1. **Работа с графикой. Трехмерная графика**

Задание 1. Изучить и выполнить примеры Лабораторной работы №10 по книге Программирование на С# Демин, Дорофеев. Выполните задания.

1.1 Нарисуйте 5 различных фигур треугольник, эллипс, закрашенный круг, закрашенный прямоугольник, сектор.

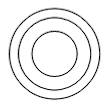


Рисунок 1.1 – Задание 1.2

Источник: собственная разработка

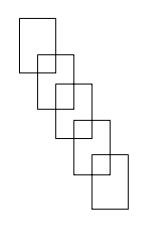


Рисунок 1.2 – Задание 1.3

Источник: собственная разработка

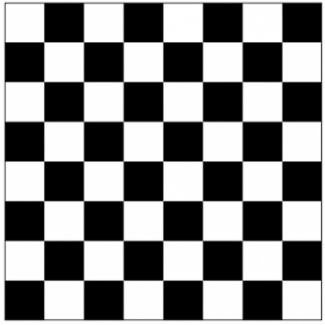


Рисунок 1.3 – Задание 1.4

Источник: собственная разработка

Листинг программы:

using System;

using System.Windows.Forms;

namespace Task\_1

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

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

{

Form2 form2 = new Form2();

form2.RD\_CheckerOne = radioButton1.Checked;

form2.RD\_CheckerTwo = radioButton2.Checked;

form2.RD\_CheckerThree = radioButton3.Checked;

form2.RD\_CheckerFore = radioButton4.Checked;

form2.RD\_CheckerFive = radioButton5.Checked;

form2.ShowDialog();

}

}

}

Таблица 1.1 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
|  |  |

Анализ результатов:

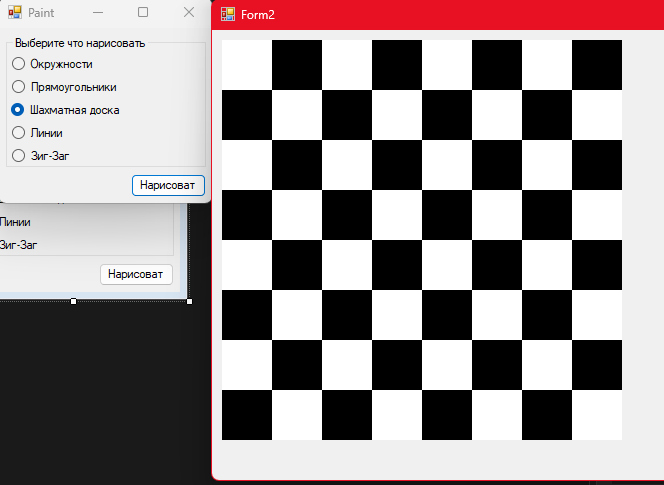


Рисунок 1.4 – Результат работы программы

Источник: собственная разработка

Задание 2. Выполните задание 1 в приложении WPF используя объект DrawingContext.

Листинг программы:

<Window x:Class="Task\_2.MainWindow"

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:Task\_2.Models"

mc:Ignorable="d"

Title="MainWindow" Height="450" Width="900">

<Grid>

<!--<local:TestDrawing Height="50" Width="50"></local:TestDrawing>-->

<local:Cercle></local:Cercle>

<local:Rectangles></local:Rectangles>

<local:ChesDask></local:ChesDask>

</Grid>

</Window>

Таблица 2.1 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
|  |  |

Анализ результатов:

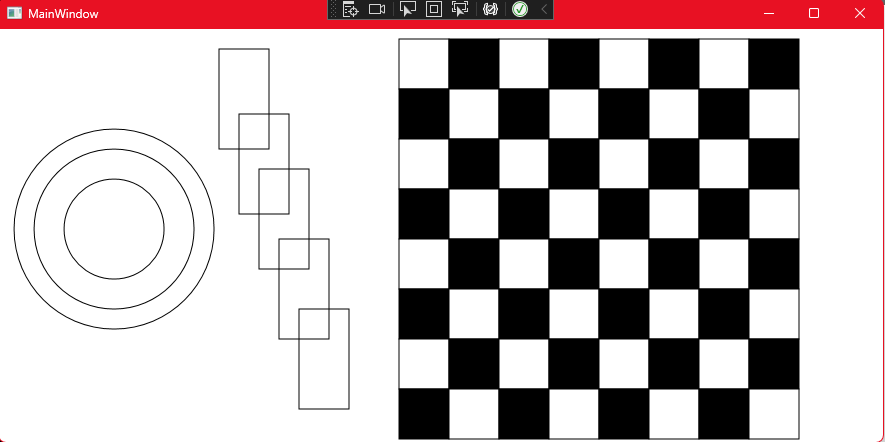


Рисунок 2.1 – Результат работы программы

Источник: собственная разработка

Задание 3. Нарисовать 3 кнопки: Треугольную, круглую, в форме пирамиды.

Листинг программы:

<Window x:Class="Task\_3.MainWindow"

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:Task\_3"

mc:Ignorable="d"

Title="MainWindow" Height="450" Width="800">

<Window.Resources>

<StreamGeometry x:Key="Geometry">M12,24 L36,0 L36,48 Z</StreamGeometry>

<Style x:Key="FocusVisual">

<Setter Property="Control.Template">

<Setter.Value>

<ControlTemplate>

<Path Stroke="{DynamicResource {x:Static SystemColors.ControlTextBrushKey}}" StrokeThickness="1" StrokeDashArray="1 2" Data="{StaticResource Geometry}" Width="48"/>

</ControlTemplate>

</Setter.Value>

</Setter>

</Style>

<SolidColorBrush x:Key="Button.Static.Background" Color="#FFDDDDDD"/>

<SolidColorBrush x:Key="Button.Static.Border" Color="#FF707070"/>

<SolidColorBrush x:Key="Button.MouseOver.Background" Color="#FFBEE6FD"/>

<SolidColorBrush x:Key="Button.MouseOver.Border" Color="#FF3C7FB1"/>

<SolidColorBrush x:Key="Button.Pressed.Background" Color="#FFC4E5F6"/>

<SolidColorBrush x:Key="Button.Pressed.Border" Color="#FF2C628B"/>

<SolidColorBrush x:Key="Button.Disabled.Background" Color="#FFF4F4F4"/>

<SolidColorBrush x:Key="Button.Disabled.Border" Color="#FFADB2B5"/>

<Style x:Key="TriangleButtonStyle" TargetType="{x:Type Button}" >

<Setter Property="FocusVisualStyle" Value="{StaticResource FocusVisual}"/>

<Setter Property="Background" Value="{StaticResource Button.Static.Background}"/>

<Setter Property="BorderBrush" Value="{StaticResource Button.Static.Border}"/>

<Setter Property="Template">

<Setter.Value>

<ControlTemplate TargetType="{x:Type Button}">

<Path Name="border" Stroke="{TemplateBinding BorderBrush}" Fill="{TemplateBinding Background}" Data="{StaticResource Geometry}" Width="48"/>

<ControlTemplate.Triggers>

<Trigger Property="IsMouseOver" Value="true">

<Setter Property="Fill" TargetName="border" Value="{StaticResource Button.MouseOver.Background}"/>

<Setter Property="Stroke" TargetName="border" Value="{StaticResource Button.MouseOver.Border}"/>

</Trigger>

</ControlTemplate.Triggers>

</ControlTemplate>

</Setter.Value>

</Setter>

</Style>

<RotateTransform x:Key="Rotate090" CenterX="24" CenterY="24" Angle="000"/>

<RotateTransform x:Key="Rotate190" CenterX="24" CenterY="24" Angle="090"/>

</Window.Resources>

<StackPanel Orientation="Horizontal" VerticalAlignment="Center" HorizontalAlignment="Center" Width="341">

<Button Style="{DynamicResource TriangleButtonStyle}" RenderTransform="{StaticResource Rotate090}"/>

<Button Width="50" Height="50" Content="OK">

<Button.Template>

<ControlTemplate TargetType="Button">

<Grid>

<Ellipse Fill="DarkOrange"></Ellipse>

<Label Content="{TemplateBinding Content}" HorizontalAlignment="Center"

VerticalAlignment="Center"></Label>

</Grid>

</ControlTemplate>

</Button.Template>

</Button>

<Button Style="{DynamicResource TriangleButtonStyle}" RenderTransform="{StaticResource Rotate190}"/>

</StackPanel>

</Window>

Таблица 3.1 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
|  |  |

Анализ результатов:

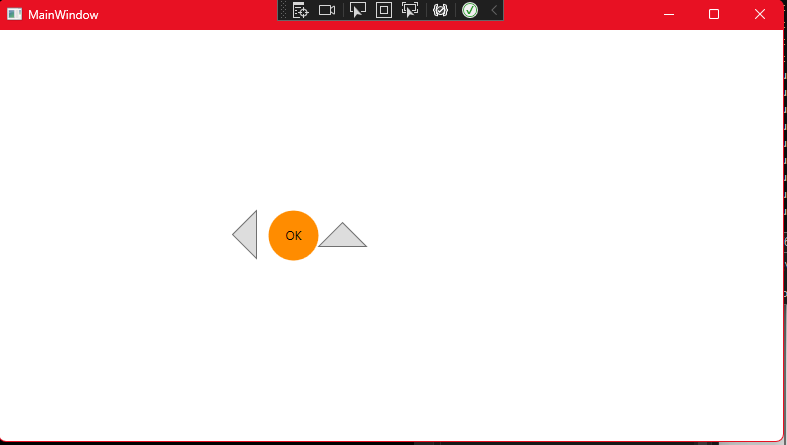


Рисунок 3.1 – Результат работы программы

Источник: собственная разработка