**МИНИСТЕРСТВО ТРАНСПОРТА**

**РОССИЙСКОЙ ФЕДЕРАЦИИ**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ**

**АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ**

**УЧРЕЖДЕНИЕ ВЫСШЕГО**

**ОБРАЗОВАНИЯ**

**“РОССИЙСКИЙ УНИВЕРСИТЕТ**

**ТРАНСПОРТА” (МИИТ)**

**Институт транспортной техники и систем**

**управления**

**(ИТТСУ)**

**Кафедра “Управление и защита информации”**

**Тема: “Лабораторная работа №2”**

**Выполнили студенты группы ТКИ-342**

**Лазухин Максим Сергеевич и Макаров Григорий Борисович**

**Специальность 10.05.01 “Компьютерная безопасность”**

**Преподаватель: к.т.н. Сафронов А.И.**

**Москва**

**2024**

1. Формулировка цели работы:

Цель работы заключается в закреплении навыков разработки визуального пользовательского интерфейса, освоении работы с текстовыми файлами и кодировкой в среде Microsoft Visual Studio, а также в освоении реализации настройки множественных состояний объектов с использованием управления компонентами со внутренней индексацией.

2. Описание задачи:

Необходимо разработать программу на языке Visual C# в среде Microsoft Visual Studio в режиме Windows Forms Application. Программа должна представлять собой экранную форму с главным меню, позволяющим начать и прервать работу приложения, а также получить справочную информацию о работе с ним. В графическом элементе управления главной формы должна отображаться геометрическая фигура, выбранная пользователем из списка.

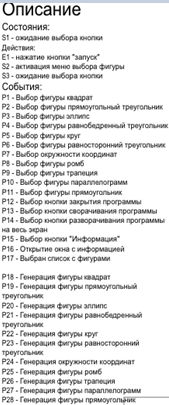
3. Составление диаграммы классов:

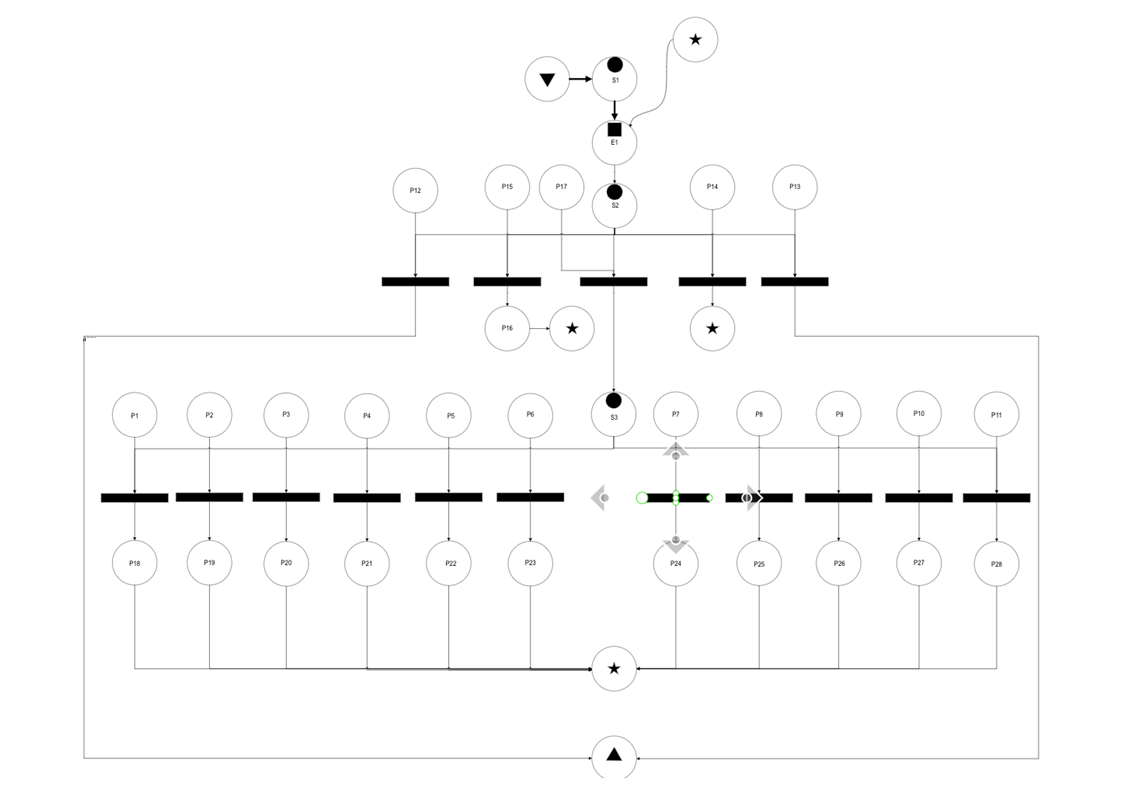
Изображение выглядит как текст, снимок экрана, диаграмма, дизайн

Автоматически созданное описание

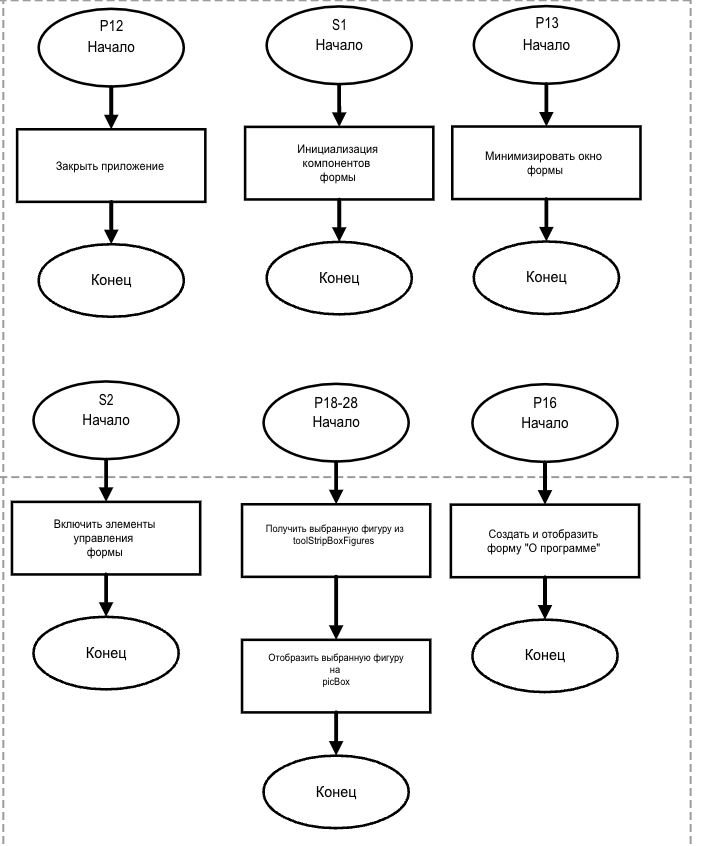
4. Составление сети Петри запрограммированного технологического процесса:







5. Составление схем алгоритмов методов:

  
Изображение выглядит как диаграмма, зарисовка, линия, Технический чертеж

Автоматически созданное описание

6. Подбор тестовых примеров:

- Выбор фигуры из списка и отображение ее на главной форме.

- Нажатие кнопки "Начать работу с приложением" и активация доступа к элементам.

- Получение справочной информации.

- Закрытие приложения.

7. Листинг (код) программного обеспечения:

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.IO;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Drawing.Drawing2D;

namespace Laba2\_Lazukhin\_Makarov\_TKI342

{

public partial class frmMain : Form

{

public frmMain()=>InitializeComponent();

public bool flgEllipse;

public bool flgLine;

public bool flgRectangle; public bool flgText;

bool flag = false;

private void rb4\_CheckedChanged(object sender, EventArgs e)

{

this.pb1.Image = new Bitmap(pb1.Width, pb1.Height);

Graphics myGraphics = Graphics.FromImage(pb1.Image);

Pen pen = new Pen(Brushes.Black, 3);

pen.CustomEndCap = new AdjustableArrowCap(3, 3);

pen.Width = 3;

pen.Color = Color.RoyalBlue;

myGraphics.DrawEllipse(pen, new Rectangle(50, 50, this.pb1.Width - 300, this.pb1.Height - 300));

}

private void rb1\_CheckedChanged(object sender, EventArgs e)

{

this.pb1.Image = new Bitmap(pb1.Width, pb1.Height);

Graphics myGraphics = Graphics.FromImage(pb1.Image);

Point[] pIsoscelesTriang = { new Point(this.pb1.Width / 2, 20), new Point(90, this.pb1.Height - 20), new Point(this.pb1.Width - 90, this.pb1.Height - 20) };

myGraphics.FillPolygon(Brushes.RoyalBlue, pIsoscelesTriang);

}

private void rb3\_CheckedChanged(object sender, EventArgs e)

{

this.pb1.Image = new Bitmap(pb1.Width, pb1.Height);

Graphics myGraphics = Graphics.FromImage(pb1.Image);

Point[] pRhomb = { new Point(0, 150), new Point(150, 70), new Point(300, 150), new Point(150, 230) };

myGraphics.FillPolygon(Brushes.RoyalBlue, pRhomb);

}

private void rb2\_CheckedChanged(object sender, EventArgs e)

{

this.pb1.Image = new Bitmap(pb1.Width, pb1.Height);

Graphics myGraphics = Graphics.FromImage(pb1.Image);

myGraphics.FillEllipse(Brushes.RoyalBlue, new Rectangle(0, 70, this.pb1.Width, this.pb1.Height - 70 \* 2));

}

private void rb5\_CheckedChanged(object sender, EventArgs e)

{

this.pb1.Image = new Bitmap(pb1.Width, pb1.Height);

Graphics myGraphics = Graphics.FromImage(pb1.Image);

myGraphics.FillRectangle(Brushes.RoyalBlue, new Rectangle(70, 70, this.pb1.Width - 70 \* 2, this.pb1.Height - 70 \* 2));

}

private void rb6\_CheckedChanged(object sender, EventArgs e)

{

this.pb1.Image = new Bitmap(pb1.Width, pb1.Height);

Graphics myGraphics = Graphics.FromImage(pb1.Image);

Point[] pPTriangle = { new Point(70, 70), new Point(70, this.pb1.Height - 70), new Point(this.pb1.Width - 70, this.pb1.Height-70) };

myGraphics.FillPolygon(Brushes.RoyalBlue, pPTriangle);

}

private void rb7\_CheckedChanged(object sender, EventArgs e)

{

this.pb1.Image = new Bitmap(pb1.Width, pb1.Height);

Graphics myGraphics = Graphics.FromImage(pb1.Image);

Point[] pParallel = { new Point(70, 70), new Point(300, 70), new Point(230, 230), new Point(0, 230) };

myGraphics.FillPolygon(Brushes.RoyalBlue, pParallel);

}

private void rb8\_CheckedChanged(object sender, EventArgs e)

{

this.pb1.Image = new Bitmap(pb1.Width, pb1.Height);

Graphics myGraphics = Graphics.FromImage(pb1.Image);

Point[] pTrapeze = { new Point(0, 270), new Point(100, 70), new Point(200, 70), new Point(300, 270) };

myGraphics.FillPolygon(Brushes.RoyalBlue, pTrapeze);

}

private void rb9\_CheckedChanged(object sender, EventArgs e)

{

this.pb1.Image = new Bitmap(pb1.Width, pb1.Height);

Graphics myGraphics = Graphics.FromImage(pb1.Image);

var distance = 200F;

var angle = 60F;

var startX = 70;

var startY = 70;

PointF[] pEquilateralTriangle = { new PointF(startX, startY),

new PointF((float)(startX - distance \* Math.Cos(angle)), (float)(startY - distance \* Math.Sin(angle))),

new PointF((float)(startX - distance \* Math.Cos(angle + Math.PI / 3)), (float)(startY - distance \* Math.Sin(angle + Math.PI / 3))) };

myGraphics.FillPolygon(Brushes.RoyalBlue, pEquilateralTriangle);

}

private void rb10\_CheckedChanged(object sender, EventArgs e)

{

this.pb1.Image = new Bitmap(pb1.Width, pb1.Height);

Graphics myGraphics = Graphics.FromImage(pb1.Image);

myGraphics.FillRectangle( Brushes.RoyalBlue, new Rectangle(200, 200, 200,200));

}

private void rb11\_CheckedChanged(object sender, EventArgs e)

{

this.pb1.Image = new Bitmap(pb1.Width, pb1.Height);

Graphics myGraphics = Graphics.FromImage(pb1.Image);

myGraphics.FillEllipse(Brushes.RoyalBlue, new Rectangle(70, 70, this.pb1.Width - 70 \* 2, this.pb1.Height - 70 \* 2));

}

private void btn1\_Click(object sender, EventArgs e)

{

(sender as Button).Enabled = false;

rb1.Enabled = true;

rb2.Enabled = true;

rb3.Enabled = true;

rb4.Enabled = true;

rb5.Enabled = true;

rb6.Enabled = true;

rb7.Enabled = true;

rb8.Enabled = true;

rb9.Enabled = true;

rb10.Enabled = true;

rb11.Enabled = true;

rb12.Enabled = true;

pb1.Visible = true;

}

private void Form1\_Load(object sender, EventArgs e)

{

}

private void lblHelp\_Click(object sender, EventArgs e)

{

}

private void btn3\_Click(object sender, EventArgs e)

{

this.Close();

}

private void chk1\_CheckedChanged(object sender, EventArgs e)

{

(sender as CheckBox).Text = (sender as CheckBox).Checked ?

"Справочная информация" : "Закрыть справочную информацию";

frmChild f = new frmChild();

f.Owner = this;

f.lblHelp.Visible = (sender as CheckBox).Checked;

string path = Application.StartupPath + @"инструкция.txt";

StreamReader sr = new StreamReader(path);

f.lblHelp.Text = sr.ReadToEnd();

sr.Close();

f.ShowDialog();

f.Dispose();

}

private void button1\_Click(object sender, EventArgs e)

{

this.WindowState = FormWindowState.Maximized;

}

private void button2\_Click(object sender, EventArgs e)

{

this.WindowState = FormWindowState.Minimized;

}

private void chk2\_CheckedChanged(object sender, EventArgs e)

{

(sender as CheckBox).Text = (sender as CheckBox).Checked ?

"Стандартное окошко" : "Во весь экран";

//this.WindowState = FormWindowState.Maximized;

if (!flag)

WindowState = FormWindowState.Maximized;

else

WindowState = FormWindowState.Normal;

flag = !flag;

//this.WindowState = FormWindowState.Normal;

}

private void rb12\_CheckedChanged(object sender, EventArgs e)

{

this.pb1.Image = new Bitmap(pb1.Width, pb1.Height);

Graphics myGraphics = Graphics.FromImage(pb1.Image);

Point center = new Point(100, 100); // Координаты центра

int vertexes=10; // Количество вершин

float radius=90; // Радиус

var angle = Math.PI \* 2 / vertexes;

var points = Enumerable.Range(0, vertexes)

.Select(i => PointF.Add(center, new SizeF((float)Math.Sin(i \* angle) \* radius, (float)Math.Cos(i \* angle) \* radius)));

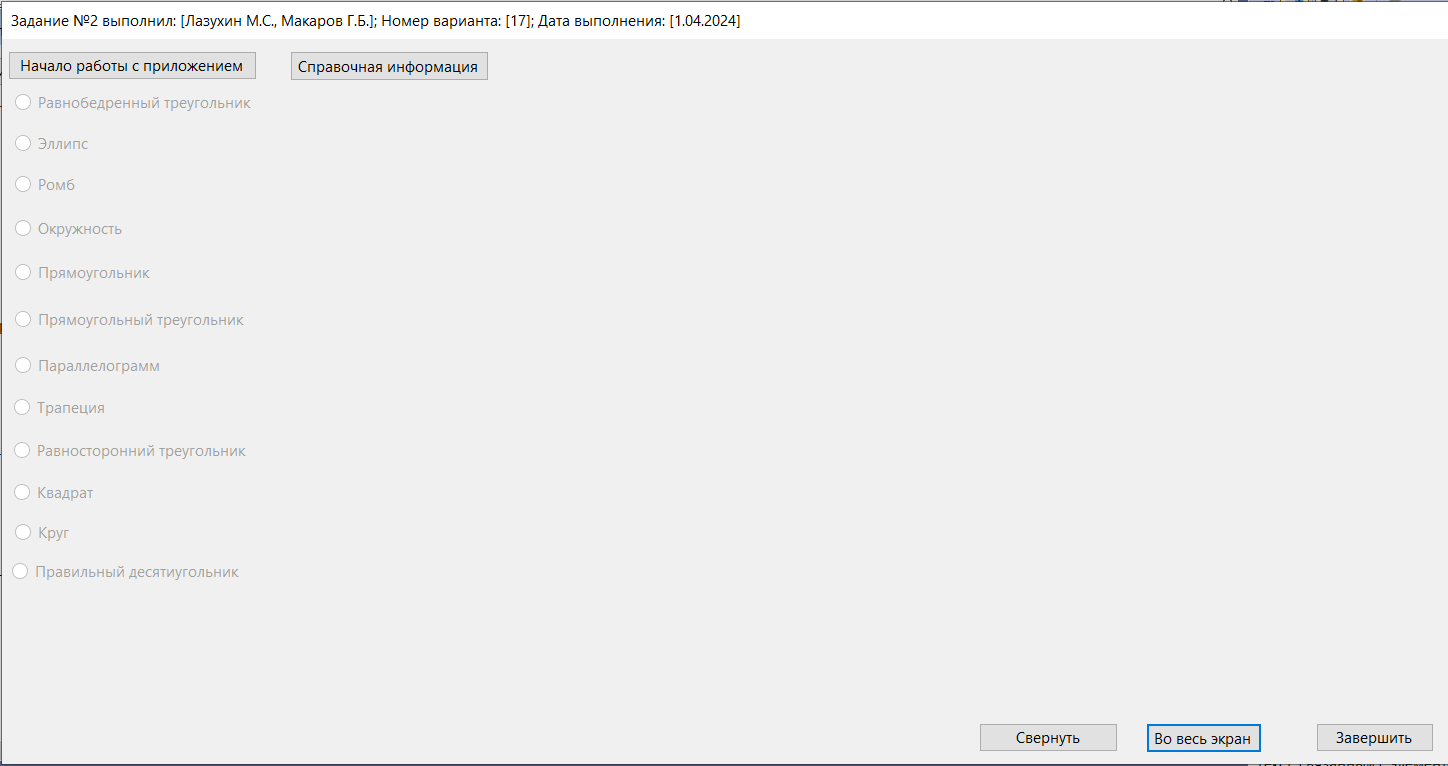
myGraphics.DrawPolygon(Pens.RoyalBlue, points.ToArray());

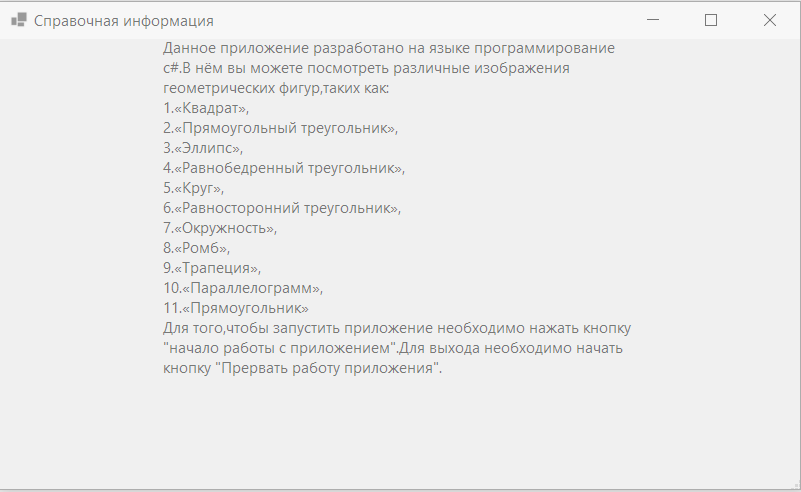
}

}

}

8. Графический пользовательский интерфейс:



Основной интерфейс приложения  
  


Полупрозрачная дочерняя форма

Для начала работы с приложением необходимо нажать кнопку “Начало работы с приложение”. После этого значение Enabled каждого элемента графического интерфейса меняется с false на true

private void btn1\_Click(object sender, EventArgs e)

{

(sender as Button).Enabled = false;

rb1.Enabled = true;

rb2.Enabled = true;

rb3.Enabled = true;

rb4.Enabled = true;

rb5.Enabled = true;

rb6.Enabled = true;

rb7.Enabled = true;

rb8.Enabled = true;

rb9.Enabled = true;

rb10.Enabled = true;

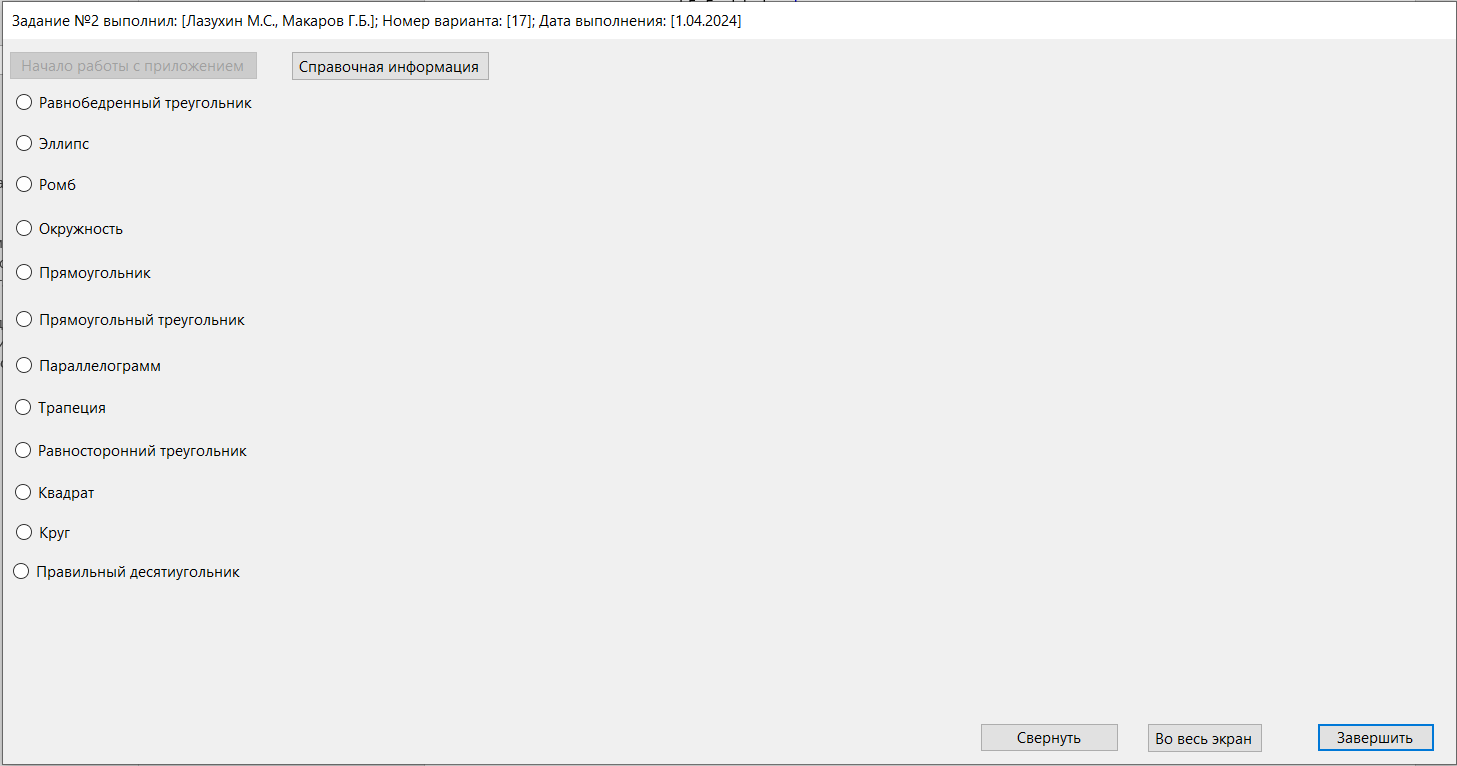
rb11.Enabled = true;

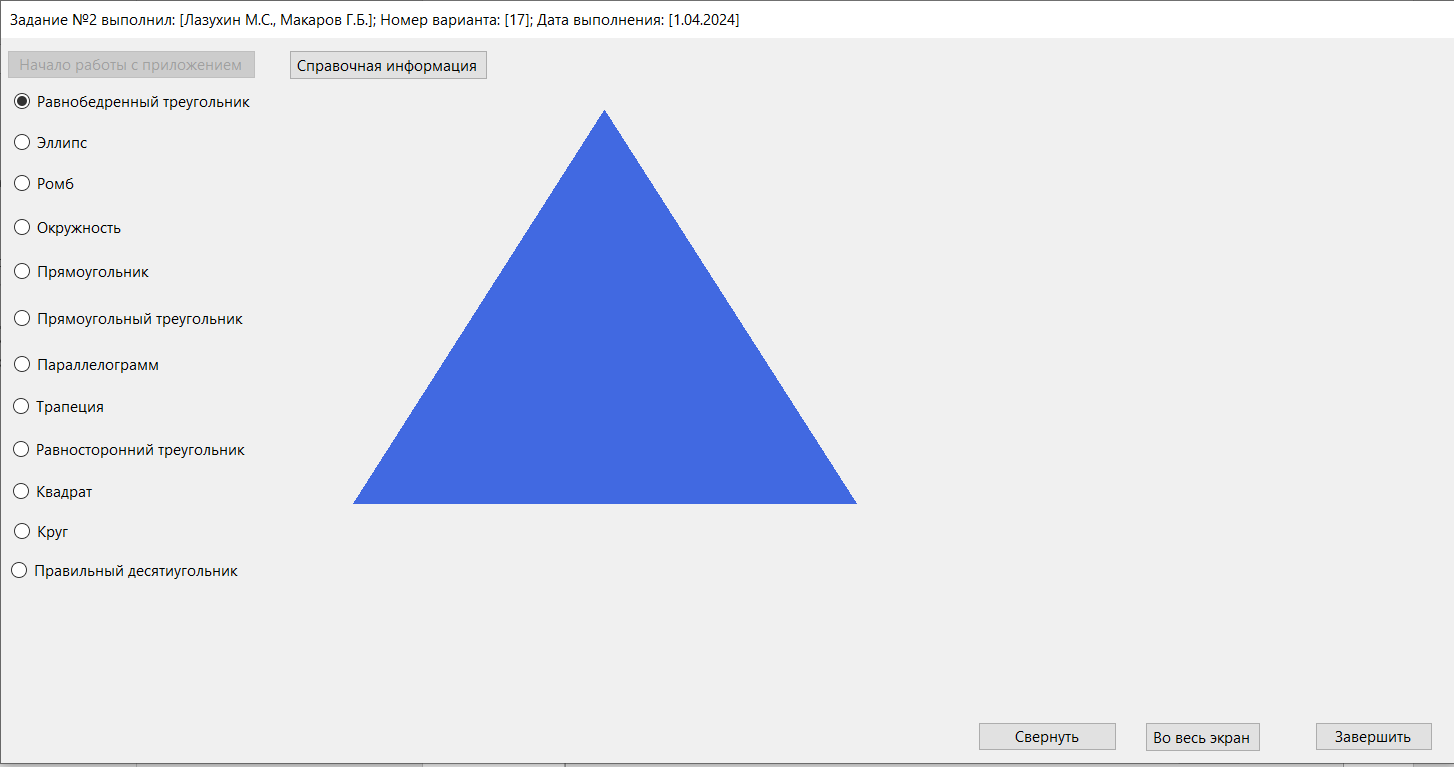
rb3.Enabled = true;

pb1.Visible = true;

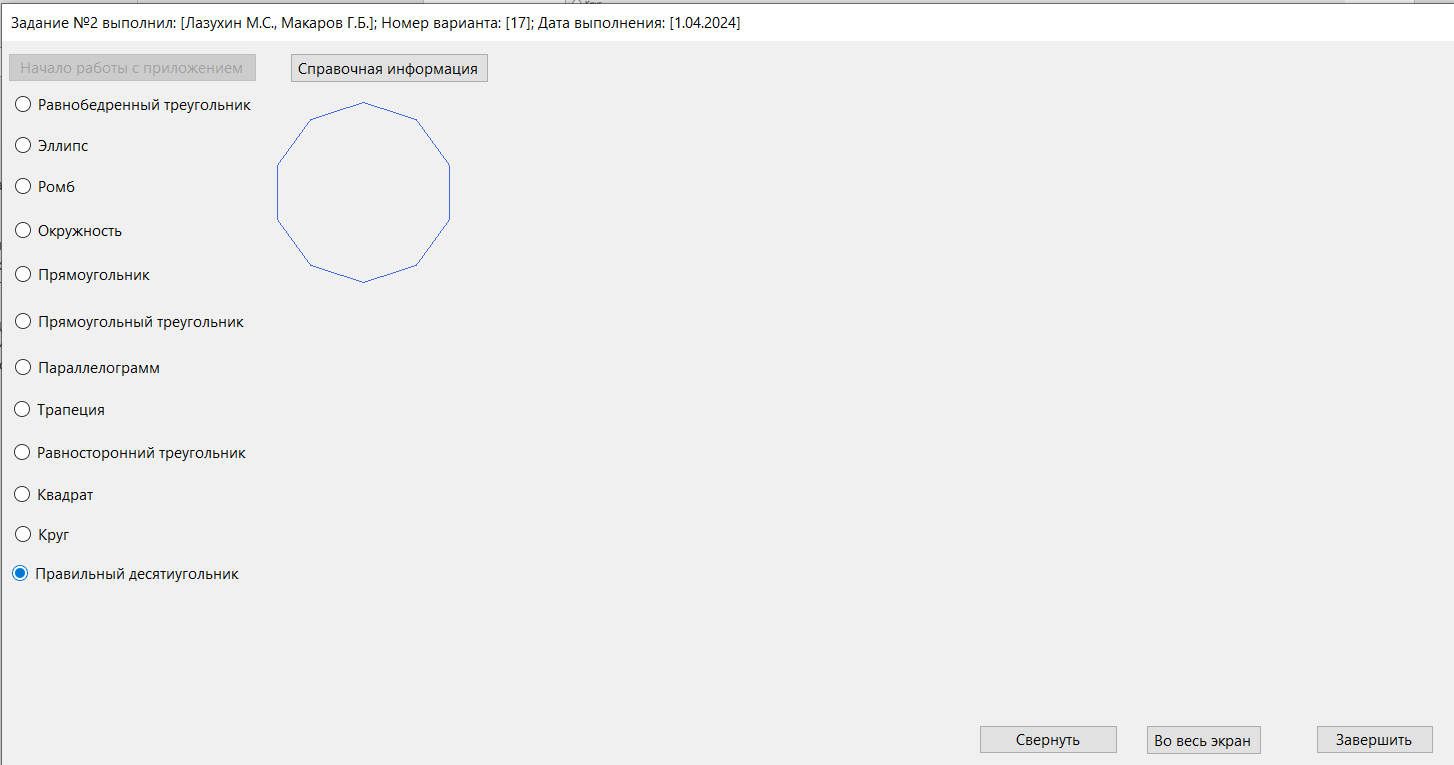
}

Пример работы программы, после нажатия кнопки “Начало работы с программой”





Пример работы программы



2 Пример работы программы программы

10. Вывод:

Было разработано графическое приложение на языке программирования c#, которое строит различные двухмерные геометрические фигуры. В процессе его разработки были получены знания и опыт работы с языком программирования c# в среде программирования Microsoft Visual Studio в режиме Windows Forms Application. Были изучены элементы и их свойства экранной формы “System.Windows.Forms.Form” (было реализовано скрытие элементов до нажатия клавиши “Начало работы с приложением” )Также были получены навыки по построению геометрических фигур в языках программирования.