**ИИТ БГУИР**

**Факультет повышения квалификации и переподготовки**   
  
**Дисциплина: Средства визуального программирования приложений**

**Лабораторная работа №2**

Элементы управления

Вариант 3

**Выполнил слушатель гр. 40322-1** Проверила

**Гончаров Максим Евгеньевич Желакович Ирина Миролюбовна**

Минск 2025

*Цели лабораторной работы*:

Ознакомиться с основными элементами управления, их свойствами, обработчиками событий. Научиться использовать стандартные окна сообщений.

*Выполнение работы:*

**Задание 1. Водительская лицензия.**

Разработана форма согласно макету.

1. Использовать следующие элементы: Label TextBox Image GroupBox RadioButton ComboBox DatePicker CheckBox Slider Button:

Изображение выглядит как текст, Человеческое лицо, снимок экрана, человек

Содержимое, созданное искусственным интеллектом, может быть неверным.

В качестве фона использован готовый файл fon.jpg.

Место картинки заполнено рисунком по умолчанию foto0.jpg.

1. Создан отдельный класс для представления объекта Driver:

public class Driver

{

int? number;

char? class1;

string? name;

string? adress;

double? hgt;

DateTime? dob;

DateTime? iss;

DateTime? exp;

GENDER? gender;

COLOREYES? coloreyes;

bool? donor;

string? uriImage;

public Driver()

{

}

public int? Number { get => number; set => number = value; }

public char? Class1 { get => class1; set => class1 = value; }

public string? Name { get => name; set => name = value; }

public string? Adress { get => adress; set => adress = value; }

public double? Hgt { get => hgt; set => hgt = value; }

public DateTime? Dob { get => dob; set => dob = value; }

public DateTime? Iss { get => iss; set => iss = value; }

public DateTime? Exp { get => exp; set => exp = value; }

public GENDER? Gender { get => gender; set => gender = value; }

public COLOREYES? Coloreyes { get => coloreyes; set => coloreyes = value; }

public bool? Donor { get => donor; set => donor = value; }

public string? UriImage { get => uriImage; set => uriImage = value; }

public override string? ToString()

{

return $"№{Number?.ToString()} {Class1?.ToString()} {Name?.ToString()} " +

$"{Adress?.ToString()} {Hgt?.ToString()} {Dob?.ToString("dd.MM.yyyy")} " +

$" от {Iss?.ToString("dd.MM.yyyy")} до {Exp?.ToString("dd.MM.yyyy")} пол {Gender} " +

$"глаза {coloreyes?.ToString()} {(Donor == true ? "Донор" : "Не донор")} \n {UriImage?.ToString()}";

}

}

* 1. 3. В заоконном коде: a. Создан объект driver.
  2. b. Заполнен comboBox значениями из перечисления COLOREYES.

Driver driver = new Driver();

foreach(COLOREYES color in Enum.GetValues(typeof(COLOREYES)))

* 1. comboBoxEyes.Items.Add(color);
  2. 4.Настроены обработчики событий:
  3. a. При щелчке по картинке вызывается диалоговое окно OpenDialog для загрузки нового изображения.
  4. b. По кнопке Save объект driver заполняется значениями из элементов управления и выводится в MessageBox.
  5. c. По кнопке Load объект driver инициализируется из кода фиксированным набором значений, затем выполняется заполнение всех элементов управления на основании этого объекта.
  6. d. По кнопке Clear очищаются все элементы управления и объект driver.

private void Image\_MouseDown(object sender, MouseButtonEventArgs e)

{

Microsoft.Win32.OpenFileDialog openFileDialog = new Microsoft.Win32.OpenFileDialog();

//openFileDialog.DefaultExt = ".jpg";

openFileDialog.Filter = "Image files (\*.png;\*.jpeg;\*.jpg)|\*.png;\*.jpeg;\*.jpg|All files (\*.\*)|\*.\*";

if (openFileDialog.ShowDialog() == false) return;

driver.UriImage = openFileDialog.FileName;

image.Source = new BitmapImage(new Uri(driver.UriImage, UriKind.RelativeOrAbsolute));

}

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

{

driver.Name = textBoxName.Text;

driver.Adress = textBoxAdress.Text;

if (textBoxClass1.Text.Length > 0 )

{

driver.Class1 = textBoxClass1.Text[0];

}

driver.Class1 = char.Parse(textBoxClass1.Text);

if (textBoxNumber.Text.Length > 0)

{

int value;

int.TryParse(textBoxNumber.Text, out value);

driver.Number = value;

}

driver.Number = int.Parse(textBoxNumber.Text);

driver.Hgt = sliderHGT.Value;

driver.Donor = checkBoxDonor.IsChecked;

driver.Dob = datePickerDOB.SelectedDate;

driver.Iss = datePickerISS.SelectedDate;

driver.Exp = datePickerEXP.SelectedDate;

driver.Gender = GENDER.other;

if (radioButtonM.IsChecked == true)

{

driver.Gender = GENDER.male;

}

if (radioButtonF.IsChecked == true)

{

driver.Gender = GENDER.female;

}

if(comboBoxEyes.SelectedIndex > -1)

{

driver.Coloreyes = (COLOREYES)comboBoxEyes.SelectedValue;

}

MessageBox.Show(driver.ToString());

}

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

{

NewDriver();

textBoxName.Text = driver.Name;

textBoxAdress.Text = driver.Adress;

textBoxClass1.Text = driver.Class1.ToString();

textBoxNumber.Text = driver.Number.ToString();

sliderHGT.Value = (double)driver.Hgt;

checkBoxDonor.IsChecked = driver.Donor;

datePickerDOB.SelectedDate = driver.Dob;

datePickerISS.SelectedDate = driver.Iss;

datePickerEXP.SelectedDate = driver.Exp;

comboBoxEyes.SelectedItem = driver.Coloreyes;

if (driver.Gender == GENDER.male) radioButtonM.IsChecked = true;

if (driver.Gender == GENDER.female) radioButtonF.IsChecked = true;

if (driver.Gender == GENDER.other) radioButtonO.IsChecked = true;

image.Source = new BitmapImage(new Uri(driver.UriImage, UriKind.RelativeOrAbsolute));

}

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

{

textBoxName.Text = "";

textBoxNumber.Text = "";

textBoxAdress.Text = "";

textBoxClass1.Text = "";

datePickerDOB.SelectedDate = null;

datePickerISS.SelectedDate = null;

datePickerEXP.SelectedDate = null;

radioButtonM.IsChecked = false;

radioButtonF.IsChecked = false;

radioButtonO.IsChecked = false;

comboBoxEyes.SelectedIndex = -1;

checkBoxDonor.IsChecked = false;

sliderHGT.Value = sliderHGT.Minimum;

image.Source = new BitmapImage(new Uri("/images/foto0.jpg", UriKind.Relative));

driver = new Driver();

MessageBox.Show("Все данные очищены!");

}

Кнопки отрабатываются корректно в любой последовательности. Отчистить:

Изображение выглядит как текст, Человеческое лицо, снимок экрана, человек

Содержимое, созданное искусственным интеллектом, может быть неверным.

Загрузить: Изображение выглядит как текст, человек, снимок экрана, Человеческое лицо

Содержимое, созданное искусственным интеллектом, может быть неверным.

Сохранить:

Изображение выглядит как текст, человек, Человеческое лицо, снимок экрана

Содержимое, созданное искусственным интеллектом, может быть неверным.

**Задание 2. Карточка регистрации (по вариантам).**

Вариант 3. **Карточка боевого юнита в компьютерной игре**. Контейнер компоновки - WrapPanel.

Разработано приложение, аналогичное предыдущему примеру.

Изображение выглядит как текст, Человеческое лицо, снимок экрана, человек

Содержимое, созданное искусственным интеллектом, может быть неверным.

Использован указанный контейнер компоновки WrapPanel, как основной Приложение содержит следующие элементы управления:

* 1. WrapPanel, Label, TextBlock,TextBox, Image,GroupBox, RadioButton, CheckBox, Expander, ComboBox или ListBox, DatePicker, Slider, Button.

<Window x:Class="LB2\_CombatUnitCard.MainWindow"

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

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

Title="Карточка боевого юнита" Height="800" Width="600">

<WrapPanel Margin="10">

<TextBlock Text="Карточка боевого юнита"

FontSize="18" FontWeight="Bold"

Margin="5" Width="550"

TextAlignment="Center"/>

<WrapPanel Width="550" Margin="5">

<Image x:Name="unitImage" Source="Images/foto0.jpg"

Height="120" Width="120" Margin="0,0,20,10"

MouseDown="UnitImage\_MouseDown"/>

<StackPanel Width="400">

<Label Content="Название юнита:"/>

<TextBox x:Name="txtName" Width="200" Margin="0,0,0,10"/>

<Label Content="Тип юнита:"/>

<ComboBox x:Name="cmbType" Width="200" Margin="0,0,0,10">

<ComboBoxItem>Пехота</ComboBoxItem>

<ComboBoxItem>Кавалерия</ComboBoxItem>

<ComboBoxItem>Артиллерия</ComboBoxItem>

<ComboBoxItem>Маги</ComboBoxItem>

</ComboBox>

</StackPanel>

</WrapPanel>

<WrapPanel Width="550" Margin="5">

<GroupBox Header="Фракция" Width="250" Margin="0,0,20,10">

<StackPanel>

<RadioButton x:Name="radFaction1" Content="Альянс" Margin="5"/>

<RadioButton x:Name="radFaction2" Content="Орда" Margin="5"/>

<RadioButton x:Name="radFaction3" Content="Нейтралы" Margin="5"/>

</StackPanel>

</GroupBox>

<GroupBox Header="Уровень" Width="250" Margin="0,0,0,10">

<StackPanel>

<Slider x:Name="sliderLevel" Minimum="1" Maximum="100" Value="1"

Margin="5" Width="200"/>

<TextBlock x:Name="txtLevel" Text="Уровень: 1" Margin="5"

HorizontalAlignment="Center"/>

</StackPanel>

</GroupBox>

</WrapPanel>

<WrapPanel Width="550" Margin="5">

<GroupBox Header="Характеристики" Width="250" Margin="0,0,20,10">

<StackPanel>

<Label Content="Здоровье:"/>

<TextBox x:Name="txtHealth" Text="100" Width="100" Margin="0,0,0,10"/>

<Label Content="Атака:"/>

<TextBox x:Name="txtAttack" Text="20" Width="100" Margin="0,0,0,10"/>

</StackPanel>

</GroupBox>

<GroupBox Header="Статус" Width="250" Margin="0,0,0,10">

<StackPanel>

<CheckBox x:Name="chkElite" Content="Элитный юнит" Margin="10"/>

<Label Content="Дата создания:" Margin="10,5,10,0"/>

<DatePicker x:Name="dpCreationDate" Width="180" Margin="10"/>

</StackPanel>

</GroupBox>

</WrapPanel>

<WrapPanel Width="550" Margin="5">

<Expander Header="Способности" Width="530" Margin="0,0,0,10">

<ListBox x:Name="lstAbilities" Height="100" Width="500" SelectionMode="Multiple">

<ListBoxItem>Ближний бой</ListBoxItem>

<ListBoxItem>Дальний бой</ListBoxItem>

<ListBoxItem>Защита</ListBoxItem>

<ListBoxItem>Лечение</ListBoxItem>

<ListBoxItem>Магия</ListBoxItem>

<ListBoxItem>Скорость</ListBoxItem>

</ListBox>

</Expander>

</WrapPanel>

<WrapPanel Width="550" Margin="5" HorizontalAlignment="Center">

<Button x:Name="btnSave" Content="Сохранить" Width="120" Height="30"

Margin="10" Click="BtnSave\_Click"/>

<Button x:Name="btnLoad" Content="Загрузить" Width="120" Height="30"

Margin="10" Click="BtnLoad\_Click"/>

<Button x:Name="btnClear" Content="Очистить" Width="120" Height="30"

Margin="10" Click="BtnClear\_Click"/>

</WrapPanel>

</WrapPanel>

* 1. </Window>

Создан класс:

public class CombatUnit

{

public CombatUnit()

{

}

public string Name { get; set; } = "";

public string Type { get; set; } = "";

public int Level { get; set; } = 1;

public int Health { get; set; } = 100;

public int Attack { get; set; } = 20;

public string Faction { get; set; } = "";

public bool IsElite { get; set; } = false;

public DateTime CreationDate { get; set; } = DateTime.Now;

public string ImagePath { get; set; } = "";

public List<string> Abilities { get; set; } = new List<string>();

public override string ToString()

{

string abilitiesStr = Abilities.Count > 0 ? string.Join(", ", Abilities) : "Нет способностей";

return $"Имя: {Name}\nТип: {Type}\nУровень: {Level}\n" +

$"Здоровье: {Health}\nАтака: {Attack}\nФракция: {Faction}\n" +

$"Элитный: {(IsElite ? "Да" : "Нет")}\n" +

$"Дата создания: {CreationDate:dd.MM.yyyy}" +

$"Способности: {abilitiesStr}"; ;

}

}

Кнопки отрабатываются корректно в любой последовательности.

// Сохранить данные из формы

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

{

currentUnit.Name = txtName.Text;

currentUnit.Type = cmbType.Text;

currentUnit.Level = (int)sliderLevel.Value;

currentUnit.Health = int.Parse(txtHealth.Text);

currentUnit.Attack = int.Parse(txtAttack.Text);

currentUnit.IsElite = chkElite.IsChecked ?? false;

currentUnit.CreationDate = dpCreationDate.SelectedDate ?? DateTime.Now;

if (radFaction1.IsChecked == true) currentUnit.Faction = "Альянс";

else if (radFaction2.IsChecked == true) currentUnit.Faction = "Орда";

else if (radFaction3.IsChecked == true) currentUnit.Faction = "Нейтралы";

currentUnit.Abilities.Clear();

foreach (ListBoxItem item in lstAbilities.SelectedItems)

{

currentUnit.Abilities.Add(item.Content.ToString());

}

MessageBox.Show("Данные сохранены!\n\n" + currentUnit.ToString());

}

// Загрузить данные в форму

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

{

// Создаем случайные данные

Random random = new Random();

string[] names = { "Элитный воин", "Магический стрелок", "Тяжелый пехотинец", "Летучий разведчик" };

string[] types = { "Пехота", "Кавалерия", "Артиллерия", "Маги" };

string[] factions = { "Альянс", "Орда", "Нейтралы" };

string[] allAbilities = { "Ближний бой", "Дальний бой", "Защита", "Лечение", "Магия", "Скорость" };

currentUnit.Name = names[random.Next(names.Length)];

currentUnit.Type = types[random.Next(types.Length)];

currentUnit.Level = random.Next(1, 101);

currentUnit.Health = random.Next(50, 501);

currentUnit.Attack = random.Next(20, 201);

currentUnit.Faction = factions[random.Next(factions.Length)];

currentUnit.IsElite = random.Next(2) == 0;

currentUnit.CreationDate = DateTime.Now.AddDays(-random.Next(1000));

currentUnit.ImagePath = $"Images/{random.Next(1, 8)}.jpg";

currentUnit.Abilities.Clear();

for (int i = 0; i < allAbilities.Length; i++)

{

if (random.Next(2) == 0)

{

currentUnit.Abilities.Add(allAbilities[i]);

}

}

// Заполняем элементы управления данными из объекта

txtName.Text = currentUnit.Name;

cmbType.Text = currentUnit.Type;

sliderLevel.Value = currentUnit.Level;

txtHealth.Text = currentUnit.Health.ToString();

txtAttack.Text = currentUnit.Attack.ToString();

chkElite.IsChecked = currentUnit.IsElite;

dpCreationDate.SelectedDate = currentUnit.CreationDate;

// Устанавливаем фракцию

radFaction1.IsChecked = currentUnit.Faction == "Альянс";

radFaction2.IsChecked = currentUnit.Faction == "Орда";

radFaction3.IsChecked = currentUnit.Faction == "Нейтралы";

// Устанавливаем способности в ListBox

lstAbilities.SelectedItems.Clear();

foreach (ListBoxItem item in lstAbilities.Items)

{

if (currentUnit.Abilities.Contains(item.Content.ToString()))

{

item.IsSelected = true;

}

}

unitImage.Source = new BitmapImage(new Uri(currentUnit.ImagePath, UriKind.RelativeOrAbsolute));

MessageBox.Show("Данные загружены\n\n" + currentUnit.ToString());

}

// Очистить форму

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

{

// Очистка полей

txtName.Text = "";

cmbType.SelectedIndex = -1;

sliderLevel.Value = 1;

txtHealth.Text = " ";

txtAttack.Text = " ";

chkElite.IsChecked = false;

dpCreationDate.SelectedDate = null;

// Сброс радиокнопок

radFaction1.IsChecked = false;

radFaction2.IsChecked = false;

radFaction3.IsChecked = false;

// Сброс ListBox

lstAbilities.SelectedItems.Clear();

// Сброс изображения

unitImage.Source = new BitmapImage(new Uri("Images/foto0.jpg", UriKind.Relative));

// Очистка переменной

currentUnit = new CombatUnit();

MessageBox.Show("Форма и переменная очищены!");

}

// Обработчик клика по изображению

private void UnitImage\_MouseDown(object sender, System.Windows.Input.MouseButtonEventArgs e)

{

OpenFileDialog dialog = new OpenFileDialog();

dialog.Filter = "Image files (\*.png;\*.jpeg;\*.jpg)|\*.png;\*.jpeg;\*.jpg|All files (\*.\*)|\*.\*";

if (dialog.ShowDialog() == false) return;

currentUnit.ImagePath = dialog.FileName;

unitImage.Source = new BitmapImage(new Uri(currentUnit.ImagePath, UriKind.RelativeOrAbsolute));

}

Изображение выглядит как текст, программное обеспечение, Человеческое лицо, человек

Содержимое, созданное искусственным интеллектом, может быть неверным.

Изображение выглядит как текст, снимок экрана, человек, программное обеспечение

Содержимое, созданное искусственным интеллектом, может быть неверным.

Изображение выглядит как текст, Человеческое лицо, снимок экрана, программное обеспечение

Содержимое, созданное искусственным интеллектом, может быть неверным.

Код проектов всех лабораторных работ, задания и отчеты находятся в публичном доступе на сайте GitHub: <https://github.com/Maxevgen555/SVPP_LB>. Файл решения SVPP\_LB.sln находится в папке \ЛБ1\LB1\_Calc\.

*Вывод:*

Я ознакомился с основными элементами управления, их свойствами, обработчиками событий. Научился использовать стандартные окна сообщений, разработав форму согласно макету в задании 1 и выполняя индивидуальное задание 2.