

Лабораторная работа №11

Создание проекта на Visual C# для работы с XML-файлами

Данная программа показывает, как используя **LINQ-запрос** создать XML-файл, загрузить существующий документ и добавить в него информацию, выбрать нужную информацию и отобразить ее в поле компонента **List View**.

***Задание:** создать приложение, которое позволит выполнять добавление информации в XML-файл, её извлечение и поиск.*

Выходная форма приложения должна иметь примерно следующий вид:

Автор	Название

Для начала работы необходимо:

1. Объявите в классе переменную

```
XDocument doc;
```

2. Настройте компонент **List View**

- 2.1. Добавьте две колонки: *Автор* и *Название*. Установите их ширину
- 2.2. Установите видимость линий сетки между колонками и строками
3. Для кнопок «Найти», «Добавить» и «Сохранить» установить **Enabled** в false

Для кнопки «Загрузить»:

1. Загрузить данные из файла в переменную

```
doc = XDocument.Load(@"\books.xml");
```

2. Активировать кнопки «Найти», «Добавить» и «Сохранить»

3. Объявить

```
IEnumerable<XElement> books = doc.Elements();
```

4. В цикле добавить содержимое XML-файла в **ListView**

```
foreach (XElement aBooks in books.Elements())
{
    listView1.Items.Add(aBooks.Element("Author").Value); // первый узел- элемент
    listView1.Items[listView1.Items.Count - 1].
    SubItems.Add(aBooks.Element("Title").Value); // остальные узлы - подэлементы
}
```

5. Деактивировать кнопку «Создать»

Для кнопки «Добавить»:

```
doc.Element("корневой_элемент").Add(new XElement("book",
    new XElement("Author", источник1),
    new XElement("Title", источник2),
    new XElement("Description", источник3)
));
```

источник – объект, из которого извлекаются данные для записи в файл (текстовое поле, метка, список и т.д.)

Для кнопки «Создать»:

1. Проверить существует ли уже файл

1.1. Объявить

```
FileInfo fi = new FileInfo(@"\books.xml");
```

1.2. Для проверки использовать условие с методом **Exists**

1.3. Если файл уже существует, то сообщить пользователю и дать возможность выбора – заменить или нет.

2. Если файл не существует или пользователь выбрал команду заменить:

```
doc =
    new XDocument(
//данный блок может повторяться несколько раз
        new XElement("books",
            new XElement("book",
                new XElement("Author", "Иванов И.И."),
                new XElement("Title", "Математика"),
                new XElement("Description", "")
            ),
//
        )
    )
```

Таким образом добавьте 5 записей.

3. Сохранить в физическом файле

```
doc.Save(@"\books.xml");
```

Для кнопки «Найти»

1. Очистить **ListView**

2. Создать LINQ-запрос для выбора названий книг одного автора

```
IEnumerable<XElement> query =
    from b in doc.Elements().Elements()
    where b.Elements("Author").Any(n =>
        n.Value.Contains(источник))
    select b;
```

3. Использовать цикл для формирования и вывода результата

```
foreach (XElement e1 in имя_LINQ_запрос) //1 цикл
{
    bool firstElement = true;
    foreach (XElement e2 in e1.Elements()) // 2 цикл
    {
        } }
}
```

4. В теле второго цикла необходимо сделать проверку: если `firstElement == true`, тогда добавляем значение первого узла

(элемента) из `e2.Value` и присваиваем `firstElement = false` иначе добавляем значение остальных узлов (подэлементов).

Задание для самостоятельного выполнения

Создать приложение, которое позволит выполнять добавление информации в XML-файл, её извлечение и поиск. XML-файл содержит информацию о баллах ЕГЭ пяти дисциплин абитуриентов. Пользователь может производить поиск информации по фамилии абитуриента.