**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**

**Павлючков Даниил Дмитриевич**

**Рудов Лев Андреевич**

**Специальность 10.05.01 “Компьютерная безопасность”**

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

**Москва**

**2024**

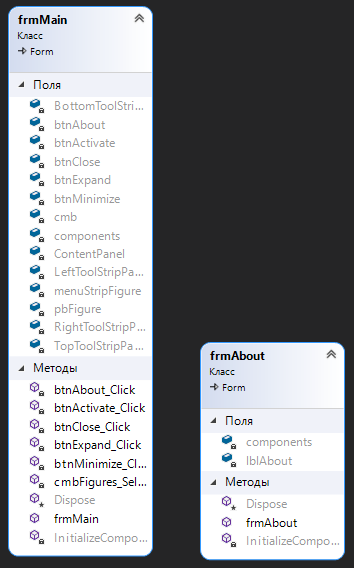
1. Формулировка цели работы:

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:

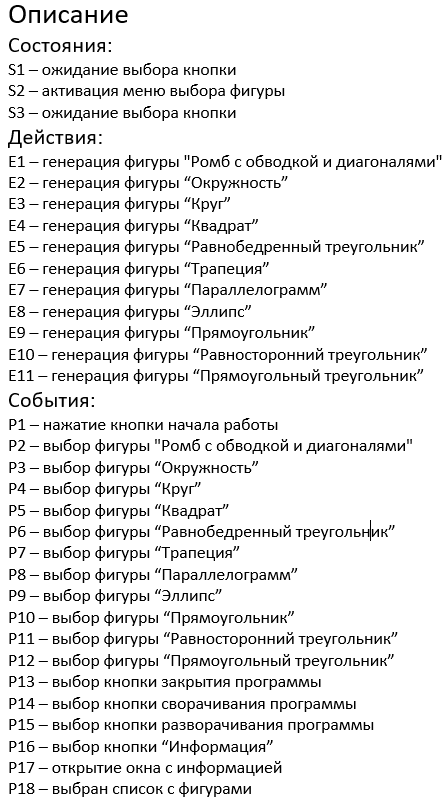
Необходимо разработать программу на языке Visual C# в среде Microsoft Visual Studio в режиме Windows Forms Application. Программа должна представлять собой экранную форму с главным меню, позволяющим начать и прервать работу приложения, а также получить справочную информацию о работе с ним. В графическом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

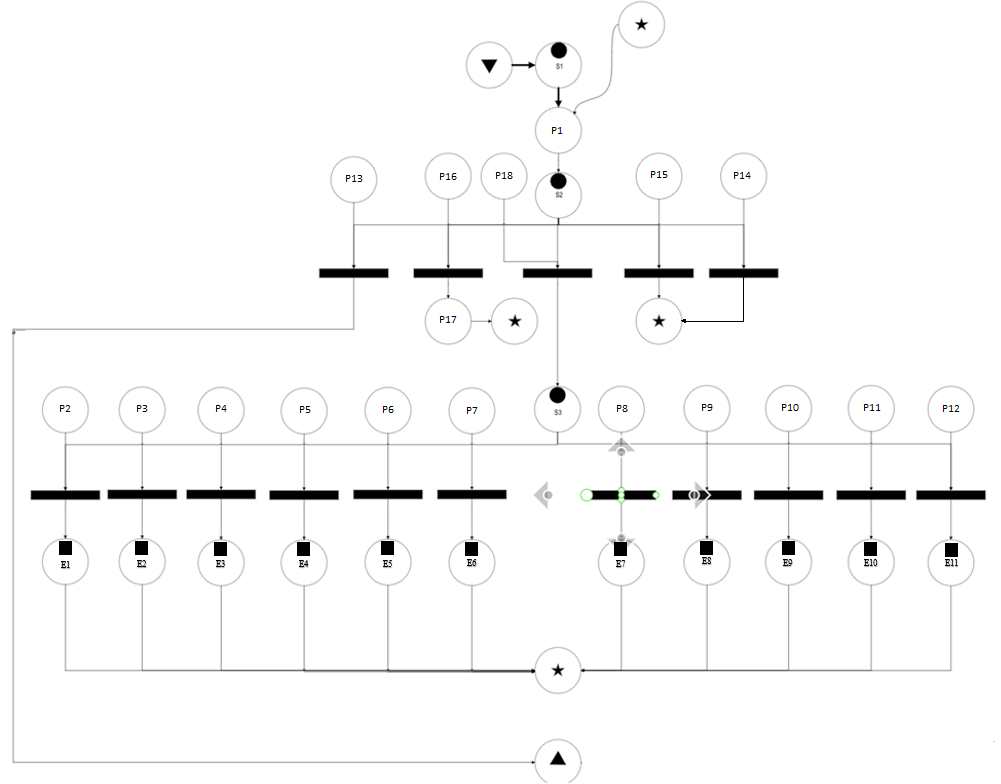
3. Составление диаграммы классов:



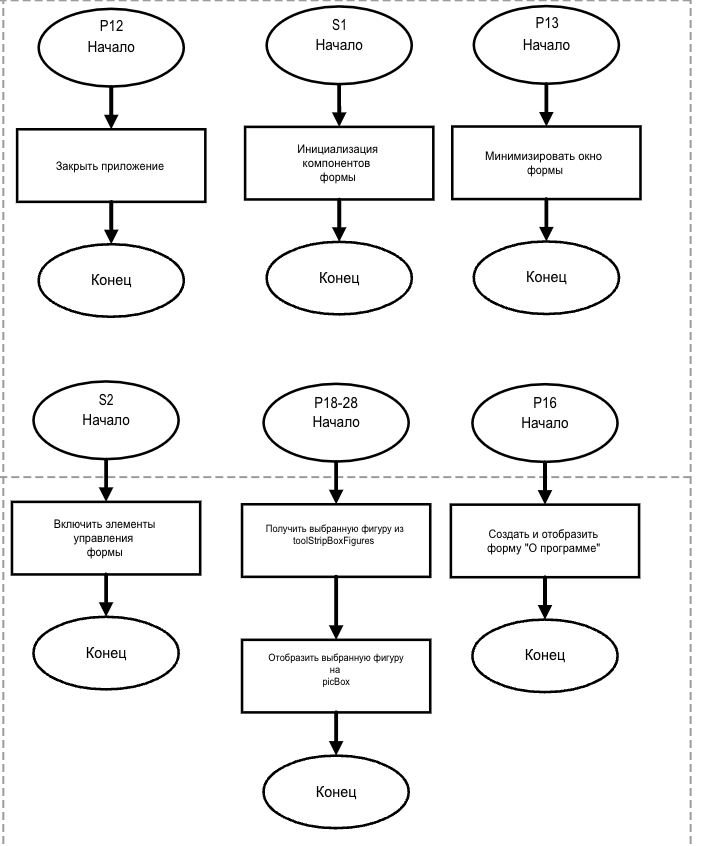
4. Составление сети Петри запрограммированного технологического процесса:







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.Drawing;

using System.Windows.Forms;

namespace Lab2

{

public partial class frmMain : Form

{

public frmMain()

{

InitializeComponent();

}

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

{

Application.Exit();

}

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

{

this.WindowState = FormWindowState.Minimized;

}

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

{

this.btnMinimize.Enabled = true;

this.btnExpand.Enabled = true;

this.btnClose.Enabled = true;

this.btnAbout.Enabled = true;

this.menuStripFigure.Enabled = true;

}

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

{

if (this.WindowState == FormWindowState.Normal)

{

this.WindowState = FormWindowState.Maximized;

}

else {

this.WindowState = FormWindowState.Normal;

}

}

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

{

frmAbout frmAbout = new frmAbout();

frmAbout.Show();

}

private void cmbFigures\_SelectedIndexChanged(object sender, EventArgs e)

{

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

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

switch (this.cmb.Text)

{

case "Ромб с обводкой и диагоналями":

Pen penThick = new Pen(Color.FromArgb(255, 0, 0, 0), 5);

Pen penThin = new Pen(Color.FromArgb(255, 0, 0, 0), 2);

Point[] rhomb = { new Point(10, 150), new Point(160, 70), new Point(310, 150), new Point(160, 230) };

myGraphics.FillPolygon(Brushes.Blue, rhomb);

myGraphics.DrawPolygon(penThick, rhomb);

Point[] rhombDiagonal1 = { new Point(10, 150), new Point(310, 150) };

Point[] rhombDiagonal2 = { new Point(160, 70), new Point(160, 230) };

myGraphics.DrawPolygon(penThin, rhombDiagonal1);

myGraphics.DrawPolygon(penThin, rhombDiagonal2);

break;

case "Окружность":

Pen pen = new Pen(Color.FromArgb(255, 0, 0, 0), 5);

myGraphics.DrawEllipse(pen, new Rectangle(100, 100, this.pbFigure.Width - 200, this.pbFigure.Height - 200));

break;

case "Круг":

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

break;

case "Квадрат":

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

break;

case "Равнобедренный треугольник":

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

myGraphics.FillPolygon(Brushes.Blue, isoscelesTriangle);

break;

case "Трапеция":

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

myGraphics.FillPolygon(Brushes.Blue, trapeze);

break;

case "Ромб":

Point[] rhomb1 = { new Point(0, 150), new Point(150, 70), new Point(300, 150), new Point(150, 230) };

myGraphics.FillPolygon(Brushes.Blue, rhomb1);

break;

case "Параллелограмм":

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

myGraphics.FillPolygon(Brushes.Blue, parallelogram);

break;

case "Эллипс":

myGraphics.FillEllipse(Brushes.Blue, new Rectangle(10, 70, this.pbFigure.Width, this.pbFigure.Height - 70 \* 2));

break;

case "Прямоугольник":

myGraphics.FillRectangle(Brushes.Blue, new Rectangle(10, 100, 300, 100));

break;

case "Равносторонний треугольник":

var distance = 200F;

var angle = 60F;

var startX = 70;

var startY = 70;

PointF[] equilateralTriangle = { 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.Blue, equilateralTriangle);

break;

case "Прямоугольный треугольник":

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

myGraphics.FillPolygon(Brushes.Blue, rightTriangle);

break;

}

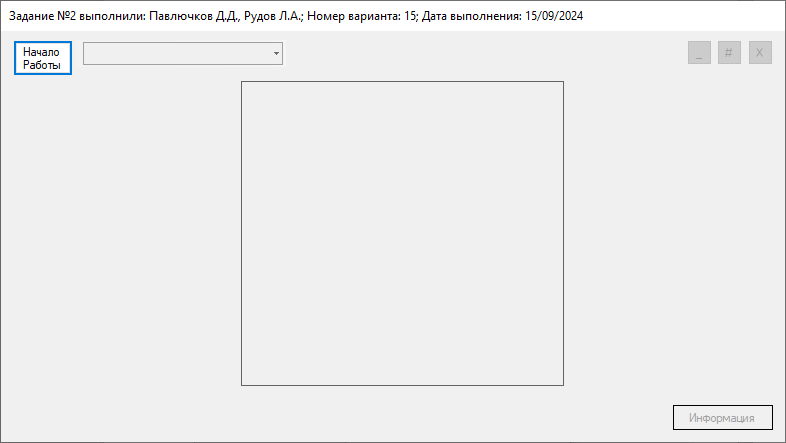
pbFigure.Refresh();

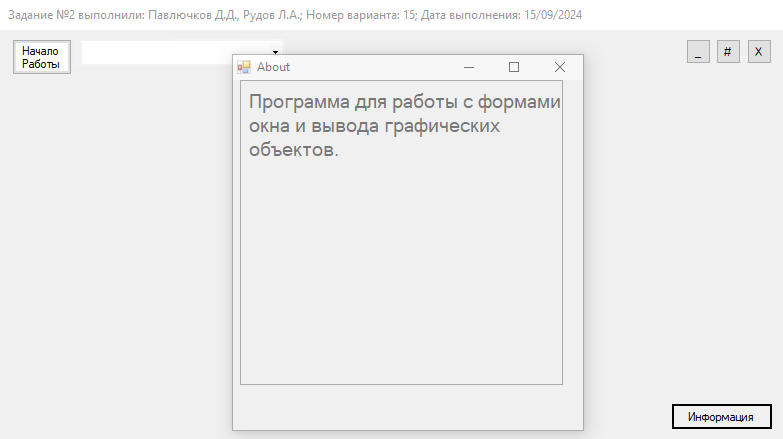
}

}

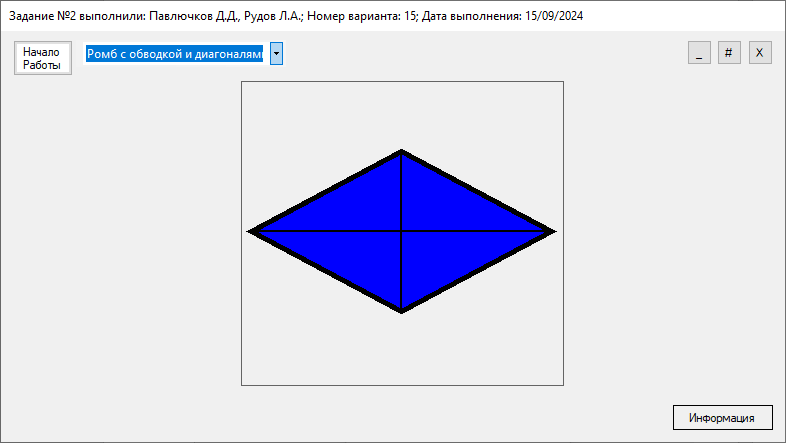
}

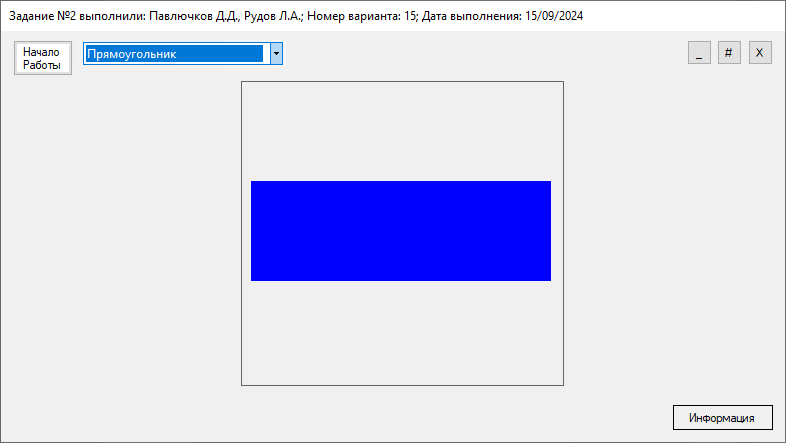
8. Графический пользовательский интерфейс:



Основной интерфейс приложения  
  


Полупрозрачная дочерняя форма





Пример работы программы

9. 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”. Также были получены навыки по построению геометрических фигур в языках программирования.