

Лабораторная работа 10

Элементы управления: дерево, таблица

Общее количество баллов: 13 баллов (8 баллов за работу + 5 баллов за индивидуальное задание)

Задание 1. Элемент TreeView

Данный элемент управления предназначен для древовидного отображения данных в окне приложения. Может содержать как коллекцию элементов `TreeViewItem`, так и другое содержимое

- 1 Создайте новый проект WPF
- 2 Задайте имя проекта в соответствии с шаблоном: **Фамилия_прХ_заданиеУ**. Где Фамилия – это ВАША фамилия, Х-номер практики, У – номер задания. Не забываем сохранять проект на рабочий стол, если проект был создан на ПК колледжа.
- 3 Добавьте к проекту следующий код разметки:

```
<Grid>
<TreeView>
  <TextBox>09.02.07 Информационные системы и программирование</TextBox>
  <TreeViewItem Header="ПМ 01 Разработка модулей программного обеспечения для компьютерных систем">
    <TreeViewItem Header="МДК 01.01 Разработка программных модулей" />
    <TreeViewItem Header="МДК 01.02 Поддержка и тестирование программных модулей" />
    <TreeViewItem Header="МДК 01.03 Разработка мобильных приложений" />
    <TreeViewItem Header="МДК 01.04 Системное программирование" />
    <TreeViewItem Header="МДК 01.05 Разработка интерактивных веб-приложений" />
  </TreeViewItem>
  <TreeViewItem Header="ПМ 02. Осуществление интеграции программных модулей">
    <TreeViewItem Header="Технология разработки программного обеспечения">
      <TreeViewItem Header="Инструментальные средства разработки программного обеспечения" />
      <TreeViewItem Header="Математическое моделирование" />
    </TreeViewItem>
  </TreeViewItem>
  <TreeViewItem Header="ПМ 04. Сопровождение и обслуживание программного обеспечения компьютерных систем">
    <TreeViewItem Header="Внедрение и поддержка компьютерных систем" />
    <TreeViewItem Header="Обеспечение качества функционирования компьютерных систем" />
  </TreeViewItem>
  <TreeViewItem Header="ПМ 11. Разработка, администрирование и защита баз данных">
    <TreeViewItem Header="Технология разработки и защиты баз данных" />
  </TreeViewItem>
</TreeView>
</Grid>
```

Однако все же лучше обертывать элементы в объекты `TreeViewItem`. С помощью его свойства **Header** мы можем установить текстовую метку или заголовок узла дерева. Элемент `TreeViewItem` предлагает также ряд свойств для управления состоянием: **IsExpanded** (принимает логическое значение и показывает, раскрыт ли узел) и **IsSelected** (показывает, выбран ли узел).

Чтобы отследить выбор или раскрытие узла, мы можем обработать соответствующие события. Событие **Expanded** возникает при раскрытии узла, а событие **Collapsed**, наоборот, при его сворачивании.

Выбор узла дерева мы можем обработать с помощью обработки события **Selected**.

4 Добавьте события к первому элементу дерева:

```
для компьютерных систем" Expanded="TreeViewItem_Expanded" Selected="TreeViewItem_Selected">
```

5 Пропишите обработчики этих событий:

```
private void TreeViewItem_Expanded(object sender, RoutedEventArgs e)
{
    TreeViewItem tvItem = (TreeViewItem)sender;
    MessageBox.Show("Узел " + tvItem.Header.ToString() + " раскрыт");
}
```

Ссылка: 0

```
private void TreeViewItem_Selected(object sender, RoutedEventArgs e)
{
    TreeViewItem tvItem = (TreeViewItem)sender;
    MessageBox.Show("Выбран узел: " + tvItem.Header.ToString());
}
```

6 Самостоятельно добавьте эти события ко всем узлам дерева.

7 Запустите приложение. В случае необходимости, исправьте ошибки*

Задание 2. DataGrid

DataGrid во многом похож на ListView, но более сложный по характеру и допускает редактирование содержимого таблицы.

1 Добавьте самостоятельно класс «учеба» в проект. В классе опишите три свойства: предмет, преподаватель, кабинет. сделайте свойства общедоступными (см. л/р 6).

2 Выведите объекты в таблицу DataGrid. Чтобы DataGrid автоматически разбивал таблицу на столбцы, установите свойство `AutoGenerateColumns="True"`

```

xmlns:local="clr-namespace:Андреева_практика10_задание1"
xmlns:col="clr-namespace:System.Collections;assembly=microsoft.collections"
mc:Ignorable="d"
Title="MainWindow" Height="450" Width="800">
<Grid VerticalAlignment="Top">
<DataGrid x:Name="myGrid" AutoGenerateColumns="True" ItemsSource="{DynamicResource ResourceKey=Learns}">
<DataGrid.Resources>
<col:ArrayList x:Key="Learns">
<local:Learn Caption="Разработка программных модулей" Teacher="Андреева АР" Auditor="7" />
</col:ArrayList>
</DataGrid.Resources>
</DataGrid>

```

В данном случае префикс `local` ссылается на пространство имен текущего проекта, в котором определен класс `Learn` (`xmlns:local="clr-namespace:Андреева_практика10_задание1"`), а `col` - префикс-ссылка на пространство имен `System.Collections` (`xmlns:col="clr-namespace:System.Collections;assembly=microsoft.collections"`).

- 3 Самостоятельно добавьте еще 3 записи в таблицу*
- 4 Добавьте программное описание добавления элементов в таблицу сразу после инициализации формы:

```

Ссылка: 0
public MainWindow()
{
    InitializeComponent();

    List<Learn> LernList = new List<Learn>
    {
        new Learn { Caption="Основы проектирования баз данных", Teacher="Ялина ЯВ", Auditor=5 }
    };
    myGrid.ItemsSource = LernList;
}

```

- 5 Самостоятельно добавьте еще 3 записи*
- 6 Запустите приложение

Некоторые полезные свойства DataGrid

`RowBackground` и `AlternatingRowBackground` Устанавливают фон строки. Если установлены оба свойства, цветовой фон чередуется: `RowBackground` - для нечетных строк и `AlternatingRowBackground` - для четных

`ColumnHeaderHeight` Устанавливает высоту строки названий столбцов.

`ColumnWidth` Устанавливает ширину столбцов.

`RowHeight` Устанавливает высоту строк.

`GridLinesVisibility` Устанавливает видимость линий, разделяющих столбцы и строки. Имеет четыре значения - `All` - видны все линии, `Horizontal` - видны только горизонтальные линии, `Vertical` - видны только вертикальные линии, `None` - линии отсутствуют

`HeadersVisibility` Задаёт видимость заголовков

`HorizontalGridLinesBrush` и `VerticalGridLinesBrush` Задаёт цвет горизонтальных и вертикальных линий соответственно

Хотя предыдущий пример довольно прост, в нем есть несколько недочетов. Во-первых, у нас нет возможности повлиять на расстановку столбцов. Во-вторых, заголовки определены по названиям свойств, которые на английском языке, а хотелось бы на русском. В этом случае мы должны определить свойства отображения столбцов сами. Для этого надо воспользоваться свойством `DataGrid.Columns` и определить коллекцию столбцов для отображения в таблице.

Причем можно задать также и другой тип столбца, отличный от текстового. `DataGrid` поддерживает следующие варианты столбцов:

`DataGridTextColumn` Отображает элемент `TextBlock` или `TextBox` при редактировании

`DataGridHyperlinkColumn` Представляет гиперссылку и позволяет переходить по указанному адресу

`DataGridCheckBoxColumn` Отображает элемент `CheckBox`

`DataGridComboBoxColumn` Отображает выпадающий список - элемент `ComboBox`

`DataGridTemplateColumn` Позволяет задать специфичный шаблон для отображения столбца

7 Добавьте выравнивание по верху для первой таблице

8 Добавьте еще одну таблицу снизу:

```

<DataGrid x:Name="learGrid" AutoGenerateColumns="False" HorizontalGridLinesBrush="DarkGray"
RowBackground="LightGray" AlternatingRowBackground="White" VerticalAlignment="Bottom">

    <DataGrid.Items>
        <local:Learn Caption="Разработка программных модулей" Teacher="Андреева АР" Auditor="7"/>
    </DataGrid.Items>
    <DataGrid.Columns>
        <DataGridTextColumn Header="Предмет" Binding="{Binding Path=Caption}" Width="90" />
        <DataGridHyperlinkColumn Header="Преподаватель" Binding="{Binding Path=Teacher}" Width="80" />
        <DataGridTextColumn Header="Кабинет" Binding="{Binding Path=Auditor}" Width="50" />
    </DataGrid.Columns>
</DataGrid>

```

9 Самостоятельно добавьте еще 3 записи в таблицу*

Среди свойств DataGrid одним из самых интересных является RowDetailsTemplate. Оно позволяет задать шаблон отображения дополнительной информации касательно данной строки.

10 Добавьте настройки для строки:

```

<DataGrid.RowDetailsTemplate>
    <DataTemplate>
        <StackPanel Orientation="Horizontal">
            <TextBlock Text="{Binding Path=Auditor}" />
            <TextBlock Text=" корпус 2" />
        </StackPanel>
    </DataTemplate>
</DataGrid.RowDetailsTemplate>

```

11 Запустите приложение*

Задание 3. ProgressBar и Slider

ProgressBar и Slider представляют элементы, основанные на диапазонах значений. То есть они хранят и отображают числовые данные на определенном диапазоне.

Все они являются наследниками класса RangeBase, поэтому наследуют такие его свойства, как:

✓ **Value** указывает на текущее значение элемента (представлено типом Double)

✓ **Maximum/Minimum** указывает на максимальное/минимальное значение элемента.

Slider представляет собой обычный ползунок. Он добавляет следующие свойства:

✓ Orientation:	указывает	ориентацию	ползунка	-
горизонтальную(Horizontal) или вертикальную (Vertical)				
✓ Delay:	указывает	время	в миллисекундах,	по истечении которого ползунок переместится на одну единицу после щелчка.
✓ Interval:	указывает	время	в миллисекундах,	по истечении которого ползунок может перемещаться
✓ TickPlacement:	задает	визуализацию	шкалы ползунка.	По умолчанию имеет значение None (отсутствие шкалы). Значение BottomRight создают шкалу в нижней части ползунка, TopLeft - в верхней, Both - по обоим сторонам.
✓ TickFrequency:	указывает	частоту	появления	отметок на шкале ползунка.
✓ IsSelectionRangeEnabled:	задает	затенение	участка ползунка.	Если оно установлено в True, то начальная и конечная отметка затенения задаются с помощью свойств SelectionStart и SelectionEnd .

- 1 Добавьте в проект сплайдер и текстовый блок
- 2 Задайте сплайдеру значения в интервале от 0 до 100
- 3 Опишите обработчик события ValueChanged так, чтобы в текстовый блок выдавалось значение сплайдера *

ProgressBar представляет индикатор, отображающий выполнение определенного процесса. Также имеет свойство Orientation, которое устанавливает вертикальное или горизонтальное расположение индикатора. Для связи с реальными процессами вся логика изменения индикатора, его свойства Value должна отрабатываться в коде.

- 4 Добавьте на форму строку прогресса и кнопку.
- 5 Пропишите обработчик события нажатия на кнопку так, чтобы при каждом нажатии на кнопке изменялось состояние строки прогресса **

Задание 4 (индивидуальное). Решение задач

Вариант задания определяется порядковым номером студента в журнале преподавателя и является УНИКАЛЬНЫМ в пределах курса!!!

НОМЕР ВАРИАНТА НЕОБХОДИМО УТОЧНИТЬ У ПРЕПОДАВАТЕЛЯ

Для группы ИСП-1-20 номер задания = порядковому номеру

Для группы ИС-1-19 номер задания = порядковому номеру+25

Задание – проект необходимо сохранить в папке с названием ИНДИВИДУАЛЬНАЯ РАБОТА и номером практической работы

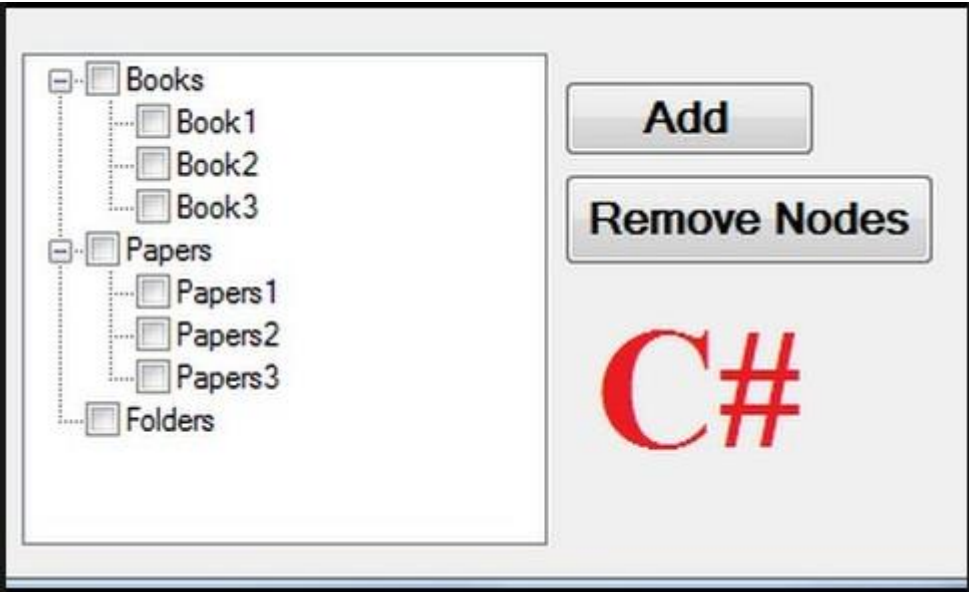
Программа должна иметь комментарии к действиям. Также перед кодом обработчика события нажатия на кнопку необходимо добавить комментарий с личными данными студента (фамилия, имя, группа и текст задания).

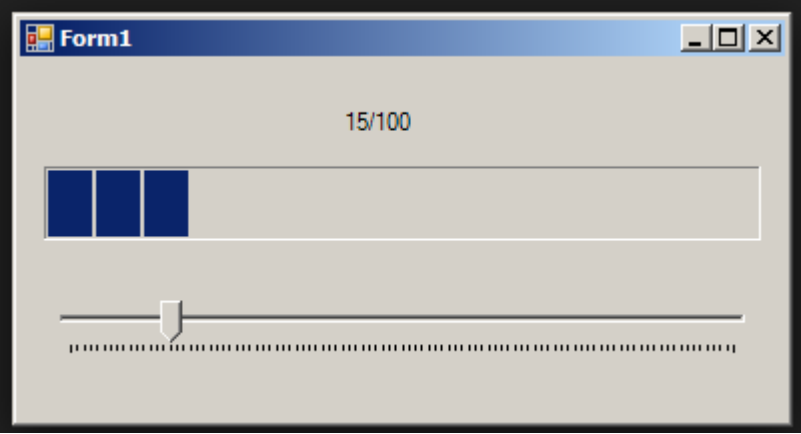
Программа при закрытии должна выдавать диалоговое окно подтверждения закрытия формы.

Не забывайте проверять на правильность ввода данных. Все поля должны быть заполнены. В противном случае, программа должна выдавать ошибку.

Индивидуальное задание оценивается следующим образом: 1 балл за комментарии, 1 балл за диалоговое окно при выходе, 1 балл за проверку на валидность данных, 1 балл за верное решение задачи, 1 балл за интерфейс приложения.

Создайте WPF приложение. Для реализации использовать изученные элементы.

Вариант	Задание
1)	<p>Создать приложение:</p> 
2)	Приложение «Кулинарная книга». Рецепты представлены в виде дерева. Приложение позволяет просматривать выбранный рецепт
3)	Создать приложение, где компонент ProgressBar – показывает расстояние от Москвы до пункта назначения
4)	Создать приложение «Просмотр расписания». Дан список групп. Выбрав группу и день недели, можем увидеть ее расписание на этот день. Использовать элемент дерево.
5)	Создать приложение «Страны мира». Разделить части света на страны, а страны на города. Представить все в виде раскрывающегося списка. При выборе города, отображается его фотография. Использовать элемент дерево.
6)	Создать приложение, где перемещение ползунка TrackBar меняет день недели от понедельника до воскресенья.
7)	Создать приложение, где ProgressBar отображает часовой пояс для страны.
8)	Создать следующее приложение:

	
9)	Создать проект, где компонент TrackBar – возраст сотрудника, а компонент ProgressBar – показывает стаж работы сотрудника.
10)	Создать приложение, где компонент TrackBar – количество порций блюда. А компонент ProgressBar – стадия процесса готовки.
11)	Приложение «Электронный журнал». Создать приложение, позволяющее просматривать состав учебных групп. Использовать элемент дерево.
12)	Приложение «Отдел кадров». Создать приложение, позволяющее просматривать информацию о сотрудниках. Сотрудники объединены в отделы. Использовать элемент дерево.
13)	Приложение «Матрицы». Создать приложение, позволяющее изучать теоретические сведения по работе с матрицами. Использовать элемент дерево.
14)	Приложение «Движение тела под углом к горизонту». Создать приложение, позволяющее изучать теоретические сведения по теме. Использовать элемент дерево.
15)	Приложение «ветеринарная клиника». Приложение позволяет просматривать информацию о клиентах ветеринарной клиники
16)	Создать приложение «Игра», позволяющее выбрать уровень сложности. В зависимости от выбранного уровня сложности, отображаются различные лабиринты.

17)	Приложение «Биржа труда». Приложение позволяет просматривать информацию о вакансиях в зависимости от работодателя
18)	Приложение «Магазин». Приложение позволяет просматривать информацию о товарах в зависимости от раздела
19)	Приложение «Криптография». Создать приложение, позволяющее изучать теоретические сведения по теме
20)	Приложение «Тест». Приложение позволяет просматривать вопросы по темам.
21)	Приложение «Менеджеры». Приложение позволяет просматривать информацию о менеджерах компании в различных отделах
22)	Приложение «Автомобиль». Приложение позволяет просматривать характеристики автомобиля в зависимости от марки и модели
23)	Приложение «Магазин». Приложение позволяет просматривать информацию о товарах в зависимости от раздела
24)	Приложение «Учебный курс». Создать меню Вашей учебной темы в виде дерева. При выборе элемента, выдается сообщение «Вы выбрали...»
25)	Приложение «птицефабрика». В приложении пользователь может просматривать информацию об услугах.
26)	Приложение «ювелирный магазин». В приложении пользователь может просматривать информацию о товарах.
27)	Приложение «ломбард». В приложении пользователь может просматривать информацию об услугах.
28)	Приложение «хим.чистка». В приложении пользователь может просматривать информацию об услугах.
29)	Приложение «кафе». В приложении пользователь может просматривать информацию об услугах.
30)	Приложение «клининговая компания». В приложении пользователь может просматривать информацию об услугах.

31)	Приложение «служба доставки». В приложении пользователь может просматривать информацию об услугах.
32)	Приложение «частная клиника». В приложении пользователь может просматривать информацию об услугах.
33)	Приложение «ветеринарная клиника». В приложении пользователь может просматривать информацию об услугах.
34)	Приложение «букмекерская компания». В приложении пользователь может просматривать информацию об услугах.
35)	Приложение «салон красоты». В приложении пользователь может просматривать информацию об услугах.
36)	Приложение «туристическое агентство». В приложении пользователь может просматривать информацию об услугах.
37)	Приложение «банк». В приложении пользователь может просматривать информацию об услугах.
38)	Приложение «магазин бытовой техники». Пользователь может просматривать товары и информацию о них.
39)	Приложение «геометрическая фигура». Приложение позволяет просматривать информацию о различных геометрических фигурах
40)	Создать приложение «репертуар театра». Можно посмотреть состав актеров на выбранный спектакль.
41)	Создать приложение «Цирк», где представления можно посмотреть в зависимости от дня недели. А по каждому представлению можно посмотреть артистов и номера.
42)	Создать приложение, где вертикальный и горизонтальный размер кнопки меняется с помощью компонентов TrackBar
43)	Приложение «Гостиница». Приложение позволяет просматривать номера предложенных гостиниц.
44)	Приложение «Записная книжка». Приложение позволяет просматривать информацию о контактах в записной книжке

45)	Создать приложение, где TrackBar – вес животного, а ProgressBar - % содержания анестезии для проведения операции
46)	Создать приложение, где ProgressBar – уровень топлива в баке автомобиля. При изменении расстояния, которое проехал автомобиль, меняется уровень топлива в автомобиле.
47)	Создать приложение «Бюро находок», содержащее сведения о находках: название дата находки и место находки. При выборе находки, открываются сведения
48)	Создать приложение-тест из 10 вопросов на тему «Основы логики». Компонент TrackBar – показывает порядковый номер вопроса, а компонент ProgressBar – процент верно решенных заданий.
49)	Создать приложение, где перемещение ползунка TrackBar меняет день недели от понедельника до воскресенья.
50)	Создать приложение, где ProgressBar отображает часовой пояс для страны.