НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО

Факультет программной инженерии и компьютерной техники

Прикладное программирование

Лабораторная работа №1. "Разработка простого графического приложения WPF на языке C#"

Вариант: 1

Выполнил студент

Вайкус Яков Михайлович

Группа № P3124

Преподаватель: Корж

г. Санкт-Петербург

2022

Оглавление

[**Задание:** 3](#_Toc114565067)

[**Отчет:** 4](#_Toc114565068)

[**Исходный код:** 4](#_Toc114565069)

[**Результат:** 5](#_Toc114565070)

[**Вывод:** 5](#_Toc114565071)

[**Список литературы** 5](#_Toc114565072)

**Задание:**

Разработать простое графическое приложение (WinForms или WPF) на языке C#, в котором должны использоваться основные элементы интерфейса (Label, TextBox, Button, CheckBox, RadioButton, ComboBox, ListBox и др.). Приложение должно демонстрировать умение работать со свойствами элементов, а также реагировать на действия пользователя (обработку событий).

Вариант 1. «Заказ в кафе»

TextBox для ввода имени/ника клиента.

ComboBox для выбора основного блюда (пицца, паста, бургер, салат и т.д.).

CheckBox для добавления дополнительных ингредиентов (сыр, соус, перец и т.д.).

RadioButton для способа получения заказа («Взять с собой» или «Доставка»).

Button «Оформить заказ», по нажатию которого формируется строка-описание и добавляется в ListBox.

Label для вывода предупреждений (если имя не введено) и подтверждений.

**Отчет:**

# **Исходный код формы:**

<Window x:Class="CafeOrderApp.MainWindow"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

Title="Заказ в кафе" MinHeight="640" MinWidth="907" Height="400" Width="600"

Icon="icon.ico"

Background="{DynamicResource BackgroundBrush}">

<Grid Name="myGrid" Margin="10">

<Grid.ColumnDefinitions>

<ColumnDefinition Width="260" /> <!-- Левая панель с формами -->

<ColumnDefinition Width="\*"/> <!-- Список заказов -->

</Grid.ColumnDefinitions>

<StackPanel Grid.Column="0">

<!-- Имя клиента -->

<Border BorderBrush="Gray" BorderThickness="1" CornerRadius="5" Padding="10" Margin="10,10,0,10">

<StackPanel>

<Label Content="Имя клиента" HorizontalAlignment="Center" Name ="NameL"/>

<TextBox x:Name="NameTextBox" Height="32" HorizontalAlignment="Stretch"/>

</StackPanel>

</Border>

<!-- Выбор блюда -->

<Border BorderBrush="Gray" BorderThickness="1" CornerRadius="5" Padding="10" Margin="10,0,0,10" >

<StackPanel>

<Label Content="Выберите основное блюдо"/>

<ComboBox x:Name="DishComboBox" Height="32" HorizontalAlignment="Stretch">

<ComboBox.Items>

<sys:String xmlns:sys="clr-namespace:System;assembly=mscorlib">Пицца</sys:String>

<sys:String xmlns:sys="clr-namespace:System;assembly=mscorlib">Паста</sys:String>

<sys:String xmlns:sys="clr-namespace:System;assembly=mscorlib">Бургер</sys:String>

<sys:String xmlns:sys="clr-namespace:System;assembly=mscorlib">Салат</sys:String>

</ComboBox.Items>

</ComboBox>

</StackPanel>

</Border>

<!-- Дополнительные ингредиенты -->

<Border BorderBrush="Gray" BorderThickness="1" CornerRadius="5" Padding="10" Margin="10,0,0,10">

<StackPanel >

<Label Content="Дополнительные ингредиенты"/>

<CheckBox x:Name="CheeseCheckBox" Content="Сыр" HorizontalAlignment="Stretch"/>

<CheckBox x:Name="SauceCheckBox" Content="Соус" HorizontalAlignment="Stretch"/>

<CheckBox x:Name="PepperCheckBox" Content="Перец" HorizontalAlignment="Stretch"/>

</StackPanel>

</Border>

<!-- Способ получения -->

<Border BorderBrush="Gray" BorderThickness="1" CornerRadius="5" Padding="10" Margin="10,0,0,10">

<StackPanel >

<Label Content="Способ получения заказа"/>

<RadioButton x:Name="TakeawayRadioButton" Content="Взять с собой" GroupName="DeliveryMethod" HorizontalAlignment="Stretch"/>

<RadioButton x:Name="DeliveryRadioButton" IsChecked="True" Content="Доставка" GroupName="DeliveryMethod" HorizontalAlignment="Stretch"/>

</StackPanel>

</Border>

<!-- Кнопка "Заказать" -->

<Button Content="Заказать" Height="35" HorizontalAlignment="Stretch"

Click="PlaceOrderButton\_Click" Margin="10,0,0,10"/>

<!-- Кнопка переключения темы -->

<Button Content="Переключить тему" Height="35" Margin="10,0,0,10" HorizontalAlignment="Stretch"

Click="ToggleTheme\_Click"/>

</StackPanel>

<!-- Вторая колонка (список заказов) -->

<Border Grid.Column="1" BorderBrush="Gray" BorderThickness="1" CornerRadius="5" Margin="10,10,20,0">

<StackPanel Grid.Row="0">

<Label Content="Список заказов:" Margin="20,0,0,10" FontWeight="Bold"/>

<ListBox x:Name="OrdersListBox" VerticalAlignment="Stretch"

HorizontalAlignment="Stretch" Visibility="Collapsed"/>

<!-- Сообщение об ошибке внизу колонки -->

<Label x:Name="StatusLabel" Foreground="Red"

HorizontalAlignment="Center"

Visibility="Collapsed"

Margin="10,40,0,0"/>

<!-- Кнопка "Очистить список" -->

<Button Content="Очистить список" Height="40" Width="180"

HorizontalAlignment="Center" Margin="10,10,0,0"

Click="ClearOrdersButton\_Click"/>

</StackPanel>

</Border>

</Grid>

</Window>

**Исходный код Функций:**

using System.Windows;

using System.Windows.Media.Animation;

using System.Windows.Media.Imaging;

using System.Windows.Media;

namespace CafeOrderApp

{

    public partial class MainWindow : Window

    {

        public MainWindow()

        {

            InitializeComponent();

        }

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

        {

            OrdersListBox.Items.Clear();

             if (OrdersListBox.Visibility == Visibility.Visible){

            OrdersListBox.Visibility = Visibility.Collapsed;}

        }

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

        {

            // Проверка на ввод имени

            string name = NameTextBox.Text.Trim();

            if (string.IsNullOrEmpty(name))

            {

                StatusLabel.Content = "Пожалуйста, введите имя!";

                StatusLabel.Foreground = System.Windows.Media.Brushes.Red;

                StatusLabel.Visibility = Visibility.Visible;

                StartFadeOutAnimation();

                return;

            }

            // Проверка выбора блюда

            if (DishComboBox.SelectedItem == null)

            {

                StatusLabel.Content = "Пожалуйста, выберите основное блюдо!";

                StatusLabel.Foreground = System.Windows.Media.Brushes.Red;

                StatusLabel.Visibility = Visibility.Visible;

                StartFadeOutAnimation();

                return;

            }

            string dish = DishComboBox.SelectedItem.ToString();

            string extras = "";

            if (CheeseCheckBox.IsChecked == true) extras += "Сыр, ";

            if (SauceCheckBox.IsChecked == true) extras += "Соус, ";

            if (PepperCheckBox.IsChecked == true) extras += "Перец, ";

            extras = extras.TrimEnd(',', ' ');

            string orderExtras = string.IsNullOrEmpty(extras) ? "" : $"({extras})";

            string method = TakeawayRadioButton.IsChecked == true ? "Взять с собой" : "Доставка";

            string order = $"{name} заказал {dish} {orderExtras} [{method}]";

            // Добавляем заказ в список

            OrdersListBox.Items.Add(order);

             if (OrdersListBox.Visibility == Visibility.Collapsed){

            OrdersListBox.Visibility = Visibility.Visible;}

            // Сбрасываем статус

            StatusLabel.Content = "Заказ оформлен!";

            StatusLabel.Foreground = System.Windows.Media.Brushes.Green;

            StatusLabel.Visibility = Visibility.Visible;

            // Запуск анимации для плавного исчезновения

            StartFadeOutAnimation();

        }

        private void StartFadeOutAnimation()

        {

            // Создаем анимацию для исчезновения

            DoubleAnimation fadeOutAnimation = new DoubleAnimation

            {

                From = 1.0,

                To = 0.0,

                Duration = new Duration(TimeSpan.FromSeconds(2))

            };

            // Применяем анимацию к свойству Opacity

            StatusLabel.BeginAnimation(OpacityProperty, fadeOutAnimation);

        }

        private bool isDarkTheme = false;

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

{

    var theme = isDarkTheme ? "Styles.xaml" : "DarkTheme.xaml";

    var dict = new ResourceDictionary { Source = new Uri(theme, UriKind.Relative) };

    Application.Current.Resources.MergedDictionaries.Clear();

    Application.Current.Resources.MergedDictionaries.Add(dict);

    isDarkTheme = !isDarkTheme;

}

}

    }

**Исходный код Стилей:**

<ResourceDictionary xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:sys="clr-namespace:System;assembly=mscorlib">

<ImageBrush x:Key="BackgroundBrush" ImageSource="light-theme.png" Stretch ="UniformToFill" Opacity="0.5"/>

<Style TargetType="Border">

<Setter Property=" Background" Value="#ffedd4"/>

</Style>

<Style TargetType="Button">

<Setter Property="Background" Value="#9C6E58"/>

<Setter Property="Foreground" Value="#f2f2f2"/>

<Setter Property="FontSize" Value="14"/>

<Setter Property="Padding" Value="10,5"/>

<Setter Property="BorderThickness" Value="0"/>

<Setter Property="FontWeight" Value="Bold"/>

<Setter Property="FontFamily" Value="Open Sans, Helvetica Neue, Arial, sans-serif"/>

<Setter Property="Template">

<Setter.Value>

<ControlTemplate TargetType="Button">

<Border x:Name="Border"

Background="{TemplateBinding Background}"

CornerRadius="10"

BorderThickness="1"

BorderBrush="Gray">

<ContentPresenter HorizontalAlignment="Center"

VerticalAlignment="Center"/>

</Border>

<ControlTemplate.Triggers>

<!-- Триггер для изменения фона при наведении -->

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

<Setter TargetName="Border" Property="Background" Value="#4322"/>

</Trigger>

</ControlTemplate.Triggers>

</ControlTemplate>

</Setter.Value>

</Setter>

</Style>

<!-- Стиль для меток -->

<Style TargetType="Label">

<Setter Property="FontSize" Value="14"/>

<Setter Property="Margin" Value="5"/>

<Setter Property="Foreground" Value="#6e4d2c"/>

<Setter Property="FontWeight" Value="Bold"/>

<Setter Property="FontFamily" Value="Open Sans, Helvetica Neue, Arial, sans-serif"/>

<Setter Property="HorizontalAlignment" Value="Center"/>

</Style>

<Style TargetType="TextBox">

<Setter Property="Template">

<Setter.Value>

<ControlTemplate TargetType="TextBox">

<Border Background="{TemplateBinding Background}"

BorderBrush="{TemplateBinding BorderBrush}"

BorderThickness="{TemplateBinding BorderThickness}"

CornerRadius="5">

<ScrollViewer x:Name="PART\_ContentHost" />

</Border>

</ControlTemplate>

</Setter.Value>

</Setter>

<Setter Property="Background" Value="White"/>

<Setter Property="Foreground" Value="Black"/>

<Setter Property="Padding" Value="5"/>

<Setter Property="BorderBrush" Value="Gray"/>

<Setter Property="BorderThickness" Value="1"/>

<Setter Property="FontSize" Value="14"/>

</Style>

<!-- Стиль для ComboBox -->

<Style TargetType="ComboBox">

<Setter Property="Foreground" Value="White"/>

<Setter Property="FontSize" Value="15"/>

<Setter Property="FontWeight" Value="Bold"/>

<Setter Property="FontFamily" Value="Open Sans, Helvetica Neue, Arial, sans-serif"/>

<Setter Property="Padding" Value="5"/>

<Setter Property="BorderBrush" Value="#A18A7F"/>

<Setter Property="Background" Value="#ffc361"/>

<Setter Property="BorderThickness" Value="1"/>

<Setter Property="HorizontalAlignment" Value="Stretch"/>

<Setter Property="Template">

<Setter.Value>

<ControlTemplate TargetType="ComboBox">

<Grid>

<!-- Основной блок -->

<Border Name="Border"

Background="#9C6E58"

BorderBrush="#A18A7F"

BorderThickness="{TemplateBinding BorderThickness}"

CornerRadius="5">

<Grid>

<Grid.ColumnDefinitions>

<!-- Основная область для содержимого -->

<ColumnDefinition Width="\*"/>

<!-- Столбец для стрелки -->

<ColumnDefinition Width="30"/>

</Grid.ColumnDefinitions>

<!-- Текстовое поле для отображения выбранного элемента -->

<ContentPresenter Name="ContentSite"

Grid.Column="0"

Content="{TemplateBinding SelectionBoxItem}"

ContentTemplate="{TemplateBinding SelectionBoxItemTemplate}"

VerticalAlignment="Center"

HorizontalAlignment="Left"

Margin="10,3,30,3"/>

<!-- Кнопка-стрелка -->

<ToggleButton Grid.Column="1"

Name="ToggleButton"

ClickMode="Press"

Background="Transparent"

BorderThickness="0"

IsChecked="{Binding IsDropDownOpen, RelativeSource={RelativeSource TemplatedParent}}">

<Path Data="M 0 0 L 4 4 L 8 0 Z"

Fill="White"

Width="10"

Height="6"

HorizontalAlignment="Center"

VerticalAlignment="Center"/>

</ToggleButton>

</Grid>

</Border>

<!-- Всплывающее окно -->

<Popup Name="Popup"

Placement="Bottom"

IsOpen="{TemplateBinding IsDropDownOpen}"

AllowsTransparency="True"

Focusable="False"

StaysOpen="False">

<Border Background="#7D5B45"

BorderBrush="#C3A89B"

BorderThickness="1"

CornerRadius="10"

Width="{TemplateBinding ActualWidth}">

<ScrollViewer Margin="5">

<StackPanel IsItemsHost="True"/>

</ScrollViewer>

</Border>

</Popup>

</Grid>

<ControlTemplate.Triggers>

<!-- Изменение цвета при наведении -->

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

<Setter TargetName="Border" Property="Background" Value="#83695C"/>

<Setter TargetName="Border" Property="BorderBrush" Value="#C3A89B"/>

</Trigger>

<!-- Изменение цвета при открытии -->

<Trigger Property="IsDropDownOpen" Value="True">

<Setter TargetName="Border" Property="Background" Value="#7D5B45"/>

<Setter TargetName="Border" Property="BorderBrush" Value="#C3A89B"/>

</Trigger>

</ControlTemplate.Triggers>

</ControlTemplate>

</Setter.Value>

</Setter>

</Style>

<!-- Стиль для элементов списка -->

<Style TargetType="ComboBoxItem">

<Setter Property="Foreground" Value="White"/>

<Setter Property="Background" Value="Transparent"/>

<Setter Property="Padding" Value="5,2"/>

<Setter Property="Template">

<Setter.Value>

<ControlTemplate TargetType="ComboBoxItem">

<Border Name="Border"

Background="{TemplateBinding Background}"

CornerRadius="25"

Padding="{TemplateBinding Padding}">

<ContentPresenter VerticalAlignment="Center" HorizontalAlignment="Left"/>

</Border>

<ControlTemplate.Triggers>

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

<Setter TargetName="Border" Property="Background" Value="#83695C"/>

</Trigger>

<Trigger Property="IsSelected" Value="True">

<Setter TargetName="Border" Property="Background" Value="#9D7A6A"/>

</Trigger>

</ControlTemplate.Triggers>

</ControlTemplate>

</Setter.Value>

</Setter>

</Style>

<!-- Стиль для ListBox -->

<Style TargetType="ListBox">

<Setter Property="FontSize" Value="14"/>

<Setter Property="Margin" Value="5"/>

<Setter Property="BorderBrush" Value="#FF9A755A"/>

<Setter Property="BorderThickness" Value="2"/>

<Setter Property="Background" Value="White"/>

</Style>

<!-- Стиль для CheckBox -->

<Style TargetType="CheckBox">

<Setter Property="FontSize" Value="15"/>

<Setter Property="Foreground" Value="#6e4d2c"/>

<Setter Property="FontWeight" Value="Bold"/>

<Setter Property="FontFamily" Value="Open Sans, Helvetica Neue, Arial, sans-serif"/>

<Setter Property="Margin" Value="5"/>

<Setter Property="Template">

<Setter.Value>

<ControlTemplate TargetType="CheckBox">

<StackPanel Orientation="Horizontal">

<!-- Кастомная кнопка -->

<Border Name="CheckBorder"

Width="18" Height="18"

CornerRadius="5"

BorderBrush="#C3A89B"

BorderThickness="2"

Background="Transparent"

Margin="5,0,10,0">

<Path Name="CheckMark"

Data="M2,9 L6,13 L14,4"

Stroke="Transparent"

StrokeThickness="2"/>

</Border>

<!-- Текст -->

<ContentPresenter VerticalAlignment="Center"

HorizontalAlignment="Left"

Content="{TemplateBinding Content}"

TextElement.Foreground="{TemplateBinding Foreground}"/>

</StackPanel>

<ControlTemplate.Triggers>

<!-- Цвет при наведении -->

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

<Setter TargetName="CheckBorder" Property="BorderBrush" Value="#C3A89B"/>

<Setter TargetName="CheckMark" Property="Stroke" Value="#C3A89B"/>

</Trigger>

<!-- Выбранное состояние -->

<Trigger Property="IsChecked" Value="True">

<Setter TargetName="CheckMark" Property="Stroke" Value="#C3A89B"/>

<Setter TargetName="CheckBorder" Property="BorderBrush" Value="#D1B6A9"/>

</Trigger>

<!-- Disabled -->

<Trigger Property="IsEnabled" Value="False">

<Setter TargetName="CheckBorder" Property="Opacity" Value="0.5"/>

<Setter Property="Foreground" Value="#7E6358"/>

</Trigger>

</ControlTemplate.Triggers>

</ControlTemplate>

</Setter.Value>

</Setter>

</Style>

<Style TargetType="RadioButton">

<Setter Property="FontSize" Value="15"/>

<Setter Property="Foreground" Value="#6e4d2c"/>

<Setter Property="FontWeight" Value="Bold"/>

<Setter Property="FontFamily" Value="Open Sans, Helvetica Neue, Arial, sans-serif"/>

<Setter Property="Margin" Value="5"/>

<Setter Property="Template">

<Setter.Value>

<ControlTemplate TargetType="RadioButton">

<StackPanel Orientation="Horizontal">

<!-- Кастомная кнопка -->

<Border Name="RadioBorder"

Width="18" Height="18"

CornerRadius="9"

BorderBrush="#C3A89B"

BorderThickness="2"

Background="Transparent"

Margin="5,0,10,0">

<Ellipse Name="RadioDot"

Width="10" Height="10"

Fill="Transparent"/>

</Border>

<!-- Текст -->

<ContentPresenter VerticalAlignment="Center"

HorizontalAlignment="Left"

Content="{TemplateBinding Content}"

TextElement.Foreground="{TemplateBinding Foreground}"/>

</StackPanel>

<ControlTemplate.Triggers>

<!-- Цвет при наведении -->

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

<Setter TargetName="RadioBorder" Property="BorderBrush" Value="#C3A89B"/>

<Setter TargetName="RadioDot" Property="Fill" Value="#C3A89B"/>

</Trigger>

<!-- Выбранное состояние -->

<Trigger Property="IsChecked" Value="True">

<Setter TargetName="RadioDot" Property="Fill" Value="#C3A89B"/>

<Setter TargetName="RadioBorder" Property="BorderBrush" Value="#D1B6A9"/>

</Trigger>

<!-- Disabled -->

<Trigger Property="IsEnabled" Value="False">

<Setter TargetName="RadioBorder" Property="Opacity" Value="0.5"/>

<Setter Property="Foreground" Value="#7E6358"/>

</Trigger>

</ControlTemplate.Triggers>

</ControlTemplate>

</Setter.Value>

</Setter>

</Style>

</ResourceDictionary>

Темная тема:  
  
<ResourceDictionary xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

                    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

                    xmlns:sys="clr-namespace:System;assembly=mscorlib">

    <!-- Фон и изображения для темной темы -->

    <ImageBrush x:Key="BackgroundBrush" ImageSource="dark-theme.png" Stretch="UniformToFill" Opacity="0.5"/>

    <!-- Стиль для Border -->

    <Style TargetType="Border">

        <Setter Property="Background" Value="#1f1f1f"/>

    </Style>

    <!-- Стиль для Button -->

    <Style TargetType="Button">

        <Setter Property="Background" Value="#3A3A3A"/>

        <Setter Property="Foreground" Value="#EAEAEA"/>

        <Setter Property="FontSize" Value="14"/>

        <Setter Property="Padding" Value="10,5"/>

        <Setter Property="BorderThickness" Value="0"/>

        <Setter Property="FontWeight" Value="Bold"/>

        <Setter Property="FontFamily" Value="Open Sans, Helvetica Neue, Arial, sans-serif"/>

        <Setter Property="Template">

            <Setter.Value>

                <ControlTemplate TargetType="Button">

                    <Border x:Name="Border"

                            Background="{TemplateBinding Background}"

                            CornerRadius="10"

                            BorderThickness="1"

                            BorderBrush="#444444">

                        <ContentPresenter HorizontalAlignment="Center"

                                          VerticalAlignment="Center"/>

                    </Border>

                    <ControlTemplate.Triggers>

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

                            <Setter TargetName="Border" Property="Background" Value="#5C5C5C"/>

                        </Trigger>

                    </ControlTemplate.Triggers>

                </ControlTemplate>

            </Setter.Value>

        </Setter>

    </Style>

    <!-- Стиль для Label -->

    <Style TargetType="Label">

        <Setter Property="FontSize" Value="14"/>

        <Setter Property="Margin" Value="5"/>

        <Setter Property="Foreground" Value="#EAEAEA"/>

        <Setter Property="FontWeight" Value="Bold"/>

        <Setter Property="FontFamily" Value="Open Sans, Helvetica Neue, Arial, sans-serif"/>

        <Setter Property="HorizontalAlignment" Value="Center"/>

    </Style>

    <!-- Стиль для TextBox -->

    <Style TargetType="TextBox">

        <Setter Property="Template">

            <Setter.Value>

                <ControlTemplate TargetType="TextBox">

                    <Border Background="{TemplateBinding Background}"

                            BorderBrush="{TemplateBinding BorderBrush}"

                            BorderThickness="{TemplateBinding BorderThickness}"

                            CornerRadius="5">

                        <ScrollViewer x:Name="PART\_ContentHost" />

                    </Border>

                </ControlTemplate>

            </Setter.Value>

        </Setter>

        <Setter Property="Background" Value="#2E2E2E"/>

        <Setter Property="Foreground" Value="#EAEAEA"/>

        <Setter Property="Padding" Value="5"/>

        <Setter Property="BorderBrush" Value="#444444"/>

        <Setter Property="BorderThickness" Value="1"/>

        <Setter Property="FontSize" Value="14"/>

    </Style>

    <Style TargetType="ListBox">

        <Setter Property="FontSize" Value="14"/>

        <Setter Property="Margin" Value="5"/>

        <Setter Property="BorderBrush" Value="#b7b7b7"/>

        <Setter Property="Foreground" Value="White"/>

        <Setter Property="BorderThickness" Value="2"/>

        <Setter Property="Background" Value="#6b6b6b"/>

    </Style>

    <!-- Стиль для ComboBox -->

    <Style TargetType="ComboBox">

        <Setter Property="Foreground" Value="White"/>

        <Setter Property="FontSize" Value="15"/>

        <Setter Property="FontWeight" Value="Bold"/>

        <Setter Property="FontFamily" Value="Open Sans, Helvetica Neue, Arial, sans-serif"/>

        <Setter Property="Padding" Value="5"/>

        <Setter Property="Height" Value="30"/>

        <Setter Property="BorderBrush" Value="#5C5C5C"/>

<Setter Property="Background" Value="#3A3A3A"/>

        <Setter Property="BorderThickness" Value="1"/>

        <Setter Property="Margin" Value="0,0,0,5"/>

        <Setter Property="Template">

            <Setter.Value>

                <ControlTemplate TargetType="ComboBox">

                    <Grid>

                        <Border Name="Border"

                                Background="#4C4C4C"

                                BorderBrush="#5C5C5C"

                                BorderThickness="{TemplateBinding BorderThickness}"

                                CornerRadius="5">

                            <Grid>

                                <Grid.ColumnDefinitions>

                                    <ColumnDefinition Width="\*"/>

                                    <ColumnDefinition Width="30"/>

                                </Grid.ColumnDefinitions>

                                <ContentPresenter Name="ContentSite"

                                                  Grid.Column="0"

                                                  Content="{TemplateBinding SelectionBoxItem}"

                                                  ContentTemplate="{TemplateBinding SelectionBoxItemTemplate}"

                                                  VerticalAlignment="Center"

                                                  HorizontalAlignment="Left"

                                                  Margin="10,3,30,3"/>

                                <ToggleButton Grid.Column="1"

                                              Name="ToggleButton"

                                              ClickMode="Press"

                                              Background="Transparent"

                                              BorderThickness="0"

                                              IsChecked="{Binding IsDropDownOpen, RelativeSource={RelativeSource TemplatedParent}}">

                                    <Path Data="M 0 0 L 4 4 L 8 0 Z"

                                          Fill="White"

                                          Width="10"

                                          Height="6"

                                          HorizontalAlignment="Center"

                                          VerticalAlignment="Center"/>

                                </ToggleButton>

                            </Grid>

                        </Border>

                        <Popup Name="Popup"

                               Placement="Bottom"

                               IsOpen="{TemplateBinding IsDropDownOpen}"

                               AllowsTransparency="True"

                               Focusable="False"

                               StaysOpen="False">

                            <Border Background="#333333"

                                    BorderBrush="#5C5C5C"

                                    BorderThickness="1"

                                    CornerRadius="10"

                                    Width="{TemplateBinding ActualWidth}">

                                <ScrollViewer Margin="5">

                                    <StackPanel IsItemsHost="True"/>

                                </ScrollViewer>

                            </Border>

                        </Popup>

                    </Grid>

                    <ControlTemplate.Triggers>

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

                            <Setter TargetName="Border" Property="Background" Value="#555555"/>

                        </Trigger>

                        <Trigger Property="IsDropDownOpen" Value="True">

                            <Setter TargetName="Border" Property="Background" Value="#666666"/>

                        </Trigger>

                    </ControlTemplate.Triggers>

                </ControlTemplate>

            </Setter.Value>

        </Setter>

    </Style>

<!-- Стиль для ComboBoxItem -->

    <Style TargetType="ComboBoxItem">

        <Setter Property="Foreground" Value="White"/>

        <Setter Property="Background" Value="Transparent"/>

        <Setter Property="Padding" Value="5,2"/>

        <Setter Property="Template">

            <Setter.Value>

                <ControlTemplate TargetType="ComboBoxItem">

                    <Border Name="Border"

                            Background="{TemplateBinding Background}"

                            CornerRadius="25"

                            Padding="{TemplateBinding Padding}">

                        <ContentPresenter VerticalAlignment="Center" HorizontalAlignment="Left"/>

                    </Border>

                    <ControlTemplate.Triggers>

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

                            <Setter TargetName="Border" Property="Background" Value="#5C5C5C"/>

                        </Trigger>

                        <Trigger Property="IsSelected" Value="True">

                            <Setter TargetName="Border" Property="Background" Value="#777777"/>

                        </Trigger>

                    </ControlTemplate.Triggers>

                </ControlTemplate>

            </Setter.Value>

        </Setter>

    </Style>

<!-- Стиль для CheckBox -->

<Style TargetType="CheckBox">

    <Setter Property="FontSize" Value="15"/>

    <Setter Property="FontWeight" Value="Bold"/>

    <Setter Property="FontFamily" Value="Open Sans, Helvetica Neue, Arial, sans-serif"/>

    <Setter Property="Margin" Value="5"/>

    <Setter Property="Foreground" Value="#EAEAEA"/>

    <Setter Property="Template">

        <Setter.Value>

            <ControlTemplate TargetType="CheckBox">

                <StackPanel Orientation="Horizontal">

                    <!-- Кастомная кнопка -->

                    <Border Name="CheckBorder"

                            Width="18" Height="18"

                            CornerRadius="5"

                            BorderBrush="#EAEAEA"

                            BorderThickness="2"

                            Background="Transparent"

                            Margin="5,0,10,0">

                        <Path Name="CheckMark"

                              Data="M2,9 L6,13 L14,4"

                              Stroke="Transparent"

                              StrokeThickness="2"/>

                    </Border>

                    <!-- Текст -->

                    <ContentPresenter VerticalAlignment="Center"

                                      HorizontalAlignment="Left"

                                      Content="{TemplateBinding Content}"

                                      TextElement.Foreground="{TemplateBinding Foreground}"/>

                </StackPanel>

                <ControlTemplate.Triggers>

                    <!-- Цвет при наведении -->

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

                        <Setter TargetName="CheckBorder" Property="BorderBrush" Value="#EAEAEA"/>

                        <Setter TargetName="CheckMark" Property="Stroke" Value="#EAEAEA"/>

                    </Trigger>

                    <!-- Выбранное состояние -->

                    <Trigger Property="IsChecked" Value="True">

                        <Setter TargetName="CheckMark" Property="Stroke" Value="#EAEAEA"/>

                        <Setter TargetName="CheckBorder" Property="BorderBrush" Value="#EAEAEA"/>

                    </Trigger>

                    <!-- Disabled -->

                    <Trigger Property="IsEnabled" Value="False">

                        <Setter TargetName="CheckBorder" Property="Opacity" Value="0.5"/>

                        <Setter Property="Foreground" Value="#EAEAEA"/>

                    </Trigger>

                </ControlTemplate.Triggers>

            </ControlTemplate>

        </Setter.Value>

    </Setter>

</Style>

<!-- Стиль для RadioButton -->

<Style TargetType="RadioButton">

    <Setter Property="FontSize" Value="15"/>

    <Setter Property="FontWeight" Value="Bold"/>

    <Setter Property="FontFamily" Value="Open Sans, Helvetica Neue, Arial, sans-serif"/>

    <Setter Property="Margin" Value="5"/>

    <Setter Property="Foreground" Value="#EAEAEA"/> <!-- Тот же цвет, что и у лейблов -->

    <Setter Property="Template">

        <Setter.Value>

            <ControlTemplate TargetType="RadioButton">

                <StackPanel Orientation="Horizontal">

                    <!-- Кастомная кнопка -->

                    <Border Name="RadioBorder"

                            Width="18" Height="18"

                            CornerRadius="9"

                            BorderBrush="#EAEAEA"

                            BorderThickness="2"

                            Background="Transparent"

                            Margin="5,0,10,0">

                        <Ellipse Name="RadioDot"

                                 Width="10" Height="10"

                                 Fill="Transparent"/>

                    </Border>

                    <!-- Текст -->

<ContentPresenter VerticalAlignment="Center"

                                      HorizontalAlignment="Left"

                                      Content="{TemplateBinding Content}"

                                      TextElement.Foreground="{TemplateBinding Foreground}"/>

                </StackPanel>

                <ControlTemplate.Triggers>

                    <!-- Цвет при наведении -->

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

                        <Setter TargetName="RadioBorder" Property="BorderBrush" Value="#EAEAEA"/>

                        <Setter TargetName="RadioDot" Property="Fill" Value="#EAEAEA"/>

                    </Trigger>

                    <!-- Выбранное состояние -->

                    <Trigger Property="IsChecked" Value="True">

                        <Setter TargetName="RadioDot" Property="Fill" Value="#EAEAEA"/>

                        <Setter TargetName="RadioBorder" Property="BorderBrush" Value="#EAEAEA"/>

                    </Trigger>

                    <!-- Disabled -->

                    <Trigger Property="IsEnabled" Value="False">

                        <Setter TargetName="RadioBorder" Property="Opacity" Value="0.5"/>

                        <Setter Property="Foreground" Value="#EAEAEA"/>

                    </Trigger>

                </ControlTemplate.Triggers>

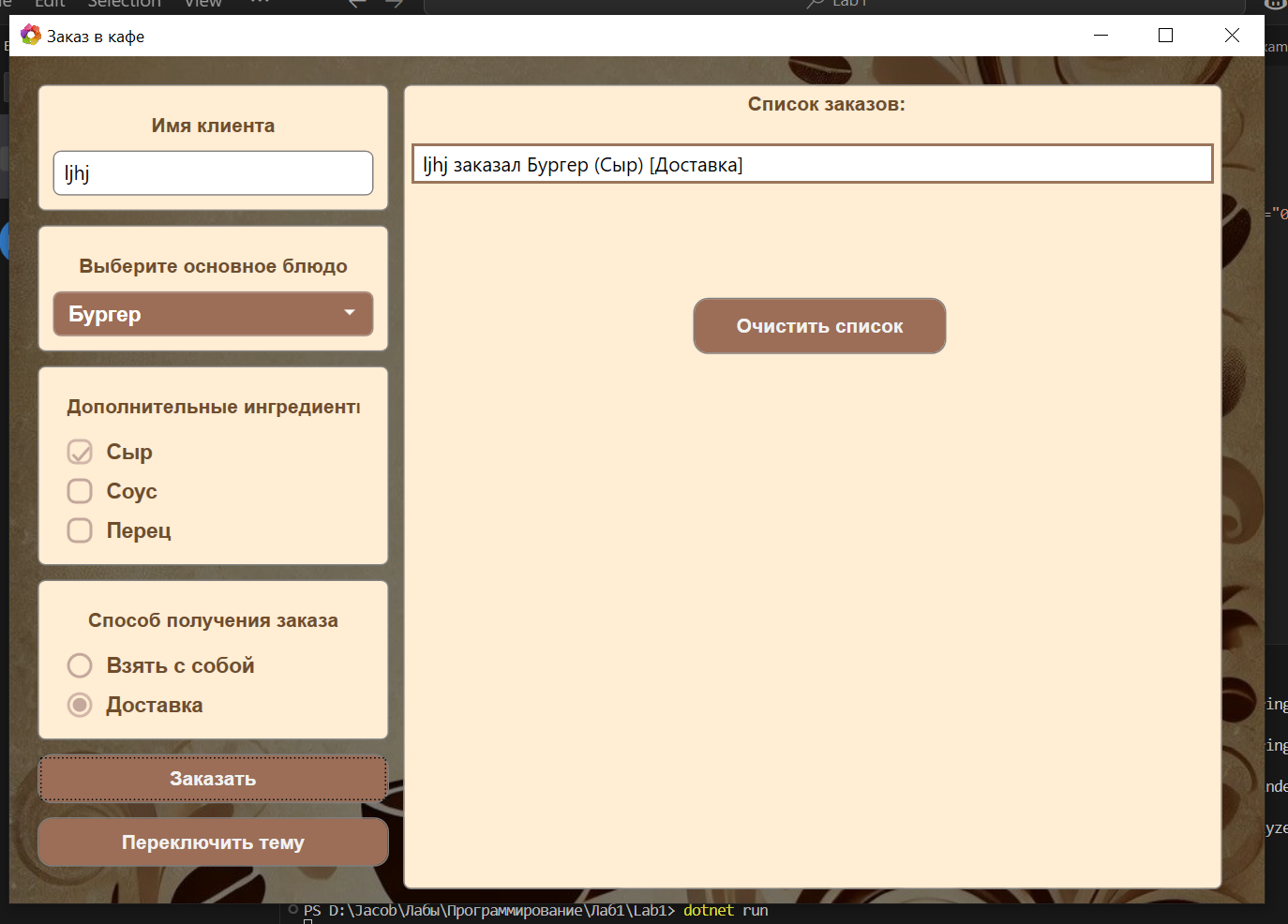
            </ControlTemplate>

        </Setter.Value>

    </Setter>

</Style>

</ResourceDictionary>

**Приложение в светлой теме :**

**Приложение в темной теме:  
A screenshot of a computer

AI-generated content may be incorrect.**

Программа составляет список заказов получая информацию из левой колонки и дополняет список заказов при нажатии кнопки

# **Вывод:**

В ходе данной лабораторной работы я изучил как разрабатывать простое графическое приложение WPF на языке C# в светлой и темной теме, в готовой программе были соблюдены основные стандарты современного дизайна.