



**МИНИСТЕРСТВО ТРАНСПОРТА РОССИЙСКОЙ ФЕДЕРАЦИИ**  
**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ**  
**УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**  
**«РОССИЙСКИЙ УНИВЕРСИТЕТ ТРАНСПОРТА»**  
**(РУТ (МИИТ))**

**ИНСТИТУТ ТРАНСПОРТНОЙ ТЕХНИКИ И СИСТЕМ УПРАВЛЕНИЯ**

**Лабораторная работа №2**  
**по дисциплине «Методы программирования»**  
**«Конструктор ГПИ. Фигуры»**

**Выполнил:** ст. гр. ТКИ-342

Никулин Д.В.

Ситало Р.В.

**Проверил:** доцент, к.т.н.

Сафронов А.И.

**Москва – 2024 г**

## Оглавление

<b>Цель работы.....</b>	<b>3</b>
<b>Формулировка задачи .....</b>	<b>3</b>
Индивидуальная задача .....	4
<b>Диаграммы классов, входящих в состав решения .....</b>	<b>5</b>
<b>Сеть Петри .....</b>	<b>6</b>
Полная сеть Петри .....	6
Краткая сеть Петри.....	7
<b>Скриншоты работы программы.....</b>	<b>9</b>
<b>Код программы.....</b>	<b>12</b>
Начальная форма .....	12
Форма Help.....	13
Форма Paint .....	14
<b>Вывод .....</b>	<b>18</b>

## 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».

## 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».

#### Индивидуальная задача

Полупрозрачность дочерней экранной формы; выбор фигуры реализован с помощью комбинированного списка; дополнительная фигура – правильный шестиугольник.

## Диаграммы классов, входящих в состав решения

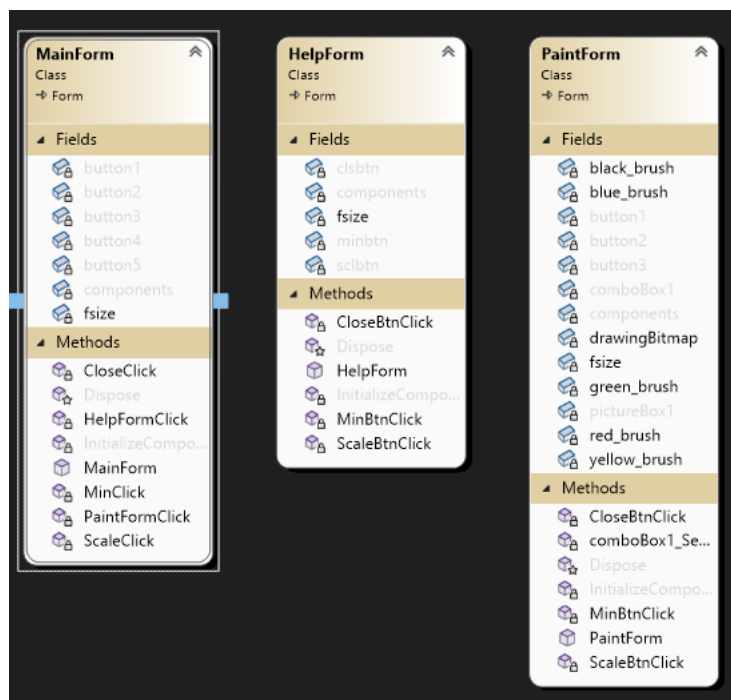


Рисунок 1 – Диаграмма классов

# Полная сеть Петри

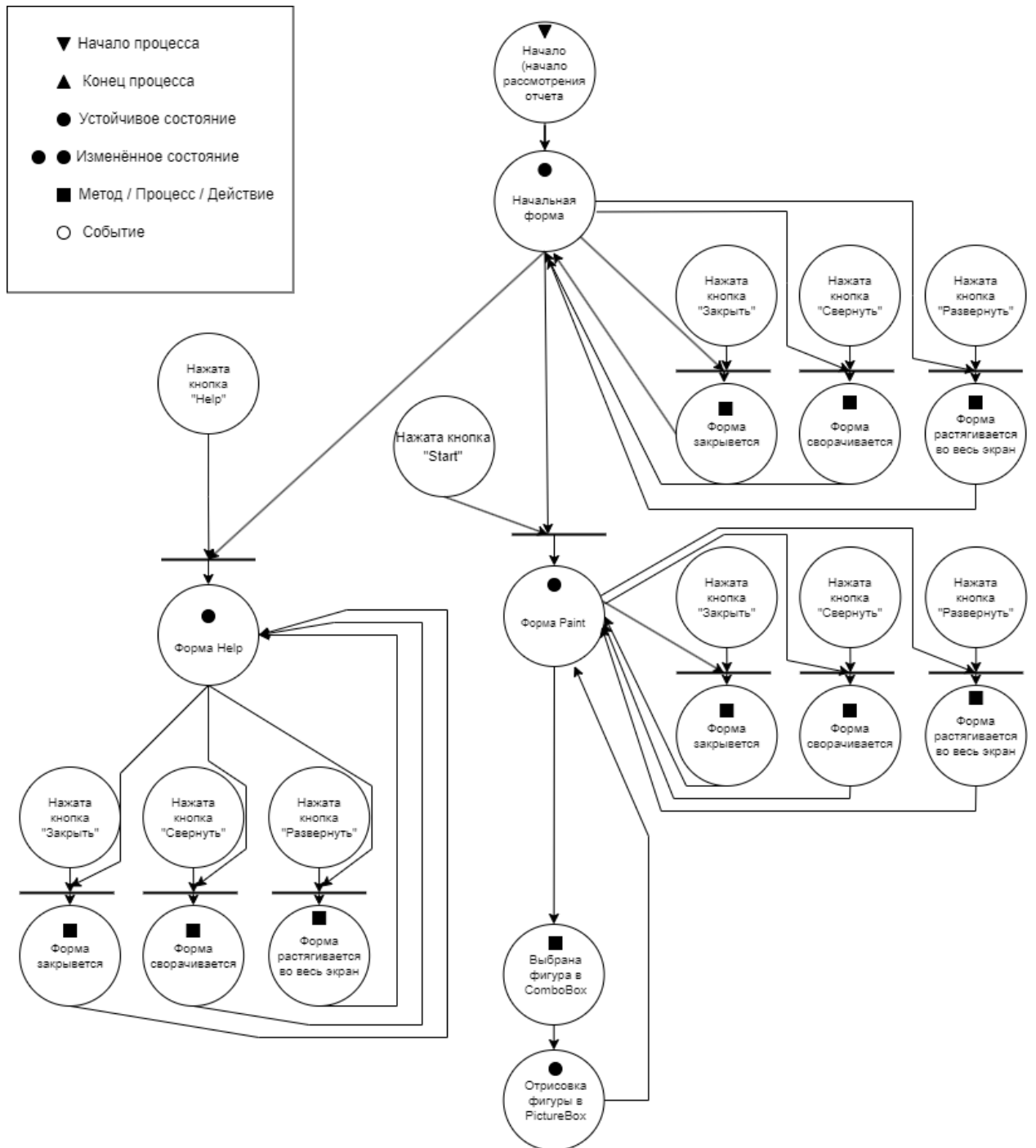


Рисунок 2 – Полная сеть Петри

## Краткая сеть Петри

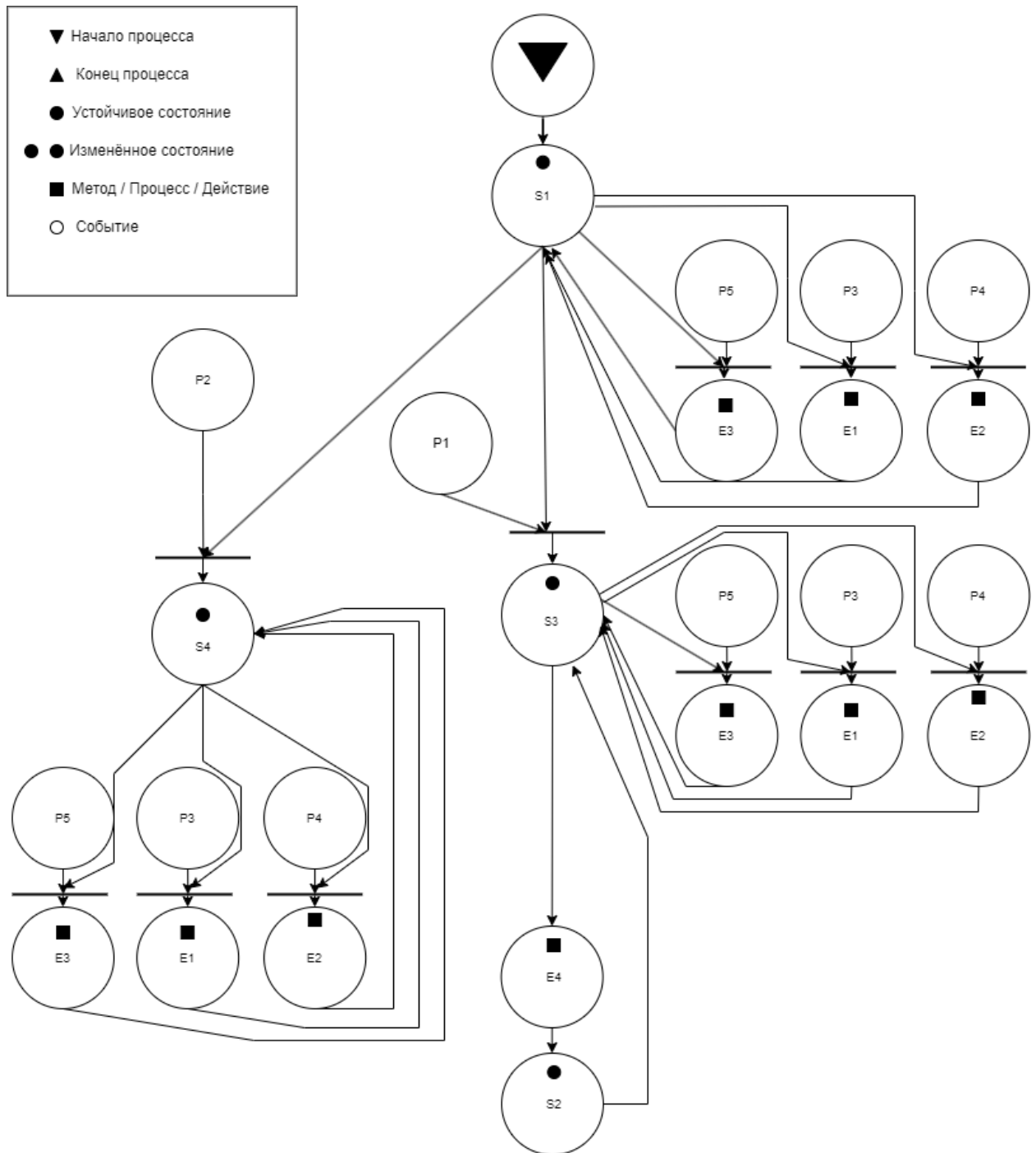


Рисунок 3 – Краткая сеть Петри

### Описание:

– состояния (states):

s1 – Начальная форма

s2 – Отрисована фигура

s3 – Форма Paint

s4 – форма Help

– действия (effects):

- e1 – Пользователь нажимает кнопку “Свернуть”
- e2 – Пользователь нажимает кнопку “Развернуть”
- e3 – Пользователь нажимает кнопку “Заккрыть”
- e4 – Пользователь выбирает фигуру в ComboBox

– события (prompts):

- p1 – кнопка “Start”
- p2 – кнопка “Help”
- p3 – Кнопка “Свернуть”
- p4 – Кнопка “Развернуть”
- p5 – Кнопка “Заккрыть”



## Скриншоты работы программы



Рисунок 4 – Начальная форма

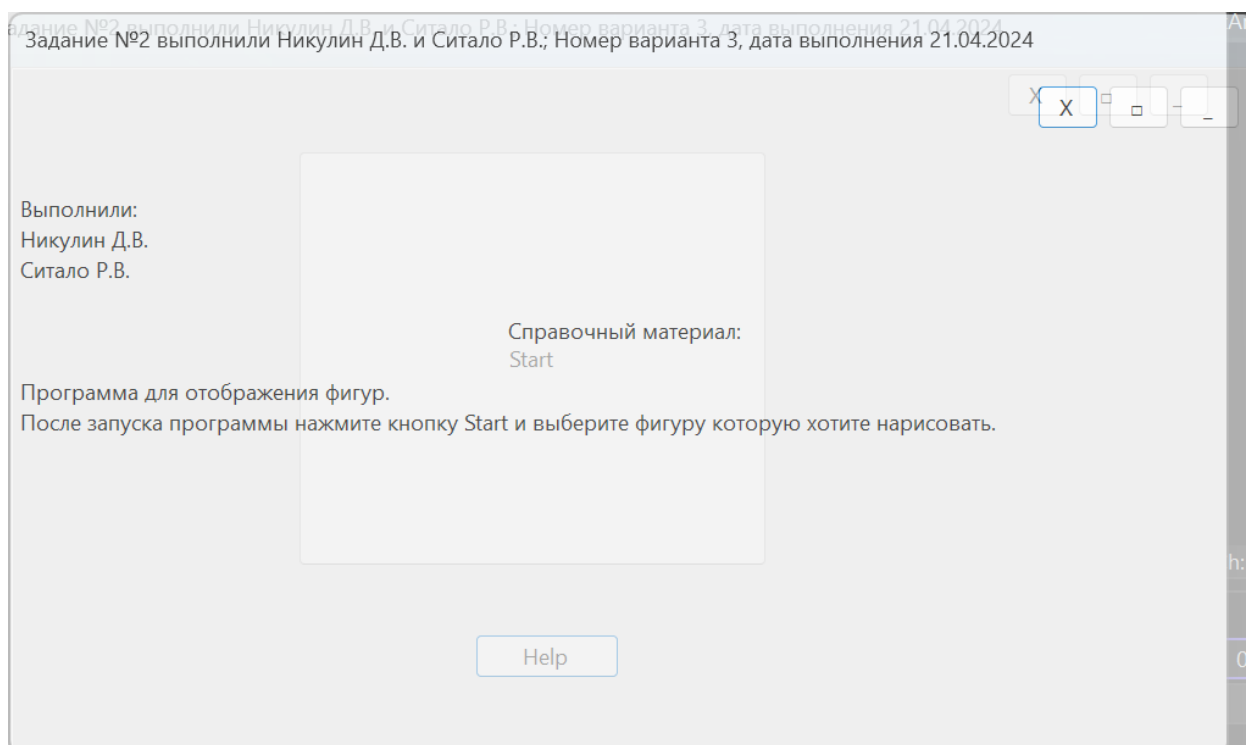


Рисунок 5 – Форма Help

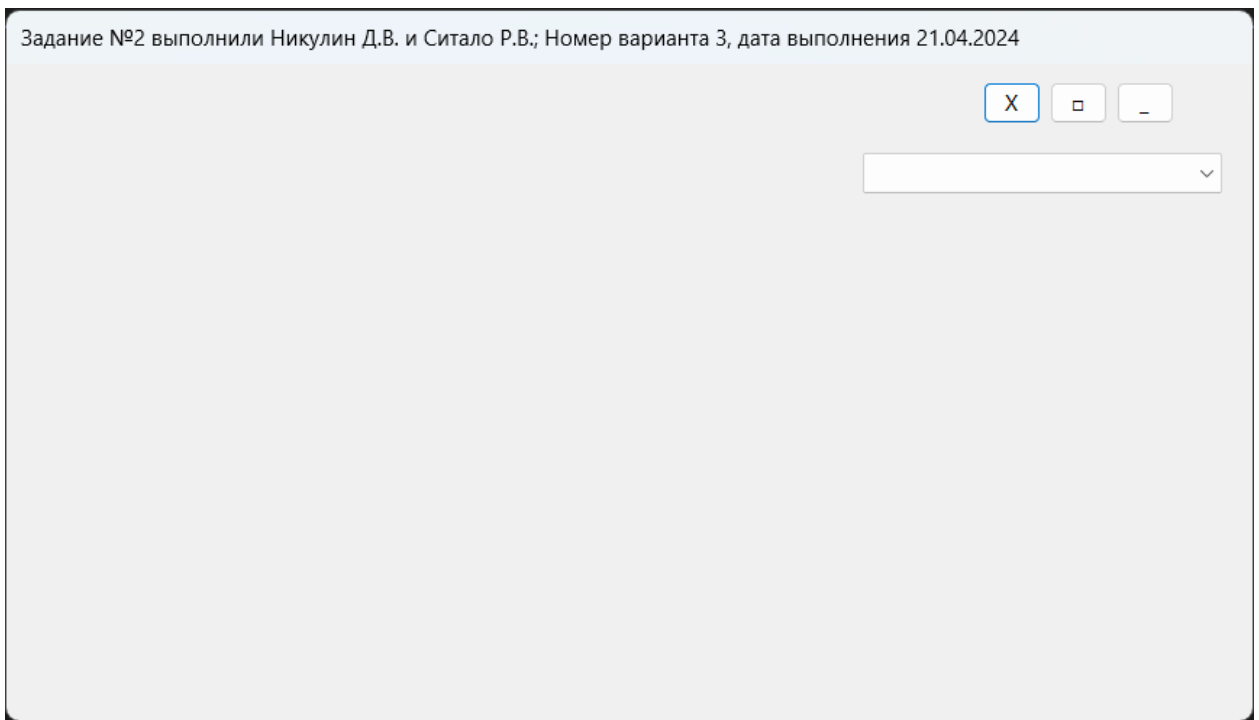


Рисунок 6 – Форма Paint

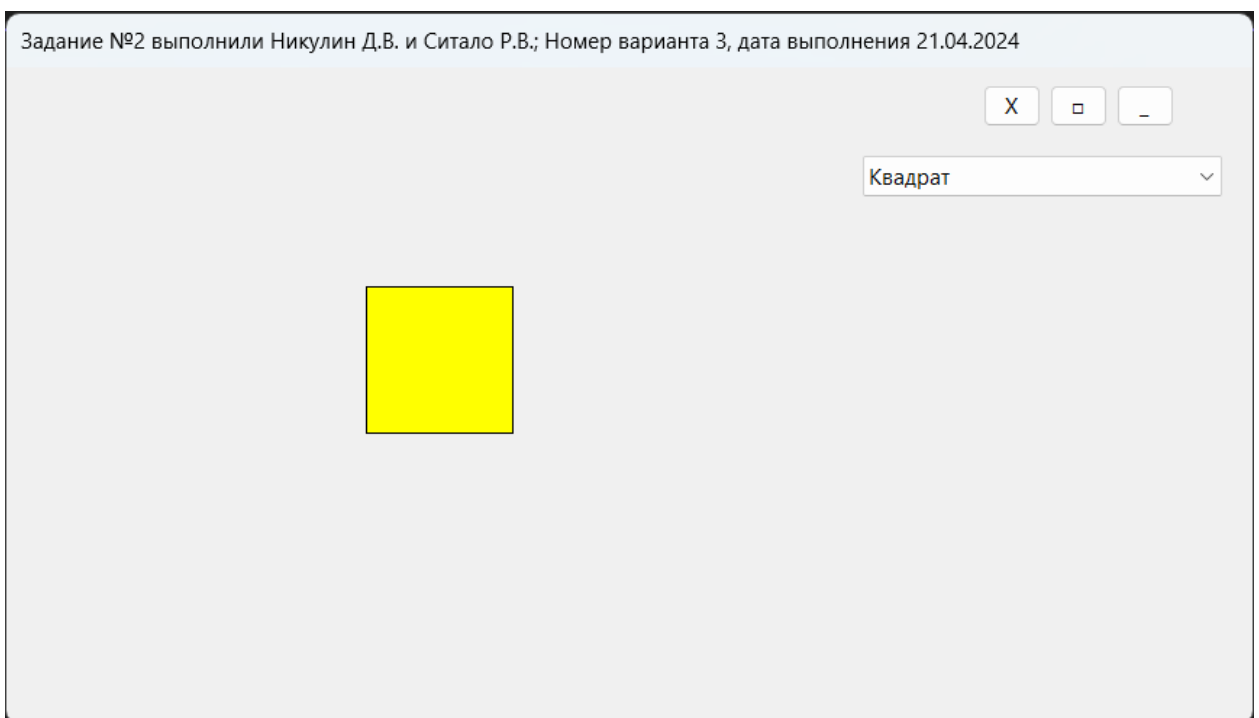


Рисунок 7 – Рисовка квадрата

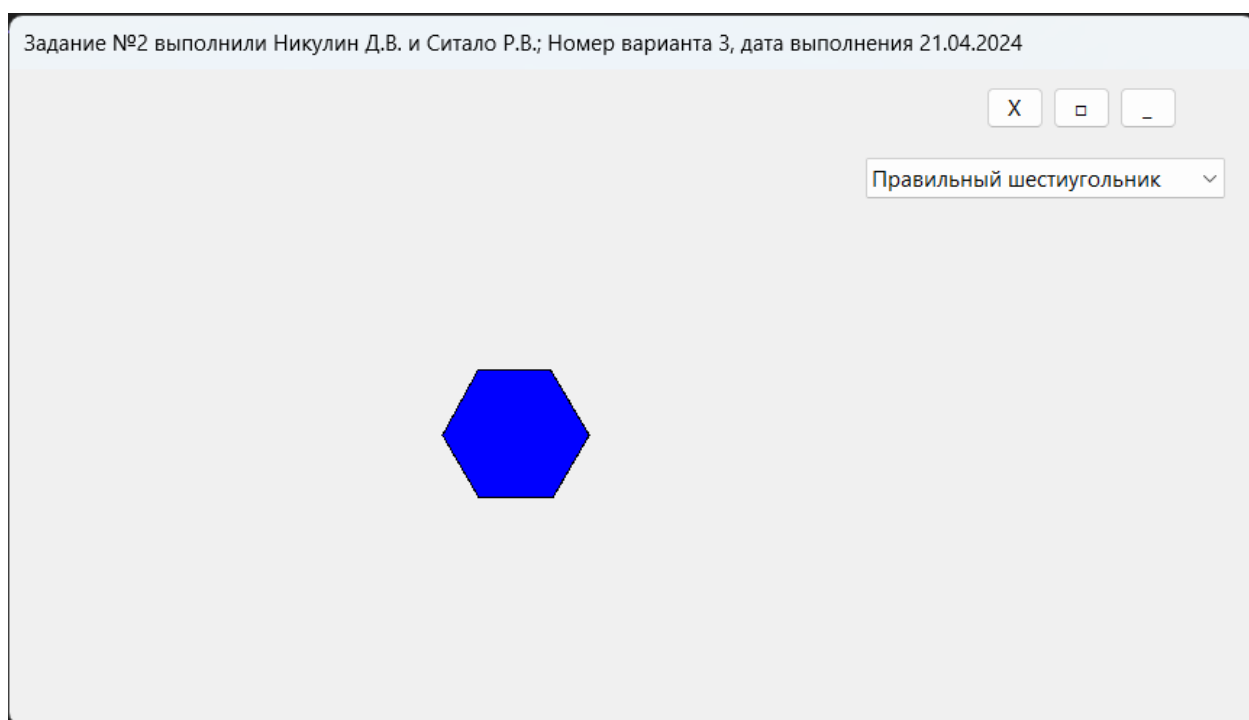


Рисунок 8 – Рисовка правильного шестиугольника

## Код программы

### Начальная форма

```
using System.Windows.Forms;

namespace WinFormsApp1
{
    public partial class MainForm : Form
    {
        bool fsize = true;
        public MainForm()
        {
            InitializeComponent();
            this.Text = "Задание №2 выполнили Никулин Д.В. и Ситало Р.В.; Номер
варианта 3, дата выполнения 21.04.2024";
            this.ControlBox = false;
        }

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

        private void ScaleClick(object sender, EventArgs e)
        {
            if (fsize)
            {
                this.TopMost = true;
                this.WindowState = FormWindowState.Maximized;
                fsize = false;
            }
            else
            {
                this.TopMost = true;
                this.WindowState = FormWindowState.Normal;
                fsize = true;
            }
        }

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

        private void PaintFormClick(object sender, EventArgs e)
        {
            PaintForm form2 = new PaintForm();
            this.Hide();
            form2.Show();
        }

        private void HelpFormClick(object sender, EventArgs e)
        {
            HelpForm helpfrm = new HelpForm();

            TextBox helpTextBox = new TextBox();
            helpTextBox.Multiline = true;
            helpTextBox.Dock = DockStyle.Fill;
            helpTextBox.ReadOnly = true;

            // Читаем содержимое файла справки
        }
    }
}
```

```

        string helpFilePath = "help.txt";
        if (File.Exists(helpFilePath))
        {
            string helpText = File.ReadAllText(helpFilePath);
            helpTextBox.Text = helpText;
        }
        else
        {
            helpTextBox.Text = "File not found: help.txt";
        }

        helpfrm.Controls.Add(helpTextBox);
        helpfrm.Show();
    }
}

```

## Форма Help

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace WinFormsApp1
{
    public partial class HelpForm : Form
    {
        bool fsize = true;

        public HelpForm()
        {
            InitializeComponent();
            this.Opacity = 0.7;
            this.Text = "Задание №2 выполнили Никулин Д.В. и Ситало Р.В.; Номер
варианта 3, дата выполнения 21.04.2024";
            this.ControlBox = false;
        }

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

        private void ScaleBtnClick(object sender, EventArgs e)
        {
            if (fsize)
            {
                this.TopMost = true;
                this.WindowState = FormWindowState.Maximized;
                fsize = false;
            }
            else
            {
                this.TopMost = true;
                this.WindowState = FormWindowState.Normal;
                fsize = true;
            }
        }
    }
}

```

```

        }
    }

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

```

## Форма Paint

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace WinFormsApp1
{
    public partial class PaintForm : Form
    {
        bool fsize = true;
        private Bitmap drawingBitmap;
        Brush yellow_brush = new SolidBrush(Color.Yellow);
        Brush red_brush = new SolidBrush(Color.Red);
        Brush black_brush = new SolidBrush(Color.Black);
        Brush green_brush = new SolidBrush(Color.Green);
        Brush blue_brush = new SolidBrush(Color.Blue);

        public PaintForm()
        {
            InitializeComponent();
            this.Text = "Задание №2 выполнили Никулин Д.В. и Ситало Р.В.; Номер
варианта 3, дата выполнения 21.04.2024";
            drawingBitmap = new Bitmap(pictureBox1.Width, pictureBox1.Height);
            this.ControlBox = false;
        }

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

        private void ScaleBtnClick(object sender, EventArgs e)
        {
            if (fsize)
            {
                this.TopMost = true;
                this.WindowState = FormWindowState.Maximized;
                fsize = false;
            }
            else
            {
                this.TopMost = true;
                this.WindowState = FormWindowState.Normal;
                fsize = true;
            }
        }
    }
}

```

```

    }

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

    private void comboBox1_SelectedIndexChanged(object sender, EventArgs e)
    {
        drawingBitmap = new Bitmap(pictureBox1.Width, pictureBox1.Height);

        string selectedFigure = comboBox1.SelectedItem.ToString();

        using (Graphics g = Graphics.FromImage(drawingBitmap))
        {
            Pen pen = new Pen(Color.Black, 2);

            switch (selectedFigure)
            {
                case "Эллипс":
                    g.DrawEllipse(pen, 50, 50, 100, 70);
                    g.FillEllipse(green_brush, 50, 50, 100, 70);
                    break;
                case "Прямоугольный треугольник":
                    Point[] rt_points = { new Point(50, 50), new Point(150, 50),
new Point(50, 150), new Point(50, 50) };
                    g.DrawLines(pen, rt_points);
                    g.FillPolygon(blue_brush, rt_points);
                    break;
                case "Правильный шестиугольник":

                    int hexagon_centerX = 150;
                    int hexagon_centerY = 150;
                    int hexagon_radius = 50;
                    int hexagon_sides = 6;

                    Point[] hexagonPoints = new Point[hexagon_sides];
                    for (int i = 0; i < hexagon_sides; i++)
                    {
                        double angle = 2 * Math.PI / hexagon_sides * i;
                        int x = (int)(hexagon_centerX + hexagon_radius *
Math.Cos(angle));
                        int y = (int)(hexagon_centerY + hexagon_radius *
Math.Sin(angle));
                        hexagonPoints[i] = new Point(x, y);
                    }

                    g.DrawPolygon(pen, hexagonPoints);
                    g.FillPolygon(blue_brush, hexagonPoints);
                    break;
                case "Квадрат":
                    g.DrawRectangle(pen, 50, 50, 100, 100);
                    g.FillRectangle(yellow_brush, 50, 50, 100, 100);
                    break;
                case "Прямоугольник":
                    g.DrawRectangle(pen, 50, 50, 150, 100);
                    g.FillRectangle(black_brush, 50, 50, 150, 100);
                    break;
                case "Равносторонний треугольник":
                    Point top = new Point(150, 50);
                    Point bottomLeft = new Point(50, 150);
                    Point bottomRight = new Point(250, 150);

```

```

        g.DrawLine(pen, top, bottomLeft);
        g.DrawLine(pen, bottomLeft, bottomRight);
        g.DrawLine(pen, bottomRight, top);
        Point[] trianglePoints = { top, bottomLeft, bottomRight };

        g.FillPolygon(red_brush, trianglePoints);
        break;
case "Трапеция":
    Point[] Trapezium = new Point[]
    {
        new Point(50, 100),
        new Point(150, 100),
        new Point(200, 200),
        new Point(0, 200)
    };

    g.DrawPolygon(pen, Trapezium);
    g.FillPolygon(yellow_brush, Trapezium);
    break;
case "Окружность":
    int centerX = 150;
    int centerY = 150;
    int radius = 50;
    int diameter = radius * 2;

    int topLeftX = centerX - radius;
    int topLeftY = centerY - radius;

    g.DrawEllipse(pen, topLeftX, topLeftY, diameter, diameter);
    break;
case "Ромб":
    Point[] rhombusPoints = new Point[]
    {
        new Point(150, 50),
        new Point(250, 150),
        new Point(150, 250),
        new Point(50, 150)
    };

    g.DrawPolygon(pen, rhombusPoints);
    g.FillPolygon(black_brush, rhombusPoints);
    break;
case "Равнобедренный треугольник":
    Point[] isoscelesTrianglePoints = new Point[]
    {
        new Point(150, 50),
        new Point(100, 150),
        new Point(200, 150)
    };

    g.DrawPolygon(pen, isoscelesTrianglePoints);
    g.FillPolygon(yellow_brush, isoscelesTrianglePoints);
    break;
case "Параллелограмм":
    Point[] parallelogramPoints = new Point[]
    {
        new Point(50, 50),
        new Point(300, 50),
        new Point(250, 150),
        new Point(0, 150)
    };

    g.DrawPolygon(pen, parallelogramPoints);

```



```

        g.FillPolygon(red_brush, parallelogramPoints);
        break;
    case "Kpyr":
        int x_round = 50;
        int y_round = 50;
        int diameter2 = 100;

        g.DrawEllipse(pen, x_round, y_round, diameter2, diameter2);
        g.FillEllipse(black_brush, x_round, y_round, diameter2,
diameter2);
        break;
    }
}

pictureBox1.Image = drawingBitmap;
}
}
}

```

## **Вывод**

Программа реализованная в интегрированной среде разработки Microsoft Visual Studio на языке Visual C#, представляет собой пример создания программы в режиме Windows Forms Application. Содержит главное меню, которое позволяет пользователю начать работу с приложением, прервать его выполнение или получить справочную информацию.

Основной функционал программы заключается в выводе графических элементов управления на главную экранную форму. Пользователь имеет возможность выбрать плоские геометрические фигуры из списка, представленного в комбинированном списке. Список фигур включает в себя такие элементы, как квадрат, прямоугольный треугольник, эллипс, равнобедренный треугольник, круг, равносторонний треугольник, окружность, ромб, трапеция, параллелограмм и прямоугольник. Кроме того, в практической работе реализована дополнительная фигура - правильный шестиугольник.

Таким образом, разработанная программа представляет собой пример реализации графического приложения с использованием различных элементов управления и функций Visual Studio, обеспечивая пользователю удобный интерфейс для работы с геометрическими фигурами.