МИНИСТРЕСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ

ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ

ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«ЛИПЕЦКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Кафедра автоматизированных систем управления Отчёт по Индивидуальному домашнему заданию №9

по ООП

«Разработка графического редактора на языке С++ с использованием механизмов ООП»

Вариант 9

Выполнила:

студентка гр. ПМ-21-1 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Кушалина А. А.

Проверил:

доц., к.п.н. кафедры АСУ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Кургасов В. В.

2022г

## Оглавление

[Цель работы 3](#_Toc17870)

[Задание кафедры 4](#_Toc17871)

[Код программы: 5](#_Toc17872)

[Работа программы: 11](#_Toc17873)

[Вывод 16](#_Toc17874)

# Цель работы

**Цель работы:** Закрепить навыки использования механизмов ООП на примере реализации графического редактора.

# Задание кафедры

**Задание кафедры:** Реализовать на языке C++ редактор графической схемы по варианту. В ходе выполнения работы обязательно применение объектно ориентированных возможностей языка C++: наследования и динамического полиморфизма. Каждый тип элемента схемы должен быть представлен в программе в виде отдельного класса, который наследован от базового класса "графический элемент" (имеющего чисто виртуальную функцию прорисовки). Также необходим один класс "поле рисования", который содержит все графические элементы и отвечает за вызов функций прорисовки. Хранение графических элементов осуществляется с использованием контейнеров стандартной библиотеки C++

. Вариант 9: Планировка гостинной

# Код программы:

*MainWindow.cs(code-behind) :*

using System;

using System.Collections.Generic; using System.Linq; using System.Text; using System.Threading.Tasks; using System.Windows; using System.Windows.Controls; using System.Windows.Data; using System.Windows.Documents; using System.Windows.Input; using System.Windows.Media; using System.Windows.Media.Imaging; using System.Windows.Navigation; using System.Windows.Shapes;

namespace WpfApp3

{

/// <summary>

/// Логика взаимодействия для MainWindow.xaml

/// </summary>

public partial class MainWindow : Window

{

int numd = 0; int i = 0;

List<Path> cg = new List<Path>();

Vector relativeMousePos;

FrameworkElement draggedObject; List<Point> coord = new List<Point>(); public MainWindow()

{

InitializeComponent();

}

private void geom\_MouseDoubleClick(object sender, MouseButtonEventArgs

e)//фокус на составную фигуру фигуру при нажатии

{

Path combFig = sender as Path; numd = int.Parse((string)(combFig.Tag)); } void StartDrag(object sender, MouseButtonEventArgs e)//начала перетаскивания

{

draggedObject = (FrameworkElement)sender;

relativeMousePos = e.GetPosition(draggedObject) - new Point(); draggedObject.MouseMove += OnDragMove; draggedObject.LostMouseCapture += OnLostCapture;

draggedObject.MouseUp += OnMouseUp;

Mouse.Capture(draggedObject);

}

void OnDragMove(object sender, MouseEventArgs e)//во время перетаскивания {

UpdatePosition(e);

}

void UpdatePosition(MouseEventArgs e)//обновление позиции

{

var point = e.GetPosition(Gostinnaya); var newPos = point - relativeMousePos; Canvas.SetLeft(draggedObject, newPos.X);

Canvas.SetTop(draggedObject, newPos.Y);

}

void OnMouseUp(object sender, MouseButtonEventArgs e)//когда кнопка мыши не нажата {

FinishDrag(sender, e);

Mouse.Capture(null);

}

void OnLostCapture(object sender, MouseEventArgs e)//когда управление жлементом потеряно

{

FinishDrag(sender, e);

}

void FinishDrag(object sender, MouseEventArgs e)//событие окончания переноса

{

draggedObject.MouseMove -= OnDragMove; draggedObject.LostMouseCapture -= OnLostCapture;

draggedObject.MouseUp -= OnMouseUp;

UpdatePosition(e);

}

void Add(object sender, RoutedEventArgs e)//добавление

{

if (BraButton.IsChecked == true)//кровати

{

EllipseGeometry first = new EllipseGeometry(); first.Center = new Point(75, 75); first.RadiusX = 50; first.RadiusY = 50;

EllipseGeometry second = new EllipseGeometry(); second.Center = new Point(125, 75); second.RadiusX = 50; second.RadiusY = 50;

CombinedGeometry bra = new CombinedGeometry(first, second);

Path braPath = new Path(); braPath.Data = bra; braPath.Fill = Brushes.Yellow; braPath.Stroke = Brushes.Black; braPath.StrokeThickness = 1; coord.Add(new Point(100,75)); cg.Add(braPath);

braPath.Tag = (cg.Count - 1).ToString();

braPath.MouseLeftButtonDown += new MouseButtonEventHandler(StartDrag); braPath.MouseLeftButtonDown += new

MouseButtonEventHandler(geom\_MouseDoubleClick); Canvas.SetLeft(braPath, 100); Canvas.SetTop(braPath, 100);

Gostinnaya.Children.Add(braPath);

}

if (SofaButton.IsChecked == true)//Углового дивана {

RectangleGeometry firstRectangle = new RectangleGeometry(); firstRectangle.Rect = new Rect(0, 0, 360, 200);

RectangleGeometry secontRectangle = new RectangleGeometry(); secontRectangle.Rect = new Rect(200, 200, 160, 320); CombinedGeometry sofa = new CombinedGeometry(firstRectangle, secontRectangle);

Path myPath = new Path(); myPath.Data = sofa; myPath.Fill = Brushes.Red; myPath.Stroke = Brushes.Black; myPath.StrokeThickness = 1; cg.Add(myPath);

coord.Add(new Point(250, 300)); myPath.Tag = (cg.Count - 1).ToString();

myPath.MouseLeftButtonDown += new MouseButtonEventHandler(StartDrag); myPath.MouseLeftButtonDown += new

MouseButtonEventHandler(geom\_MouseDoubleClick);

Canvas.SetLeft(myPath, 100);

Canvas.SetTop(myPath, 100);

Gostinnaya.Children.Add(myPath);

}

if(TableButton.IsChecked == true)

{

RectangleGeometry table = new RectangleGeometry(); table.Rect = new Rect(new Size(275,275));

Path myPath = new Path(); myPath.Data = table; myPath.Fill = Brushes.RosyBrown; myPath.Stroke = Brushes.Black; myPath.StrokeThickness = 1; cg.Add(myPath);

coord.Add(new Point(137.5, 137.5)); myPath.Tag = (cg.Count - 1).ToString();

myPath.MouseLeftButtonDown += new MouseButtonEventHandler(StartDrag); myPath.MouseLeftButtonDown += new

MouseButtonEventHandler(geom\_MouseDoubleClick);

Canvas.SetLeft(myPath, 100);

Canvas.SetTop(myPath, 100);

Gostinnaya.Children.Add(myPath);

}

if(ChairButton.IsChecked == true)

{

RectangleGeometry table = new RectangleGeometry(); table.Rect = new Rect(new Size(75, 120)); EllipseGeometry first = new EllipseGeometry(); first.Center = new Point(0, 60); first.RadiusX = 60; first.RadiusY = 60;

CombinedGeometry chair = new CombinedGeometry(table,first); chair.GeometryCombineMode = GeometryCombineMode.Exclude;

Path myPath = new Path(); myPath.Data = chair; myPath.Fill = Brushes.SandyBrown; myPath.Stroke = Brushes.Black; myPath.StrokeThickness = 1; cg.Add(myPath); coord.Add(new Point(50, 50)); myPath.Tag = (cg.Count - 1).ToString();

myPath.MouseLeftButtonDown += new MouseButtonEventHandler(StartDrag); myPath.MouseLeftButtonDown += new

MouseButtonEventHandler(geom\_MouseDoubleClick); Canvas.SetLeft(myPath, 100);

Canvas.SetTop(myPath, 100);

Gostinnaya.Children.Add(myPath);

}

if (shkafbutton.IsChecked == true)//Углового дивана

{

RectangleGeometry firstRectangle = new RectangleGeometry(); firstRectangle.Rect = new Rect(180,260, 20, 20);

RectangleGeometry secontRectangle = new RectangleGeometry(); secontRectangle.Rect = new Rect(200, 200, 220,500); CombinedGeometry sofa = new CombinedGeometry(firstRectangle, secontRectangle);

Path myPath = new Path(); myPath.Data = sofa; myPath.Fill = Brushes.Brown; myPath.Stroke = Brushes.Black; myPath.StrokeThickness = 1; cg.Add(myPath);

coord.Add(new Point(300, 380)); myPath.Tag = (cg.Count - 1).ToString();

myPath.MouseLeftButtonDown += new MouseButtonEventHandler(StartDrag); myPath.MouseLeftButtonDown += new

MouseButtonEventHandler(geom\_MouseDoubleClick);

Canvas.SetLeft(myPath, 100);

Canvas.SetTop(myPath, 100);

Gostinnaya.Children.Add(myPath);

}

if (TVButton.IsChecked == true)//Углового дивана

{

RectangleGeometry firstRectangle = new RectangleGeometry(); firstRectangle.Rect = new Rect(405, 540,225, 225); EllipseGeometry second = new EllipseGeometry(); second.Center = new Point(405, 652.5); second.RadiusX = 33.75; second.RadiusY = 122.5;

CombinedGeometry sofa = new CombinedGeometry(firstRectangle, second);

Path myPath = new Path(); myPath.Data = sofa; myPath.Fill = Brushes.DarkGray; myPath.Stroke = Brushes.Black; myPath.StrokeThickness = 1; coord.Add(new Point(400, 625)); cg.Add(myPath);

myPath.Tag = (cg.Count - 1).ToString();

myPath.MouseLeftButtonDown += new MouseButtonEventHandler(StartDrag); myPath.MouseLeftButtonDown += new

MouseButtonEventHandler(geom\_MouseDoubleClick);

Canvas.SetLeft(myPath, 100);

Canvas.SetTop(myPath, 100);

Gostinnaya.Children.Add(myPath);

}

if(def\_sofabutton.IsChecked == true)

{

RectangleGeometry firstRectangle = new RectangleGeometry(); firstRectangle.Rect = new Rect(180, 0, 20, 40);

RectangleGeometry secontRectangle = new RectangleGeometry(); secontRectangle.Rect = new Rect(200, 0, 200, 320); RectangleGeometry thrid = new RectangleGeometry(); thrid.Rect = new Rect(180, 280, 20, 40); GeometryGroup sofa = new GeometryGroup(); sofa.Children.Add(firstRectangle); sofa.Children.Add(secontRectangle);

sofa.Children.Add(thrid); Path myPath = new Path(); myPath.Data = sofa; myPath.Fill = Brushes.Red; myPath.Stroke = Brushes.Black; coord.Add(new Point(200, 200)); cg.Add(myPath); myPath.StrokeThickness = 1; myPath.Tag = (cg.Count - 1).ToString();

myPath.MouseLeftButtonDown += new MouseButtonEventHandler(StartDrag); myPath.MouseLeftButtonDown += new

MouseButtonEventHandler(geom\_MouseDoubleClick);

Canvas.SetLeft(myPath, 100);

Canvas.SetTop(myPath, 100);

Gostinnaya.Children.Add(myPath);

}

}

private void Angle\_setter(object sender, RoutedEventArgs e)//поворачиватель

{

i = int.Parse(tb.Text);

RotateTransform rotate = new RotateTransform(); rotate.CenterX =coord[numd].X; rotate.CenterY = coord[numd].Y; rotate.Angle = i;

cg[numd].RenderTransform = rotate; i = 0;

}

private void Delete (object sender, EventArgs e)

{

Gostinnaya.Children.Remove(cg[numd]); cg.Remove(cg[numd]); coord.Remove(coord[numd]);

}

} }

*MainWindow.xaml:*

<Window

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation" xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml" xmlns:d="http://schemas.microsoft.com/expression/blend/2008" xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006" xmlns:local="clr-namespace:WpfApp3" x:Class="WpfApp3.MainWindow" mc:Ignorable="d"

Title="MainWindow" Height="1032.6" Width="1179" MinWidth="800" MinHeight="450"

MaxWidth="1920" MaxHeight="1080" ScrollViewer.VerticalScrollBarVisibility="Disabled">

<Grid Margin="0,0,40,-129">

<Grid.ColumnDefinitions>

<ColumnDefinition Width="82\*"/>

<ColumnDefinition Width="17\*"/>

</Grid.ColumnDefinitions>

<Border BorderBrush="Black" BorderThickness="10" CornerRadius="10"

Margin="0,0,0,125">

<Canvas x:Name="Gostinnaya" Background="GhostWhite" Focusable ="True"

MaxWidth="3000" MaxHeight="1032.6" ClipToBounds="True" />

</Border>

<Grid HorizontalAlignment="Left" Height="1006" VerticalAlignment="Top"

Width="232" Grid.Column="1" Margin="5,0,-43,0">

<ListBox HorizontalAlignment="Left" Height="164"

VerticalAlignment="Bottom" Width="168" Margin="33,0,0,795" Foreground="White"

BorderBrush="White">

<ListBox.Background>

<LinearGradientBrush EndPoint="0.5,1" StartPoint="0.5,0">

<GradientStop Color="Black" Offset="0"/>

<GradientStop Color="#FFFBFBFB" Offset="1"/>

<GradientStop Color="WhiteSmoke"/>

</LinearGradientBrush>

</ListBox.Background>

<RadioButton x:Name="TableButton" Content="Стол"

HorizontalAlignment="Left" VerticalAlignment="Top" Background="White" Height="15"

Width="48"/>

<RadioButton x:Name="shkafbutton" Content="Шкаф"

HorizontalAlignment="Left" VerticalAlignment="Top" Background="White"/>

<RadioButton x:Name="SofaButton" Content="Угловой Диван"

Width="101"/>

<RadioButton x:Name="BraButton" Content="Люстра" Width="101"/>

<RadioButton x:Name="ChairButton" Content="Кресло" Width="101"/>

<RadioButton x:Name="TVButton" Content="Телевизор" Width="101"/>

<RadioButton x:Name="def\_sofabutton" Content="Обычный диван"

Width="113"/>

</ListBox>

<TextBox x:Name="tb" Margin="18,475,10,476" FontSize="36" />

<Button Content="Добавить" Click="Add" Height="56" Canvas.Left="669"

Canvas.Top="149" Margin="10,208,17,715"/>

<Button Content="Повернуть" Click="Angle\_setter" Height="56"

Margin="18,522,10,401"/>

<Button Content="Удалить" Click="Delete" Height="56"

Margin="11,890,17,60"/>

<Label Content="Угол поворота:" HorizontalAlignment="Left"

Margin="66,441,0,0" VerticalAlignment="Top" FontSize="18"/>

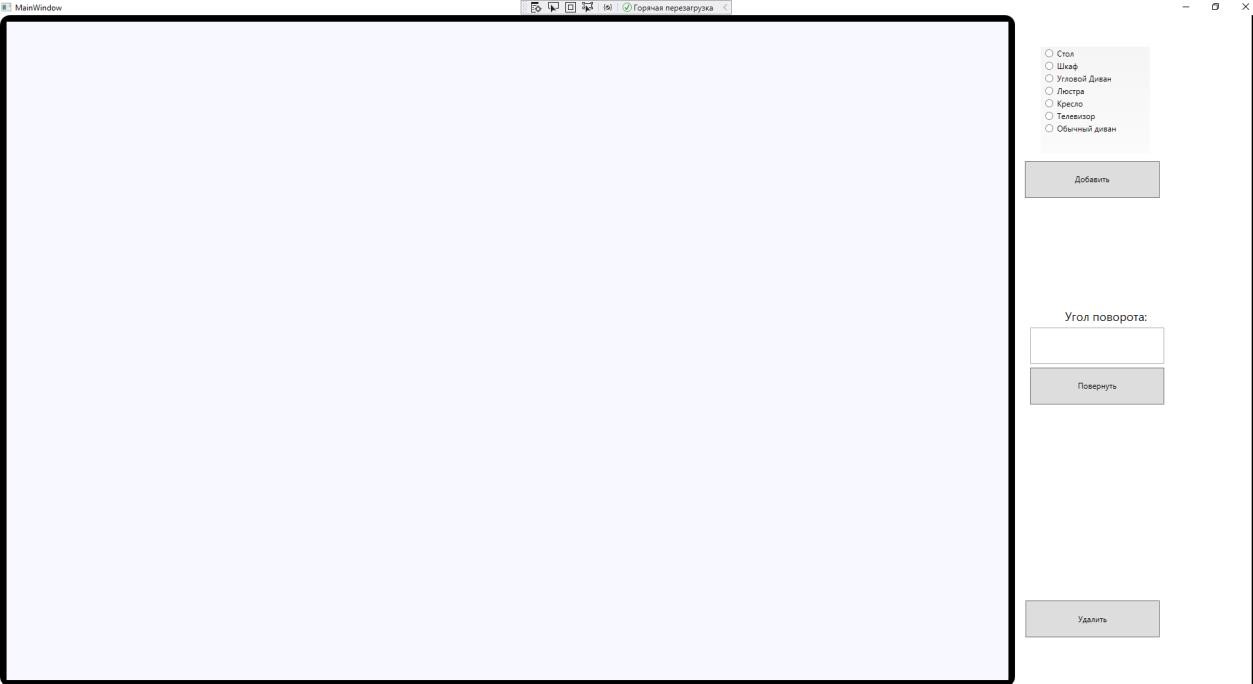
</Grid>

</Grid>

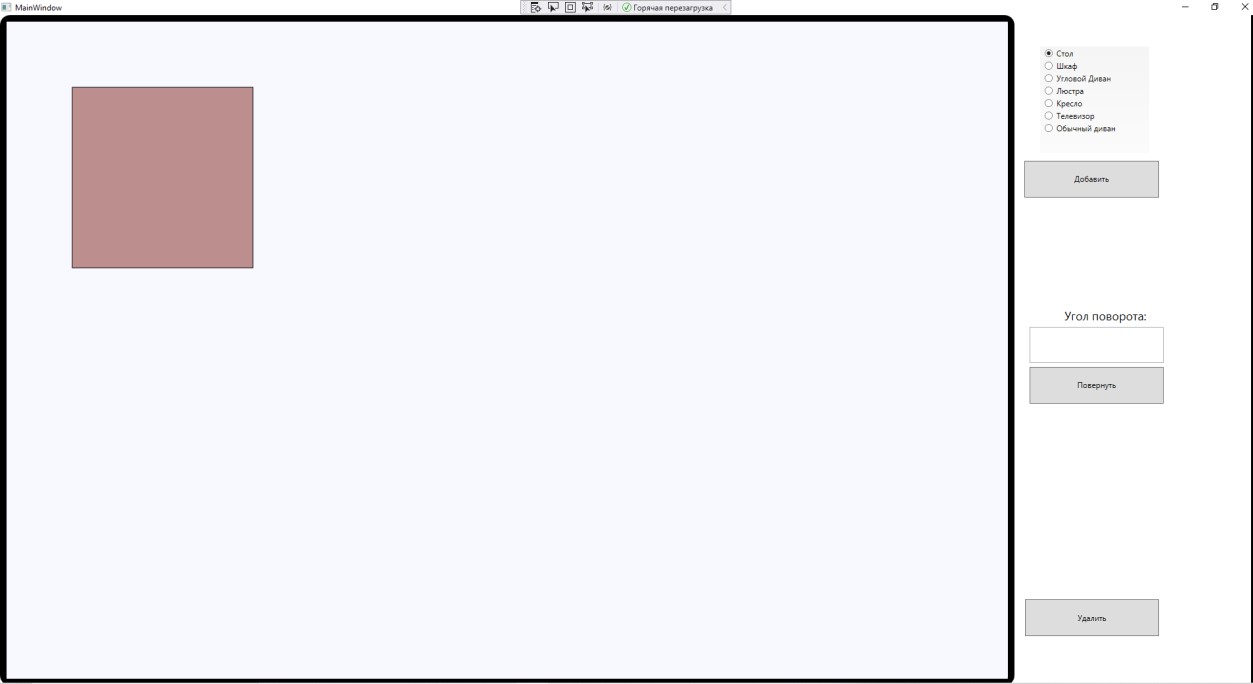
</Window>

# Работа программы:

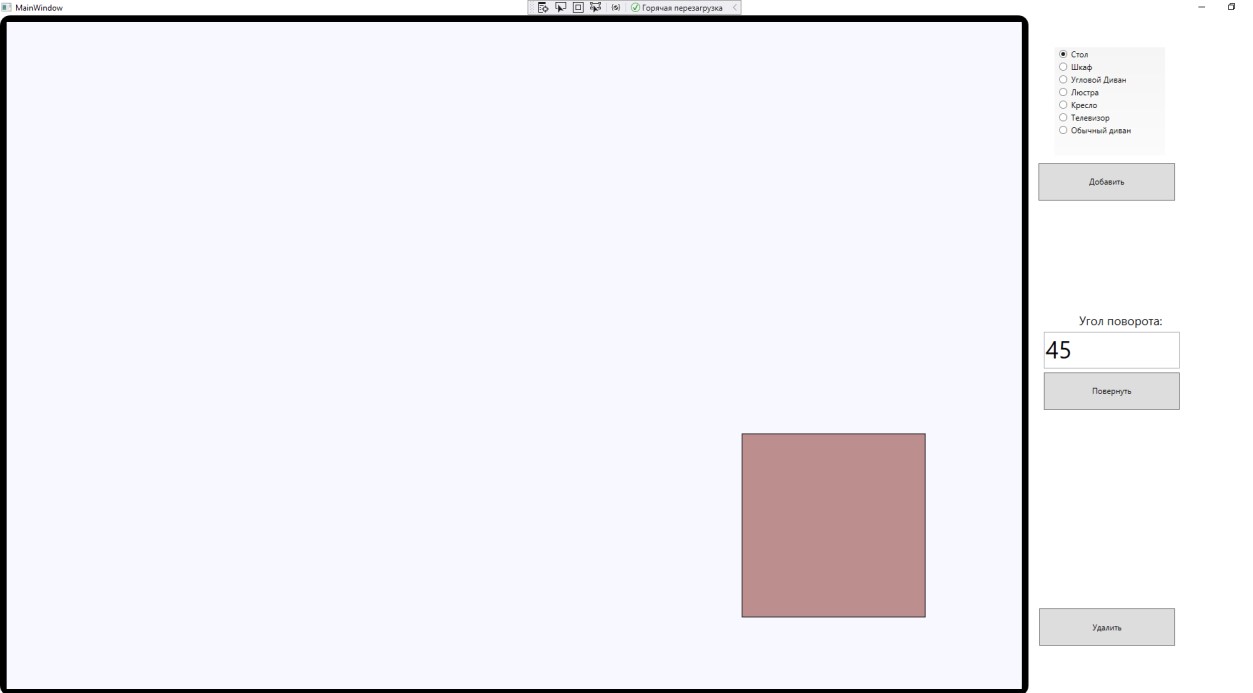
1. Окно приложения



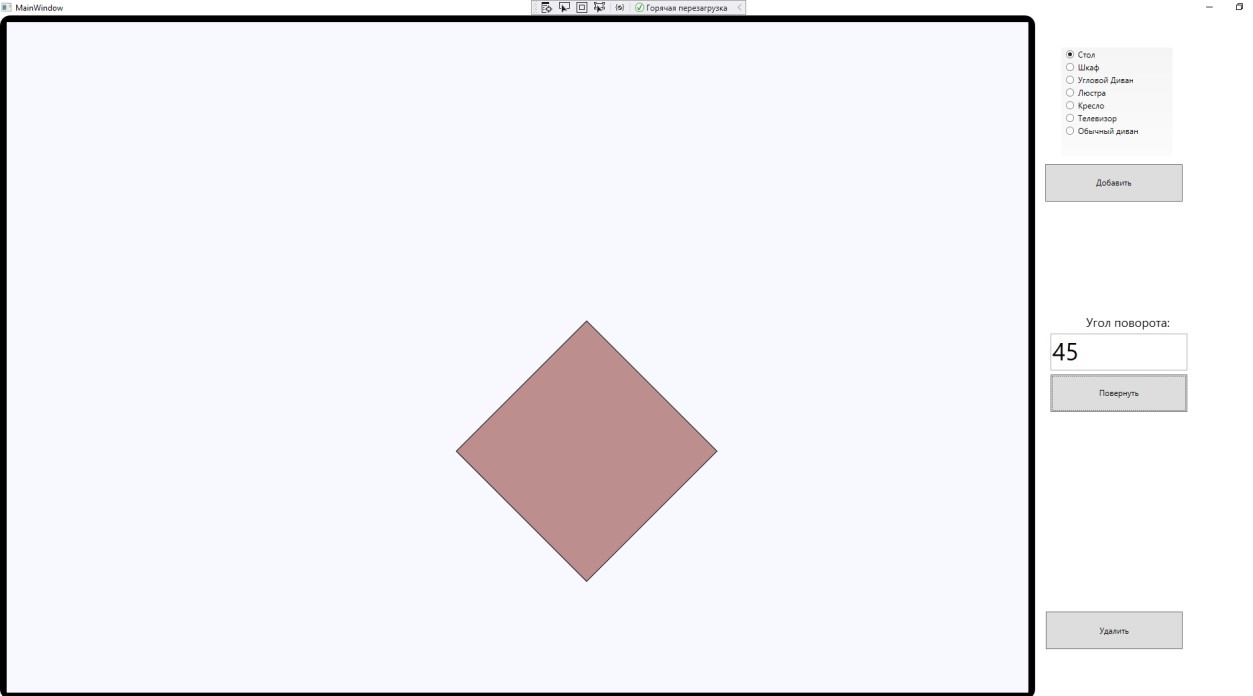
## 2. Добавление стола(элемента)



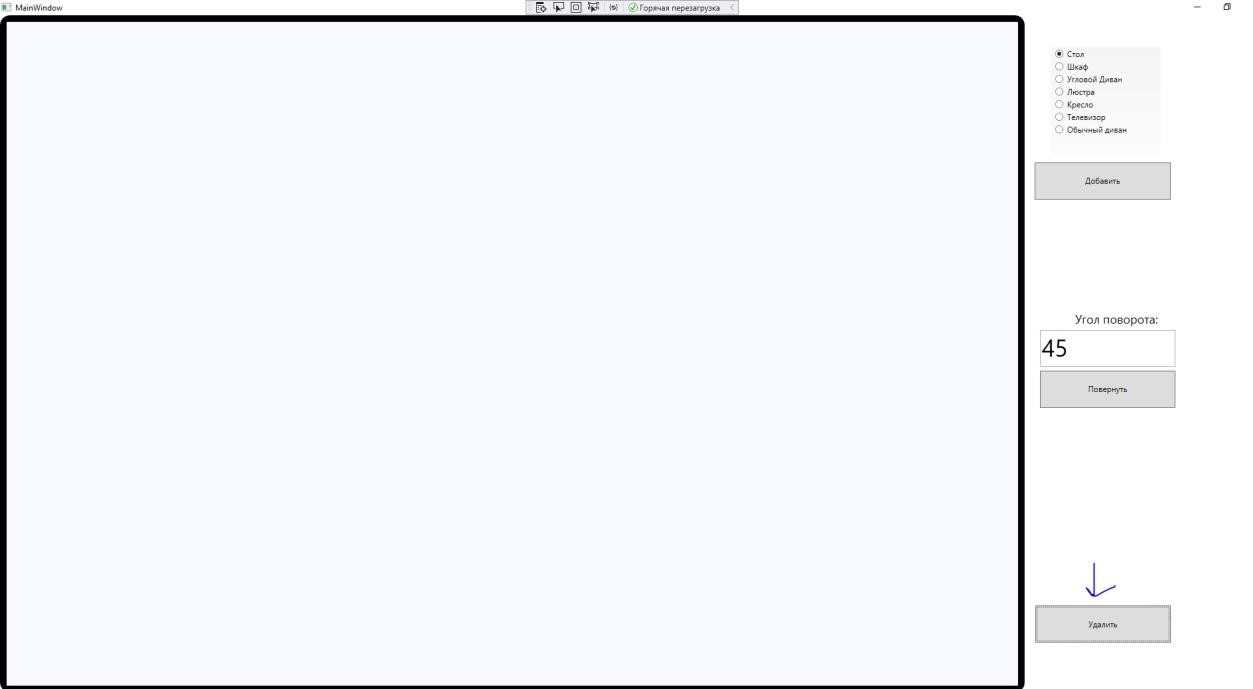
## 3. Движение стола

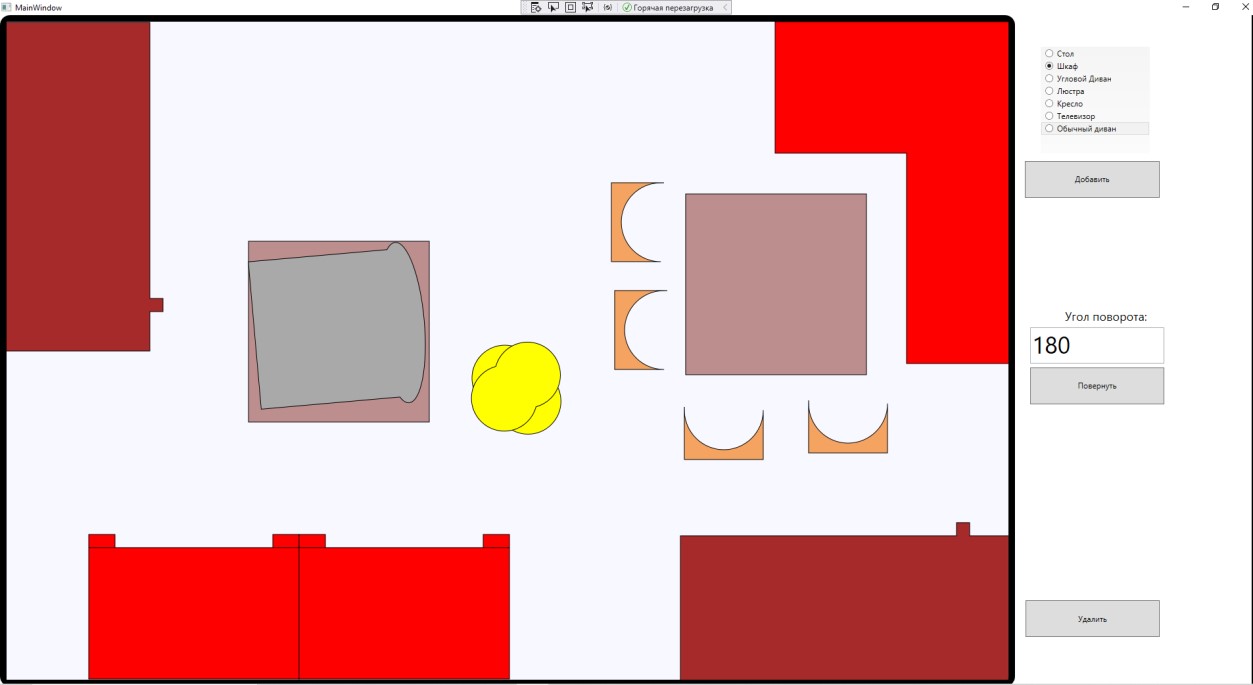


1. Поворот стола



1. Удаление стола



1. Полный пример работы программы
2. 

Контрольные вопросы:

1. Зачем нужен перегруженный оператор присваивания?

Перегруженная операция присваивания используется при присваивании одного объекта другому существующему объекту.

1. Зачем нужен механизм наследования?

Основное назначение механизма наследования — повторное использование кодов, так как большинство используемых типов данных являются вариантами друг друга, и писать для каждого свой класс нецелесообразно. 3. Зачем используются модификаторы при наследовании классов, какие это модификаторы?

Уровни доступа полей этого класса.

Модификаторы: Private, public, protected.

4. Зачем нужен механизм полиморфизма?

Полиморфизм - это свойство, которое позволяет одно и то же имя использовать для решения двух или более схожих, но технически разных задач. Целью полиморфизма, применительно к объектноориентированному программированию, является использование одного имени для задания общих для класса действий.

5.Что понимается под динамическим полиморфизмом?

Динамический (во время выполнения) полиморфизм -это полиморфизм, существующий во время выполнения.

1. Что такое интерфейс класса?

Доступ к полям. Например, наследуя класс мы получаем доступ к его интерфейсу, но реализация происходит в исходном классе.

1. Что такое чисто виртуальный метод? Как он выглядит?

Чисто виртуальная функция (pure virtual function) является функцией, которая объявляется в базовом классе, но не имеет в нем определения. Поскольку она не имеет определения, то есть тела в этом базовом классе, то всякий производный класс обязан иметь свою собственную версию определения.

Выглядит : virtual тип имя\_функции(список параметров) = 0;

1. Какой класс называется абстрактным?

Абстрактный класс в объектно-ориентированном программировании — базовый класс, который не предполагает создания экземпляров. Абстрактные классы реализуют на практике один из принципов ООП — полиморфизм.

# Вывод

Я изучила и закрепила механизмы работы ООП.