МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ

РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное бюджетное образовательное учреждение

высшего образования

«Российский государственный университет им. А.Н. Косыгина (Технологии. Дизайн. Искусство)»

Кафедра автоматизированных систем обработки информации и управления

Отчет по лабораторной работе № 9

по дисциплине «Объектно-ориентированное программирование»

Тема: Делегаты

Выполнил: Гоянов Р.Р., группа МВА-220

Проверил: Кузьмина Т.М.

Цель работы

Научиться работать с делегатами.

Задание

Вариант 6:

Рисунок - дом.

Части: крыша, окно, дверь, труба, фасад

Код класса MainWindow

public partial class MainWindow : Window

{

public static MainWindow mainWindowInstance;

public static Goyanov goyanovInstance;

public Shape[] drawnShapes = new Shape[5];

public delegate void Draw();

public Draw draw;

public MainWindow()

{

InitializeComponent();

mainWindowInstance = this;

goyanovInstance = new Goyanov();

}

private static void drawFacadeMethod()

{

Rectangle rectangle = new Rectangle();

rectangle.Stroke = Brushes.Black;

rectangle.Fill = Brushes.Chocolate;

rectangle.Margin = new Thickness(40, 90, 40, 10);

mainWindowInstance.bottomHolst.Children.Add(rectangle);

mainWindowInstance.drawnShapes[2] = rectangle;

}

public static void DrawFacade() => drawFacadeMethod();

public void DrawWindow()

{

Rectangle rectangle = new Rectangle();

rectangle.Stroke = Brushes.Black;

rectangle.Fill = Brushes.Gray;

rectangle.Width = 30;

rectangle.Height = 30;

rectangle.Margin = new Thickness(20, 90, 80, 10);

mainWindowInstance.upperHolst.Children.Add(rectangle);

mainWindowInstance.drawnShapes[3] = rectangle;

}

public static void DrawRoof()

{

Polygon triangle = new Polygon();

triangle.Stroke = Brushes.Black;

triangle.Fill = Brushes.Maroon;

Point leftCorner = new Point(30, 90);

Point upperCorner = new Point(117, 30);

Point rightCorner = new Point(204, 90);

PointCollection collection = new PointCollection();

collection.Add(leftCorner);

collection.Add(upperCorner);

collection.Add(rightCorner);

triangle.Points = collection;

mainWindowInstance.upperHolst.Children.Add(triangle);

mainWindowInstance.drawnShapes[4] = triangle;

}

private void drawButton\_Click(object sender, RoutedEventArgs e)

{

upperHolst.Children.Remove(drawnShapes[4]);

upperHolst.Children.Remove(drawnShapes[3]);

upperHolst.Children.Remove(drawnShapes[1]);

bottomHolst.Children.Remove(drawnShapes[2]);

bottomHolst.Children.Remove(drawnShapes[0]);

if (draw != null) draw();

}

private void drawFacadeCheckBox\_Checked(object sender, RoutedEventArgs e)

{

if (drawFacadeCheckBox.IsChecked.Value) draw += DrawFacade;

else draw -= DrawFacade;

}

private void drawWindowCheckBox\_Checked(object sender, RoutedEventArgs e)

{

if (drawWindowCheckBox.IsChecked.Value) draw += DrawWindow;

else draw -= DrawWindow;

}

private void drawDoorCheckBox\_Checked(object sender, RoutedEventArgs e)

{

if (drawDoorCheckBox.IsChecked.Value) draw += goyanovInstance.DrawDoor;

else draw -= goyanovInstance.DrawDoor;

}

private void drawRoofCheckBox\_Checked(object sender, RoutedEventArgs e)

{

if (drawRoofCheckBox.IsChecked.Value) draw += DrawRoof;

else draw -= DrawRoof;

}

private void drawTrubaCheckBox\_Checked(object sender, RoutedEventArgs e)

{

if (drawTrubaCheckBox.IsChecked.Value) draw += Goyanov.DrawPipe;

else draw -= Goyanov.DrawPipe;

}

}

Код класса Goyanov

public class Goyanov

{

public static void DrawPipe()

{

Rectangle rectangle = new Rectangle();

rectangle.Stroke = Brushes.Black;

rectangle.Fill = Brushes.Peru;

rectangle.Margin = new Thickness(50, 20, 160, 144);

MainWindow.mainWindowInstance.bottomHolst.Children.Add(rectangle);

MainWindow.mainWindowInstance.drawnShapes[0] = rectangle;

}

public void DrawDoor()

{

Rectangle rectangle = new Rectangle();

rectangle.Stroke = Brushes.Black;

rectangle.Fill = Brushes.Peru;

rectangle.Margin = new Thickness(130, 130, 60, 10);

MainWindow.mainWindowInstance.upperHolst.Children.Add(rectangle);

MainWindow.mainWindowInstance.drawnShapes[1] = rectangle;

}

}

Результаты выполнения программы

Начальное состояние программы представлено на рис. 1

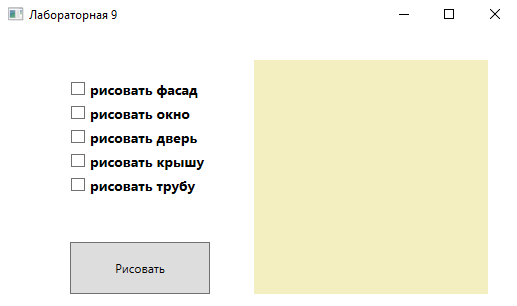


рис. 1

Состояние программы при всех проставленных галочках представлено на рис. 2

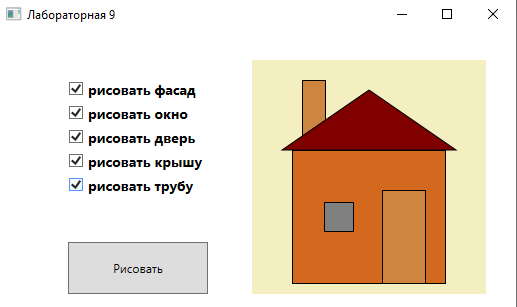


рис. 2

Состояние программы при частичном проставлении галочек представлено на рис. 3

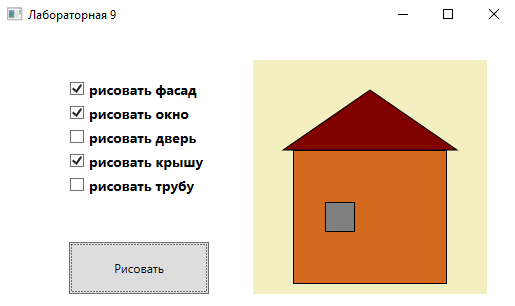


рис. 3

Состояние программы при оставленной одной галочке представлено на рис. 4

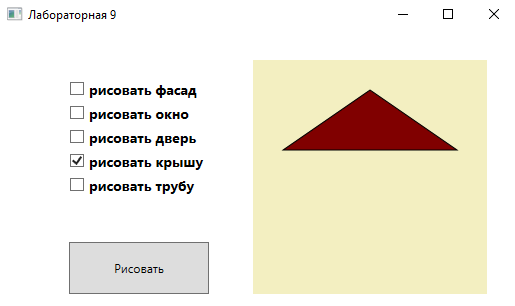


рис. 4