**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**

**Савин Лев Андреевич   
Потапов Артём Константинович**

**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:

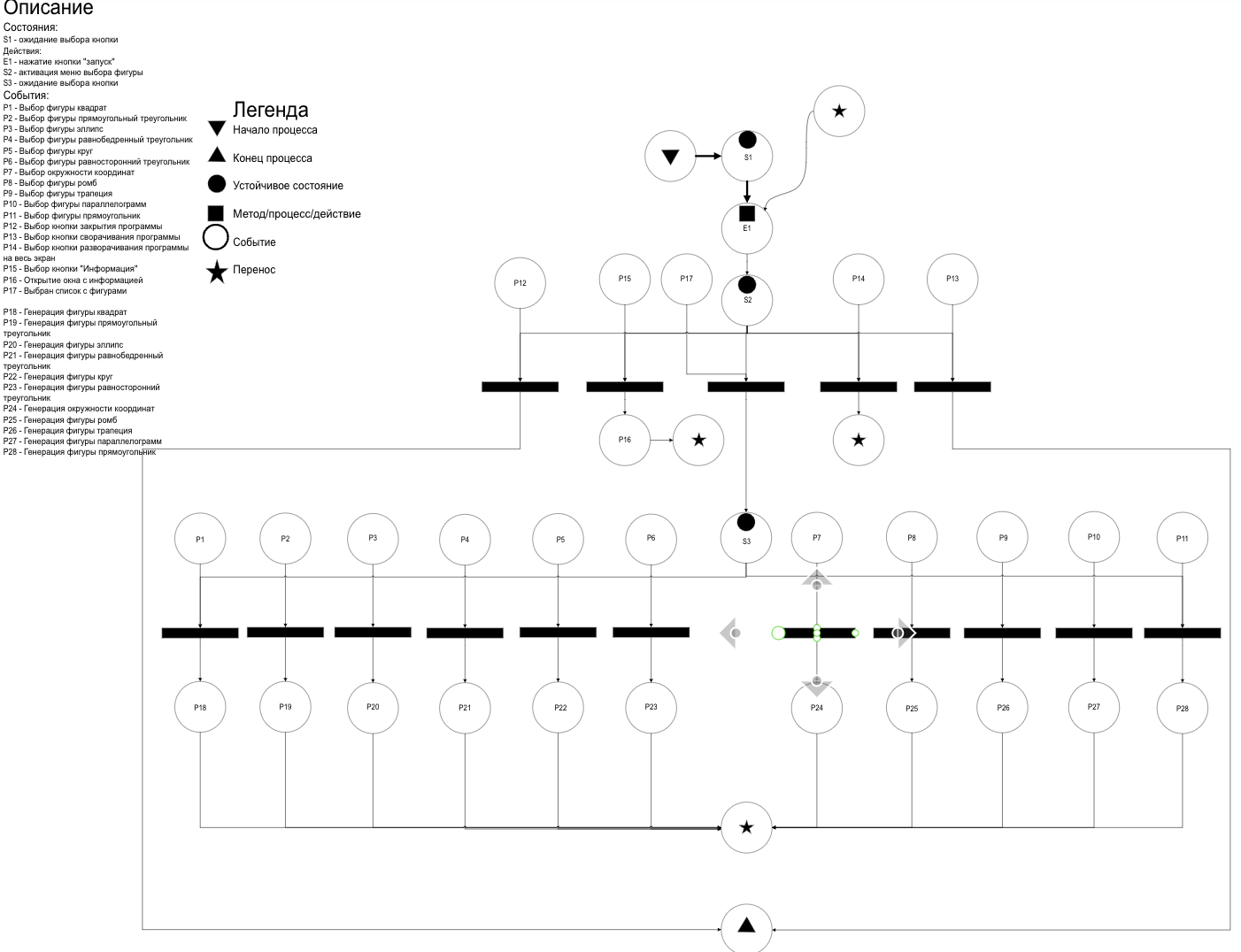
Необходимо разработать программу на языке Visual C# в среде Microsoft Visual Studio в режиме Windows Forms Application. Программа должна представлять собой экранную форму с главным меню, позволяющим начать и прервать работу приложения, а также получить справочную информацию о работе с ним. В графическом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

3. Составление диаграммы классов:

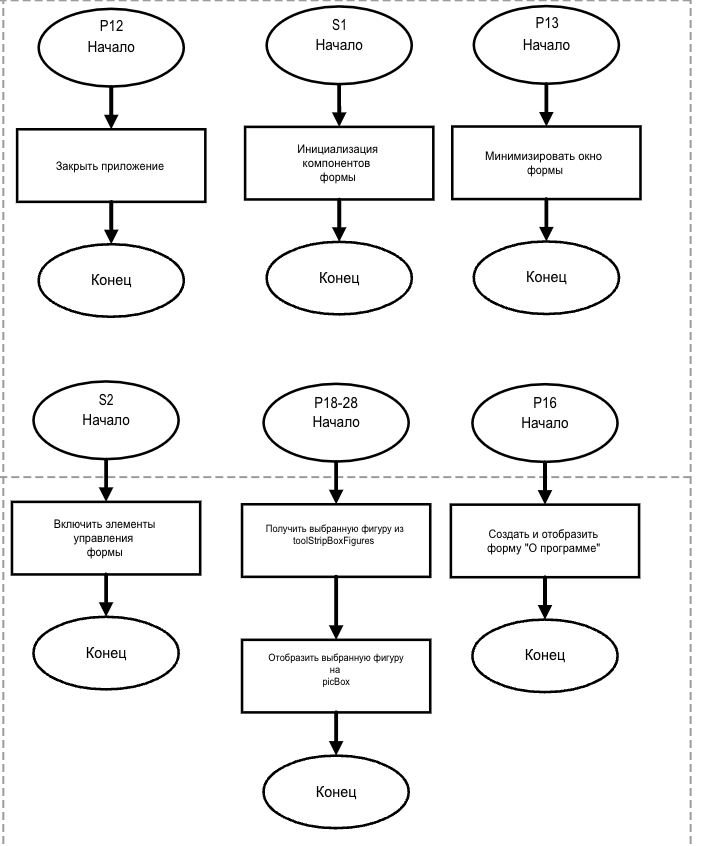
Изображение выглядит как текст, снимок экрана, диаграмма, дизайн

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.Drawing;

using System.Drawing.Drawing2D;

using System.Windows.Forms;

namespace ProgMethods\_Work2\_

{

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 toolStripBoxFigures\_DropDownClosed(object sender, EventArgs e)

{

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

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

switch (this.toolStripBoxFigures.Text)

{

case "Квадрат":

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

break;

case "Прямоугольный треугольник":

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

myGraphics.FillPolygon(Brushes.RoyalBlue, pPTriangle);

break;

case "Эллипс":

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

break;

case "Равнобедренный треугольник":

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

myGraphics.FillPolygon(Brushes.RoyalBlue, pIsoscelesTriang);

break;

case "Круг":

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

break;

case "Равносторонний треугольник":

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);

break;

case "Окружность":

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

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

myGraphics.DrawLine(pen, new Point(0, 150), new Point(300, 150));

myGraphics.DrawLine(pen, new Point(150, 300), new Point(150, 0));

pen.Width = 3;

pen.Color = Color.RoyalBlue;  
 myGraphics.DrawEllipse(pen, new Rectangle(100, 100, this.picBox.Width - 200, this.picBox.Height-200));

myGraphics.DrawString("0", new Font("Times New Roman", 20), Brushes.RoyalBlue, new Point(150,150));

myGraphics.DrawString("1", new Font("Times New Roman", 20), Brushes.RoyalBlue, new Point(200, 150));

myGraphics.DrawString("1", new Font("Times New Roman", 20), Brushes.RoyalBlue, new Point(150, 100));

break;

case "Ромб":

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

myGraphics.FillPolygon(Brushes.RoyalBlue, pRhomb);

break;

case "Трапеция":

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

myGraphics.FillPolygon(Brushes.RoyalBlue, pTrapeze);

break;

case "Параллелограмм":

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

myGraphics.FillPolygon(Brushes.RoyalBlue, pParallel);

break;

case "Прямоугольник":

myGraphics.FillRectangle(Brushes.RoyalBlue, new Rectangle(0, 100, 300, 100));

break;

}

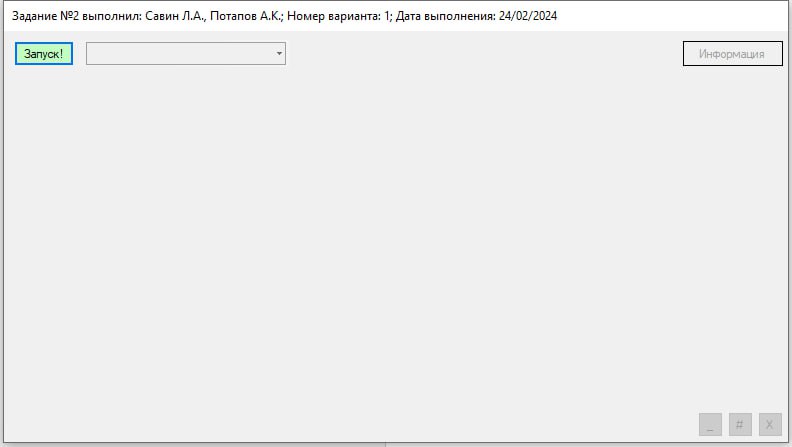
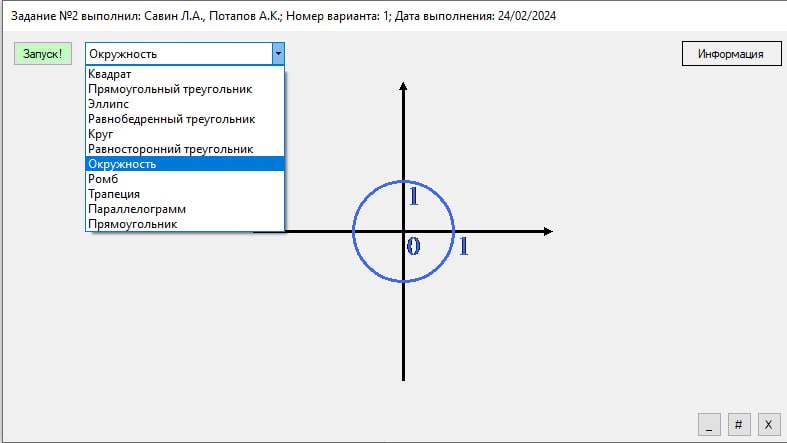
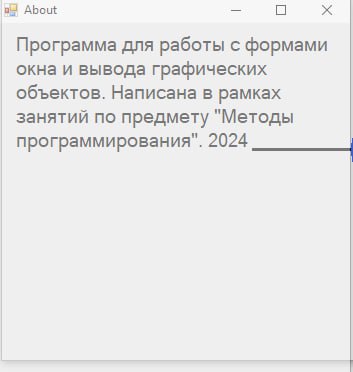
picBox.Refresh();

}

}

}

8. Графический пользовательский интерфейс:

9. Расчёт тестовых примеров:

Изображение выглядит как снимок экрана, текст, программное обеспечение

Автоматически созданное описание  
Полупрозрачная дочерняя форма с текстом из файла  
  
  
Изображение выглядит как текст, Шрифт, снимок экрана, белый

Автоматически созданное описание  
Наименование переменных.  
  
Изображение выглядит как текст, снимок экрана, программное обеспечение, число

Автоматически созданное описаниеИзображение выглядит как текст, электроника, снимок экрана, программное обеспечение

Автоматически созданное описание  
Свойства доступности элементов управления.

10. Вывод:

В ходе работы была разработана программа на языке Visual C# в среде Microsoft Visual Studio, удовлетворяющая требованиям поставленной задачи. Программа успешно реализует вывод геометрических фигур на главную форму, имеет функциональное главное меню и предоставляет справочную информацию для пользователя.