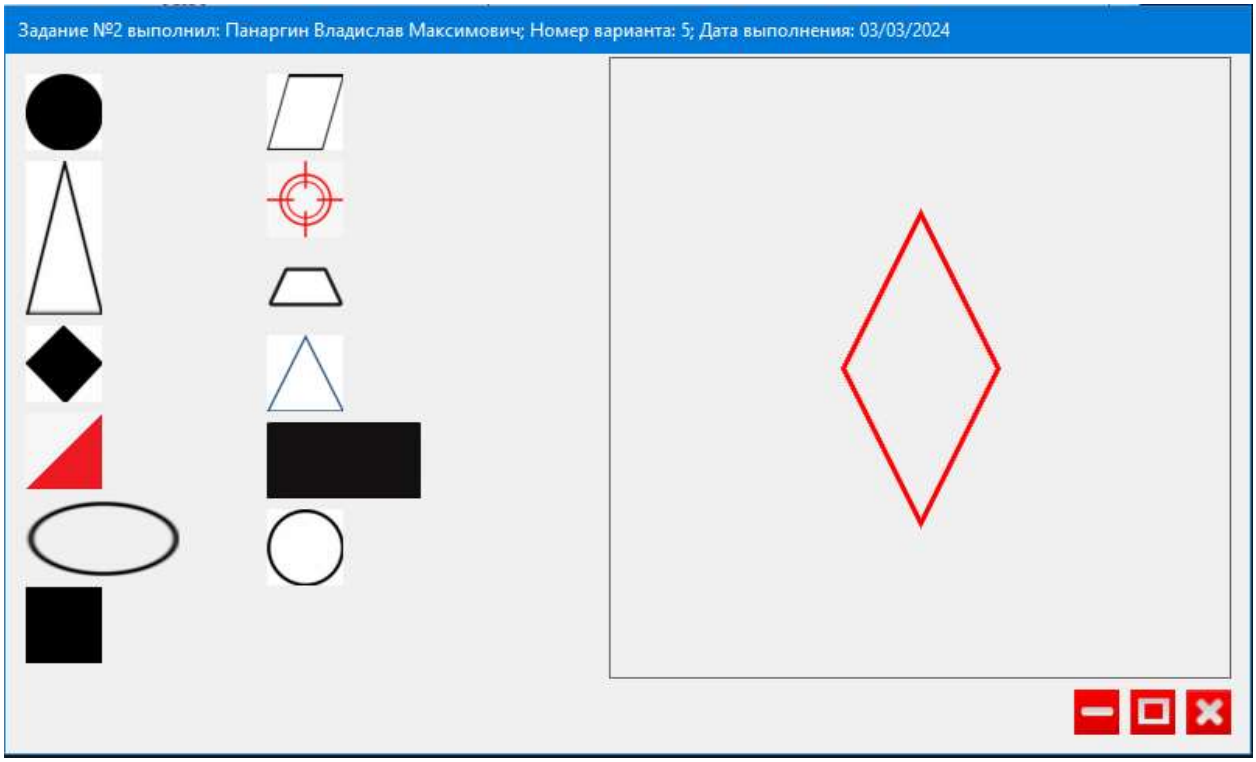

Рисунок 17 – Отрисовка ромба

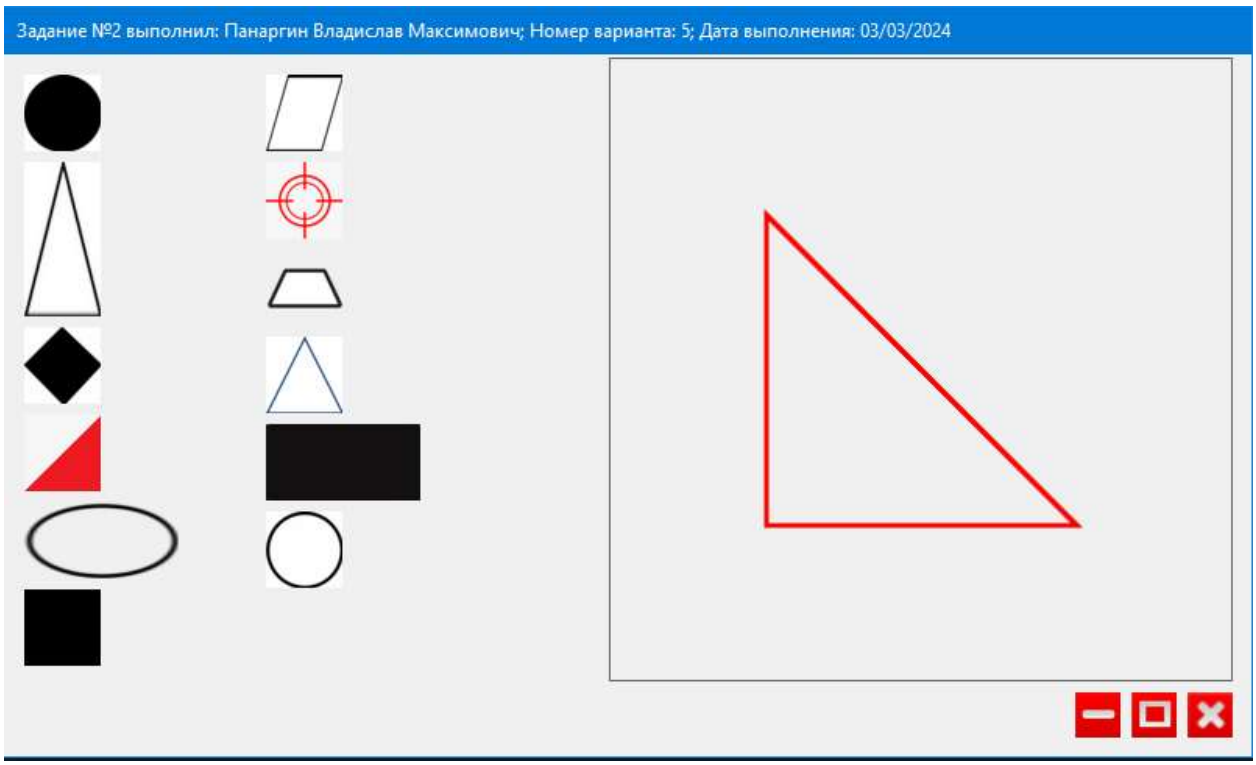

Рисунок 18 – Отрисовка прямоугольного треугольника

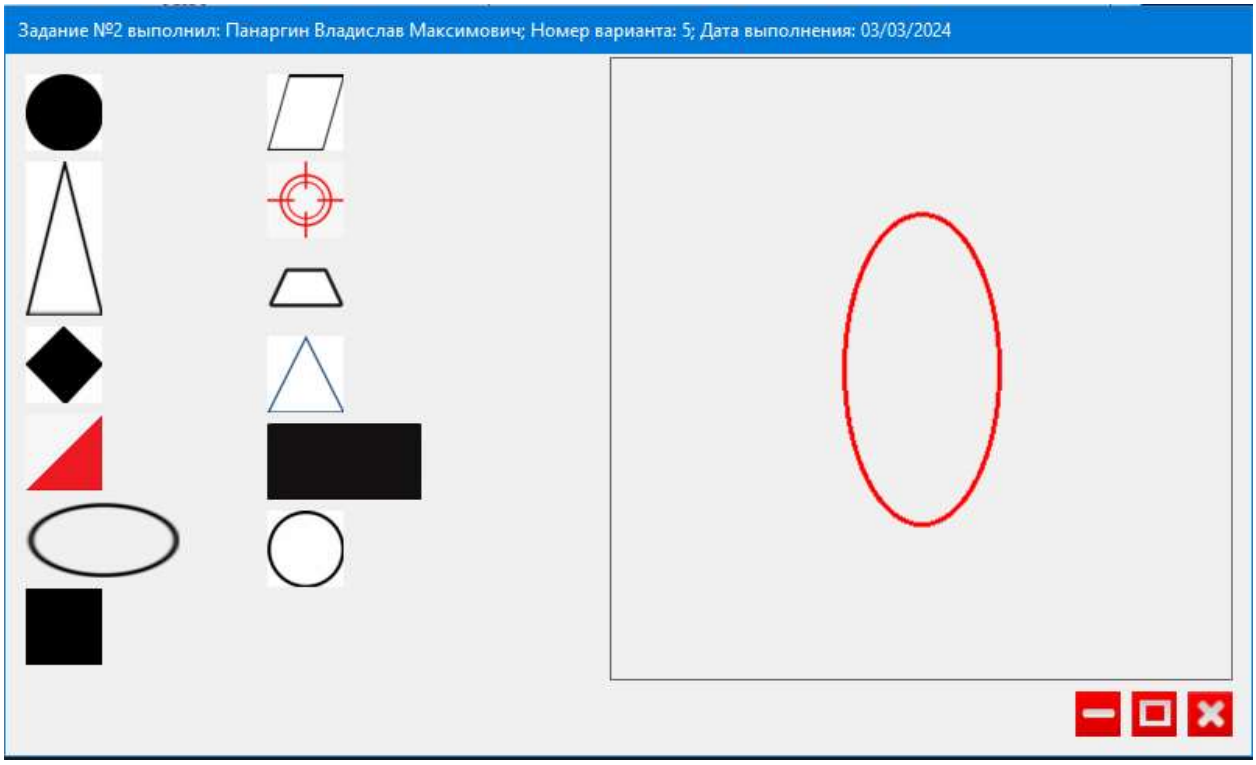

Рисунок 19 – Отрисовка эллипса

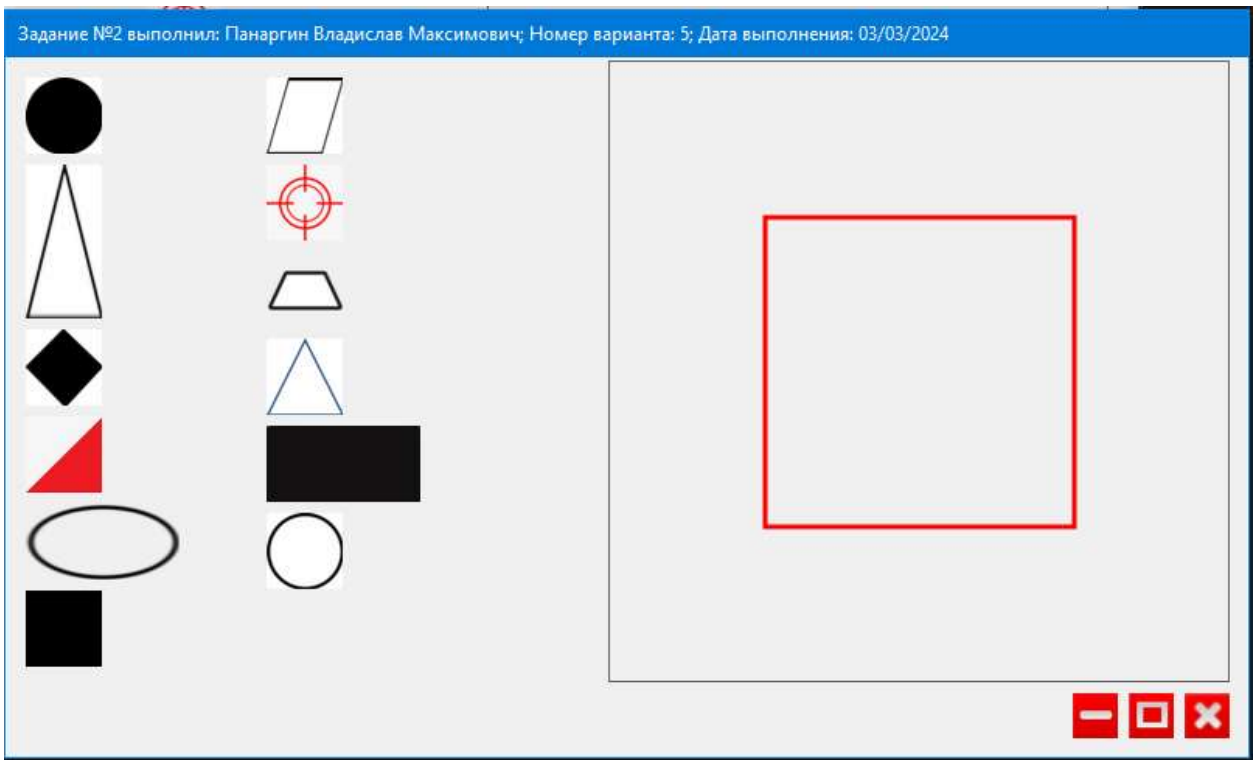

Рисунок 20 – Отрисовка квадрата

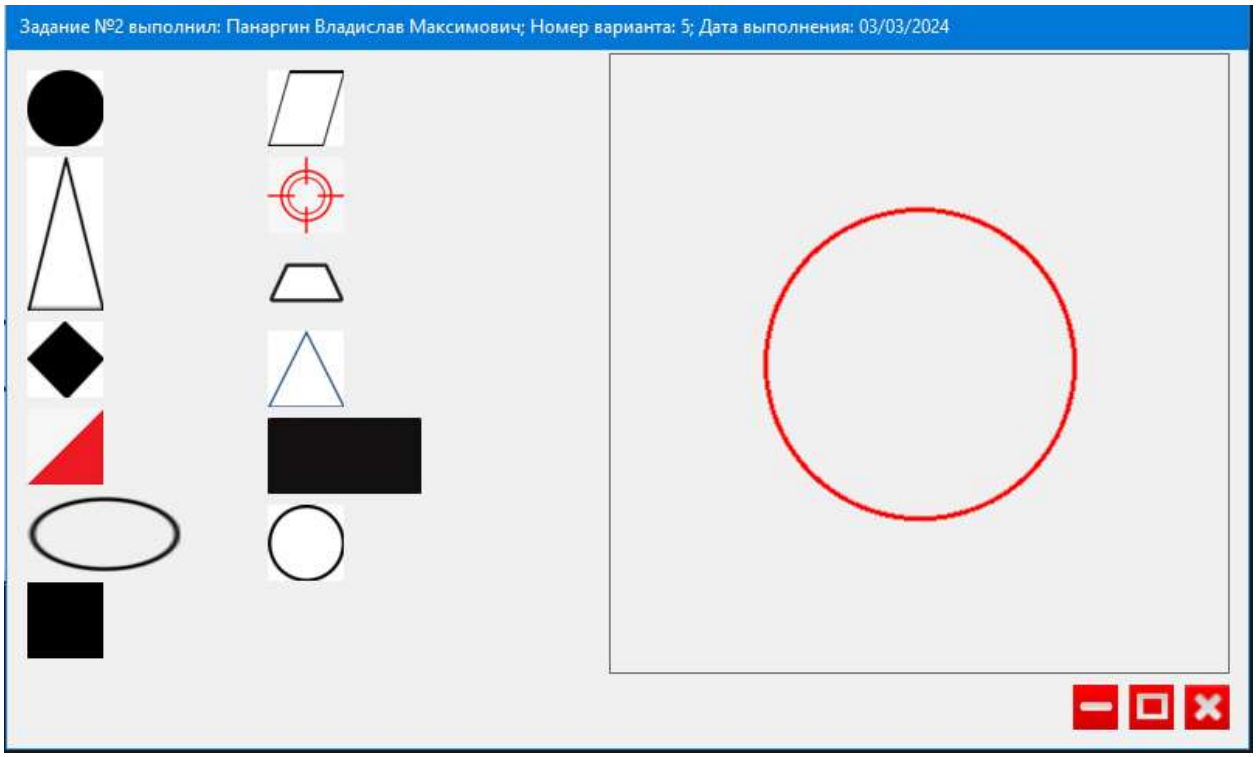

Рисунок 21 – Отрисовка окружности

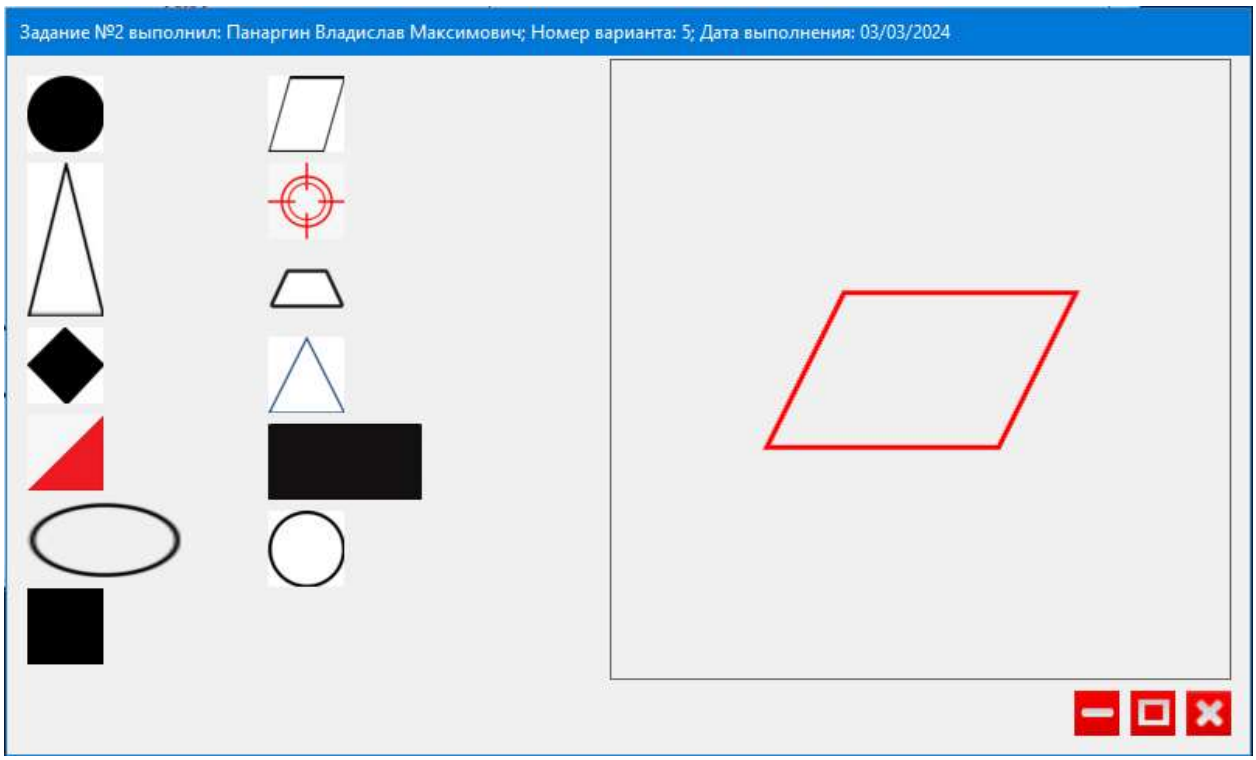

Рисунок 22 – Отрисовка параллелограмма

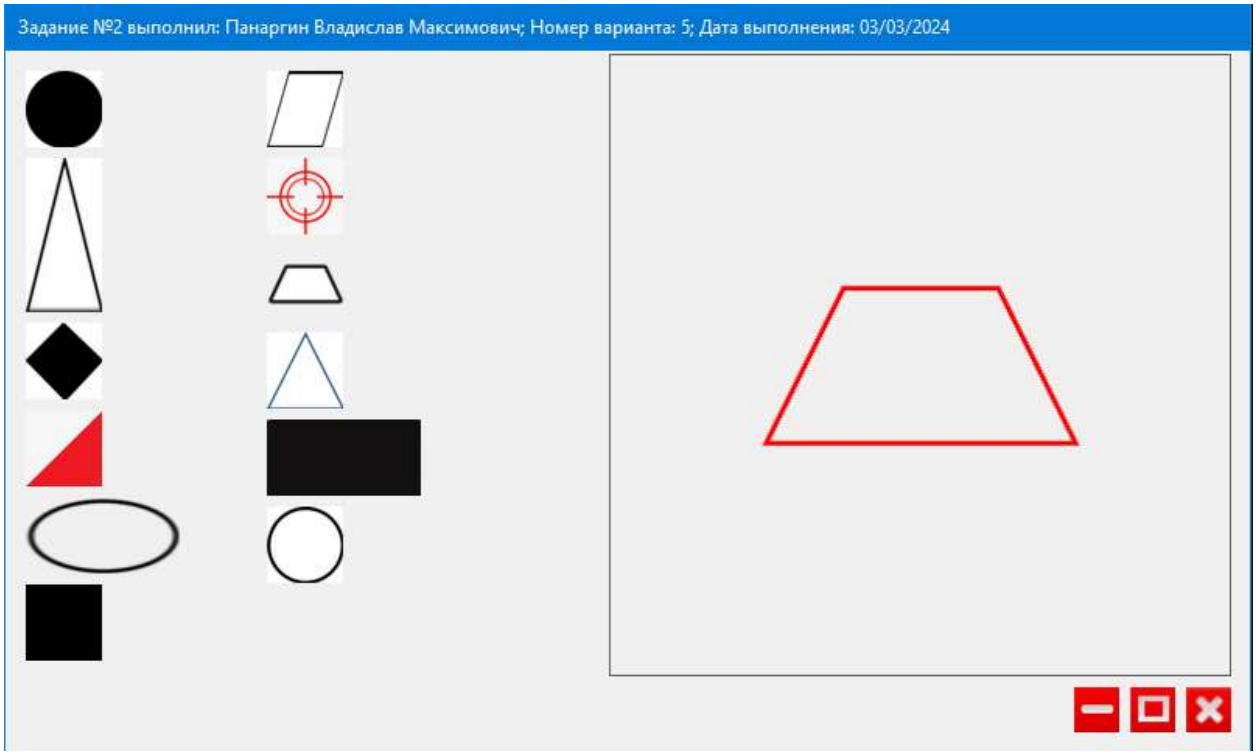

Рисунок 23 – Отрисовка трапеции

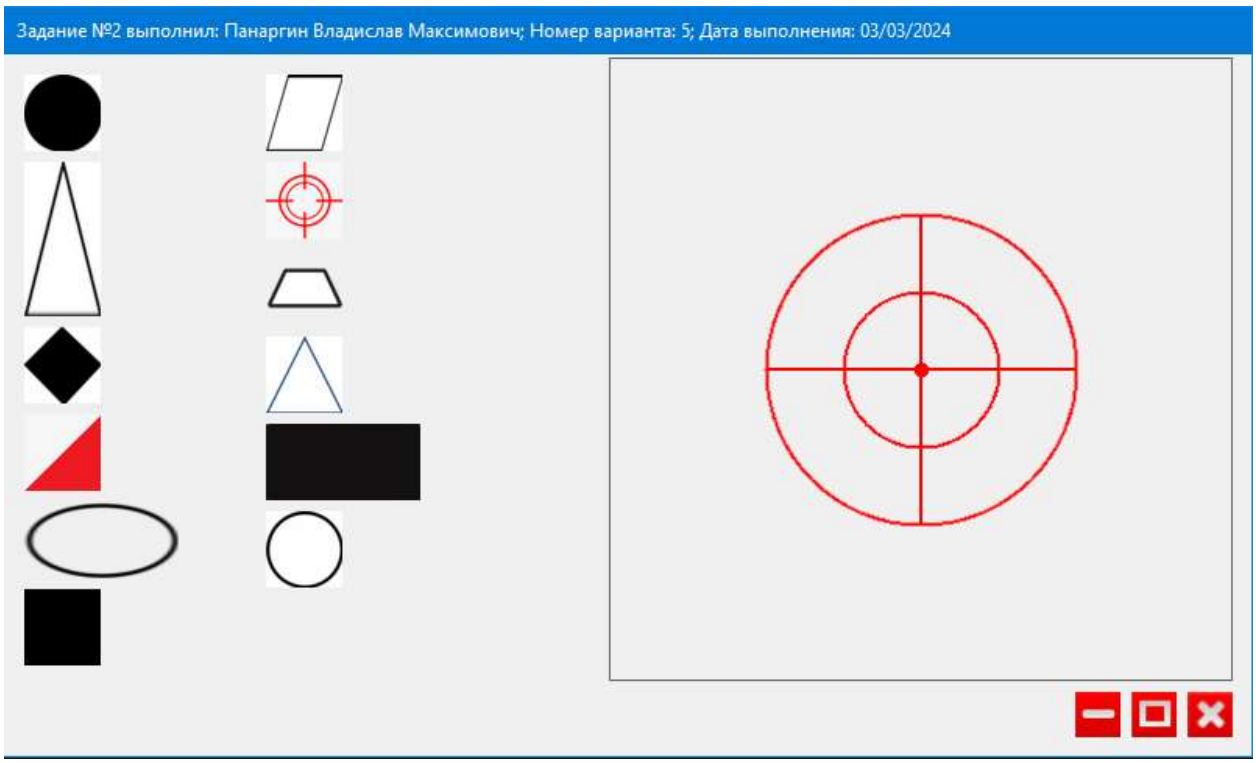

Рисунок 24 – Отрисовка прицела

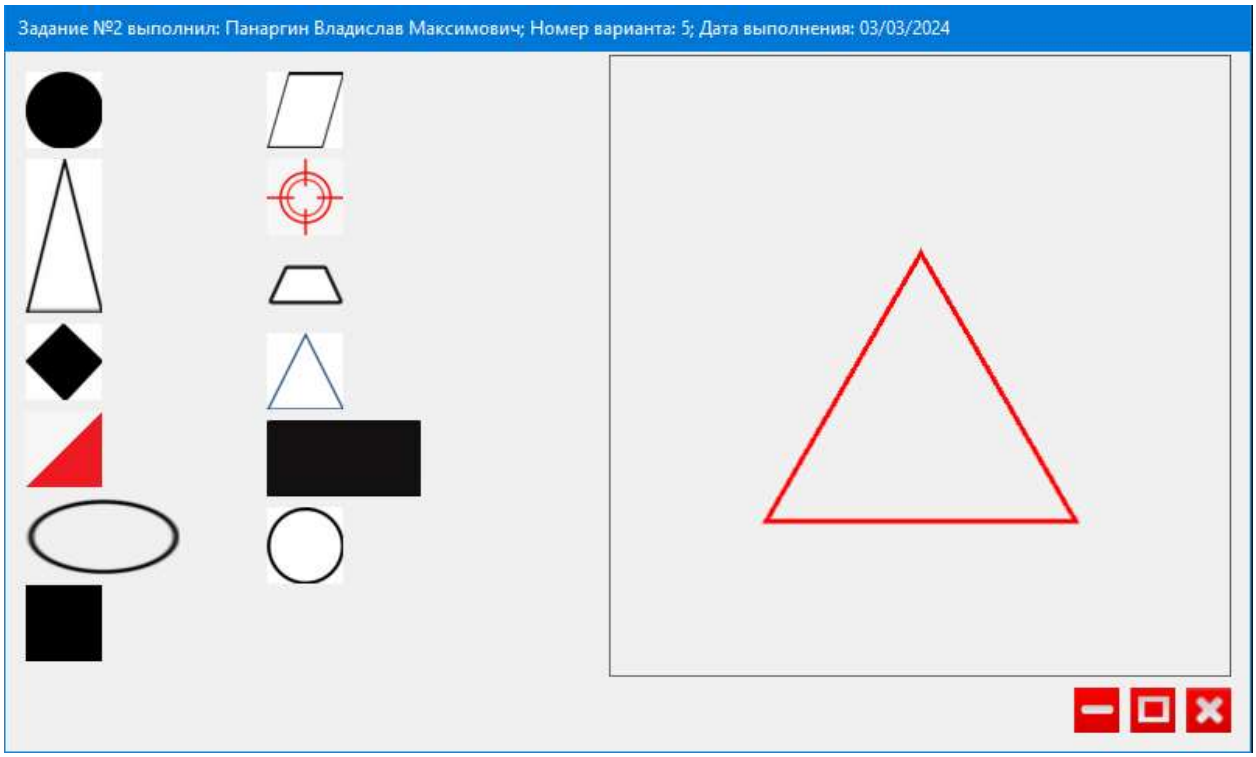

Рисунок 25 – Отрисовка равностороннего треугольника

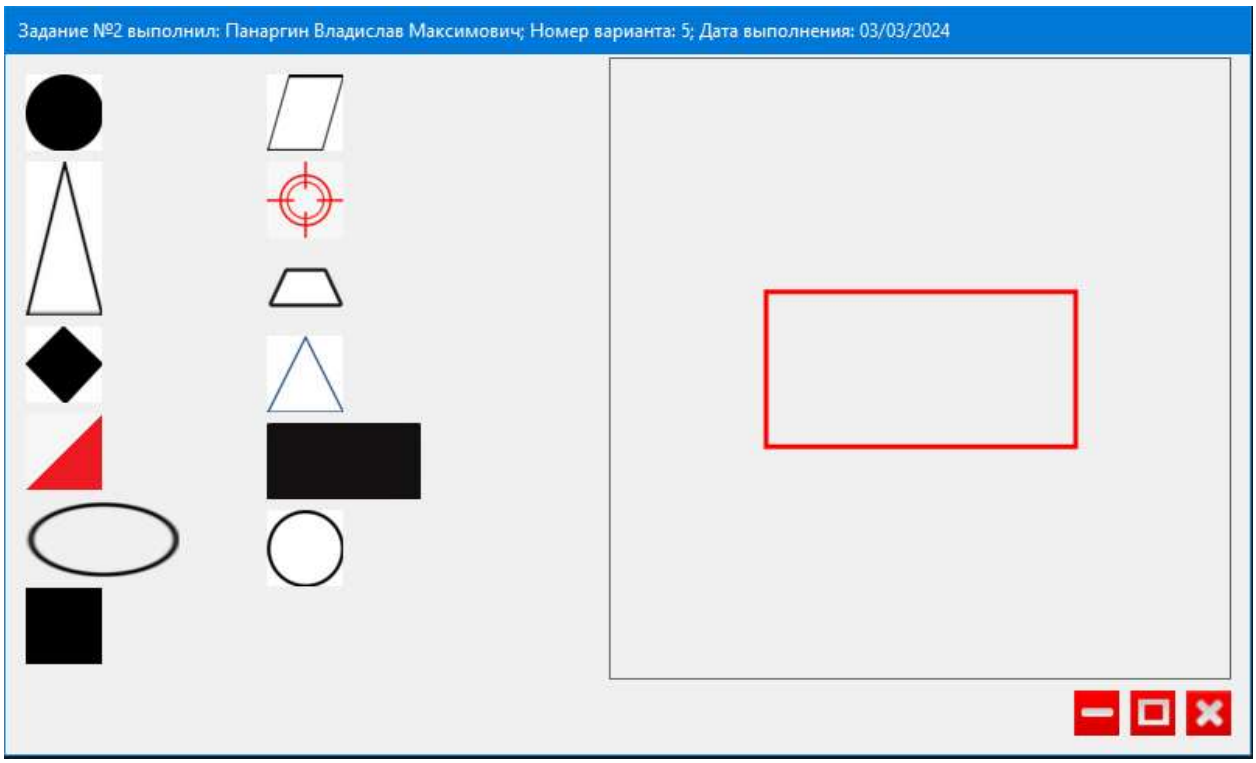

Рисунок 26 – Отрисовка прямоугольника

- Проверить кнопку «максимизировать/минимизировать»

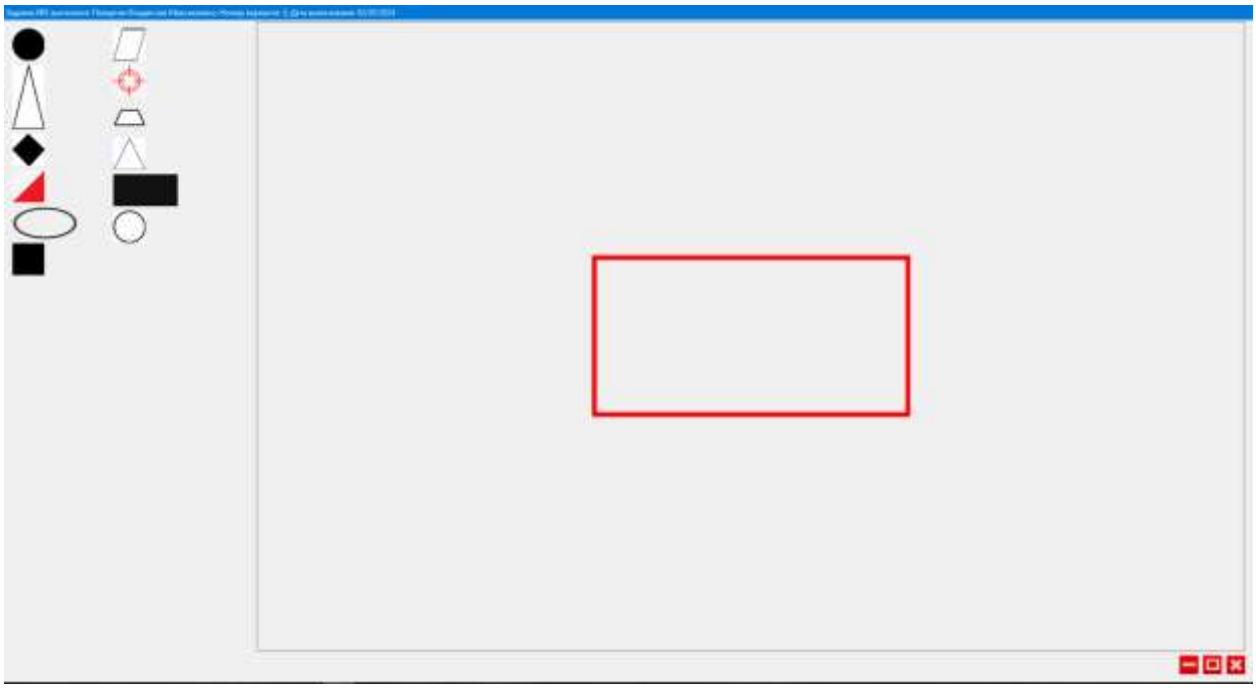

Рисунок 27 – Проверка кнопки "максимизировать/минимизировать"

11. Формулировку вывода о проделанной работе

В рамках данной работы были закреплены навыки разработки визуального пользовательского интерфейса, освоена работа с текстовыми файлами и кодировкой в среде *Microsoft Visual Studio*, были получены знания о реализации настройки множественных состояний объектов посредством управления компонентами со внутренней индексацией.