

С# Файлы

Разработка проектов. В процессе написания программы следует разработать структуру хранения данных, продумать интерфейс приложения, главное окно приложения, подходящие диалоговые окна.

Чтение из XML-файла

Чтение XML-файлов — это необходимый метод извлечения информации из XML-файла. Например, вы храните свои параметры конфигурации в XML-файле, вы можете извлечь параметры из XML-файла при загрузке приложения.

Задание №1. Разработать приложение, которое позволяет считывать данные из XML-файла, извлекает отдельные записи из XML-файла, используя кнопки для навигации и чтения других существующих записей.

Разработайте форму приложения, интерфейс которой представлен на рисунке 1.

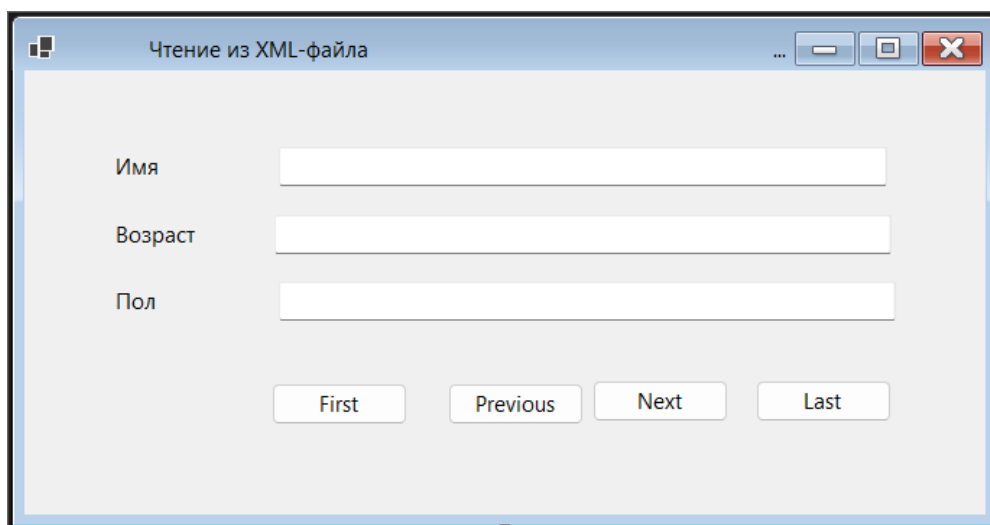
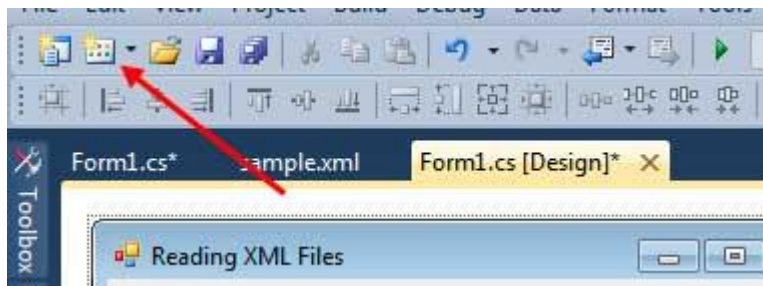


Рисунок 1

Используйте имена `buttonFirst`, `buttonPrevious`, `buttonNext` и `buttonLast` для четырех элементов управления кнопками. Используйте имена `textBoxName`, `textBoxAge` и `textBoxGender` для трех текстовых полей.

Для тестирования нам понадобится пример XML-файла, содержащего записи, которые будут считываться нашим приложением. Создайте новый XML-файл, нажав кнопку **Добавить новый элемент** на панели инструментов.



Затем выберите XML-файл из списка шаблонов. Назовите файл `sample.xml`. Visual Studio создаст для вас XML-файл. Удалите его содержимое и используйте следующую XML-разметку.

```
<?xml version="1.0" encoding="utf-8"?>
<Persons>
  <Person name="John Smith">
    <Age>30</Age>
    <Gender>Male</Gender>
  </Person>
  <Person name="Mike Folley">
    <Age>25</Age>
    <Gender>Male</Gender>
  </Person>
  <Person name="Lisa Carter">
    <Age>22</Age>
    <Gender>Female</Gender>
  </Person>
  <Person name="Jerry Frost">
    <Age>27</Age>
    <Gender>Male</Gender>
  </Person>
  <Person name="Adam Wong">
    <Age>35</Age>
    <Gender>M</Gender>
  </Person>
</Persons>
```

Файл XML содержит пять записей лиц. Наше приложение начнет чтение первой записи, и пользователь может просмотреть другие записи, нажав кнопки навигации.

Переключитесь в режим просмотра кода и импортируйте необходимое пространство имен System.Xml.

```
private XmlDocument doc;
private XmlElement root;
private XmlElement currentPerson;
private const string PATH = @"sample.xml"; // Укажите правильный путь к файлу
private int current = 0;
private int max;
```

Док типа XmlDocument используется для загрузки XML-файла и управления его содержимым. Переменная root будет использоваться для хранения корневого элемента документа. Переменная currentPerson используется для хранения текущего выбранного элемента person. Мы использовали путь «...sample.xml», поскольку XML-файл, который мы создали ранее, расположен на два каталога выше исполняемого файла (.. означает переход на один каталог выше). Опять же, вы меняете это на другое место, где находится XML-файл. Текущая переменная используется для отслеживания того, какую запись показывать, с использованием индекса, начинающегося с 0. Она инициализируется значением 0, чтобы указать, что текущая запись является первой записью. Переменная max будет использоваться для хранения максимального индекса, который пригодится позже.

Нам нужно создать метод утилиты, который будет использоваться обработчиками событий для отображения деталей выбранной записи в текстовом поле. Добавьте этот метод внутри класса Form1 .

```
private void ShowDetails(XmlElement currentPerson)
{
    textBoxName.Text = currentPerson.Attributes["name"].Value;
    textBoxAge.Text = currentPerson.GetElementsByTagName("Age")[0].InnerText;
    textBoxGender.Text = currentPerson.GetElementsByTagName("Gender")[0].InnerText;
}
```

Этот метод принимает выбранного в данный момент человека. Сначала значение атрибута «имя» извлекается и отображается в textBoxName . Мы использовали свойство

Attribute класса XElement и использовали индексатор строк, который представляет имя атрибута. Затем мы используем свойство Value указанного атрибута.

Мы извлекаем значение элемента Age , используя метод GetElementsByTagName XElement . Мы передаем имя элемента, и метод ищет все дочерние элементы с указанным именем и возвращает их как XmlNodeList . Поскольку XmlNodeList — это коллекция, и ожидается только один результат, мы можем получить доступ к первому совпадающему элементу, используя индекс 0.

```
textBoxAge.Text = currentPerson.GetElementsByTagName("Age")[0].InnerText;
```

Эта строка может быть довольно запутанной для новичков, потому что я сделал все в одной строке. Строка просто означает получить первый элемент [0] коллекции, возвращенной GetElementsByTagName , а затем получить доступ к свойству InnerText этого первого элемента. Мы можем переписать эту одну строку кода в более читаемый код, но это потребует больше кода, как показано ниже.

```
XmlNodeList results = currentPerson.GetElementsByTagName("Age");
```

```
XmlNode first = results[0];
```

```
textBoxAge.Text = first.InnerText;
```

Следующая строка кода похожа на строку для извлечения возраста, но на этот раз элемент пола ищется с помощью метода GetElementsByTagName .

После завершения необходимого метода утилиты мы теперь можем создать наши обработчики событий. Переключитесь в представление Design, щелкнув вкладку Design или используя сочетание клавиш Shift+F7 .

Щелкните пустое место на форме, чтобы создать обработчик событий для события Load . Используйте следующий код для обработчика событий.

```
private void Form1_Load(object sender, EventArgs e)
{
    try
    {
        if (!File.Exists(PATH))
        {
            MessageBox.Show("XML файл не найден!");
            return;
        }

        doc = new XmlDocument();
        doc.Load(PATH);

        // Получаем корневой элемент
        root = doc.DocumentElement;

        // Определяем максимальный индекс
        max = root.GetElementsByTagName("Person").Count - 1;

        // Получаем первую запись
        currentPerson = (XXmlElement)root.ChildNodes[current];

        // Отображаем данные
        ShowDetails(currentPerson);

        UpdateNavigationButtons();
    }
}
```

```

catch (Exception ex)
{
    MessageBox.Show($"Ошибка загрузки XML: {ex.Message}");
}
}

```

Этот код будет выполнен после завершения загрузки формы. Он извлечет и отобразит первую запись в XML-файле. Сначала мы создаем объект `XmlDocument`. Затем мы загружаем XML-файл, указанный в `PATH`, с помощью метода `Load` класса `XmlDocument`. Мы извлекаем корневой элемент, используя свойство `DocumentElement`. Мы вычислили максимально возможный индекс, получив все дочерние элементы с именем `Person`, а затем подсчитали их. Мы вычитаем из него 1, поскольку индексы начинаются с 0. Значение `max` нам понадобится позже. Затем мы извлекаем первого человека, используя метод `ChildNodes` и передавая значение `current` (которое теперь равно 0) в качестве индекса. Свойство `ChildNodes` представляет собой коллекцию всех дочерних узлов узла, и, передав 0 в качестве индекса, мы можем извлечь первую запись. Поскольку `ChildNodes` состоит из объектов `XmlNode`, нам нужно преобразовать узел в `XmlElement`, чтобы мы могли сохранить его в объекте `XmlElement`. Конечно, мы также можем использовать метод `GetElementsByTagName`, но я использовал метод `ChildNodes`, чтобы продемонстрировать вам альтернативную технику. Теперь мы передаем извлеченный элемент в метод, который мы создали ранее. Передача его в этот метод извлекает детали из текущего выбранного элемента `person`.

Вернитесь в конструктор и дважды щелкните `buttonFirst`. Используйте следующий код обработчика событий.

```

private void buttonFirst_Click(object sender, EventArgs e)
{
    current = 0;
    currentPerson = (XmlElement)root.ChildNodes[current];
    ShowDetails(currentPerson);
    UpdateNavigationButtons();
}

```

Первая строка сбрасывает текущий индекс до 0, а затем извлекает дочерний узел с индексом 0. Затем мы использовали метод `ShowDetails` для отображения сведений об извлеченном `XmlElement`. Теперь добавим обработчик событий для `buttonPrevious`.

```

private void buttonPrevious_Click(object sender, EventArgs e)
{
    current = (current - 1 < 0) ? 0 : current - 1;
    currentPerson = (XmlElement)root.ChildNodes[current];
    ShowDetails(currentPerson);
    UpdateNavigationButtons();
}

```

Код похож на обработчик событий `buttonFirst`, за исключением первой строки. Предыдущая запись просто представлена выражением `(current - 1)`. Первая строка использует условный оператор, проверяющий, меньше ли предыдущий элемент нуля. Если это так, мы просто присваиваем ноль и запрещаем пользователю доступ к отрицательным индексам. Это необходимо для предотвращения `IndexOutOfRangeException`. Если пользователь находится в начальной записи, то нажатие `buttonPrevious` не даст никакого эффекта. Если предыдущий индекс равен или больше нуля, то текущий индекс присваивается со значением, уменьшенным на 1, чтобы указать, что теперь мы показываем предыдущую запись. Затем запись с новым индексом извлекается и отображается с помощью метода `ShowDetails`.

Теперь добавим обработчик событий для кнопки Next, которая показывает следующую запись.

```
private void buttonNext_Click(object sender, EventArgs e)
{
    current = (current + 1 > max) ? max : current + 1;
    currentPerson = (XmlElement)root.ChildNodes[current];
    ShowDetails(currentPerson);
    UpdateNavigationButtons();
}
```

Опять же, единственное отличие здесь — это вычисление нового текущего индекса. Следующая запись представлена выражением (current + 1), и мы проверяем, превысит ли она максимально возможный индекс, который хранится в переменной max . Если превышает, то мы просто назначаем значение max в качестве текущего индекса. Если следующий индекс равен или меньше max , то следующий индекс назначается в качестве нового текущего индекса. Элемент с новым индексом извлекается и отображается пользователю.

Наконец, добавьте обработчик событий к buttonLast , дважды щелкнув его в Designer. Используйте следующий код.

```
private void buttonLast_Click(object sender, EventArgs e)
{
    current = max;
    currentPerson = (XmlElement)root.ChildNodes[current];
    ShowDetails(currentPerson);
    UpdateNavigationButtons();
}
```

Поскольку мы будем извлекать последнюю запись, текущему индексу просто присваивается конечный индекс, указанный значением max . Затем элемент извлекается и отображается пользователю.

Запустите приложение, и вам будут представлены данные первой записи. Щелкните кнопки, чтобы перейти к другим записям в XML-файле. Теперь вы успешно создали приложение, которое считывает записи из XML-файла, используя классы XML DOM.

Запись в XML-файл

Используя классы и методы XML Document Object Model, можно создавать XML-документ и сохранить его в файл.

Задание №2. Разработать приложение, которое позволяет пользователю указать свое имя, возраст и пол. Затем данные будут добавлены в XML-файл.

- Создайте новое приложение Windows Forms и создайте форму, похожую на рисунок 2.

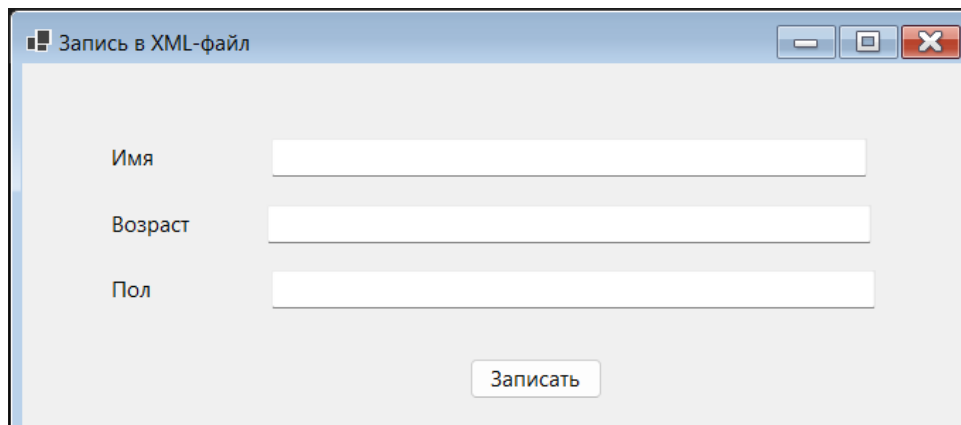


Рисунок 2

- Измените имя текстовых полей на `textBoxName` , `extBoxAge` и `textBoxGender` . Измените имя кнопки на `buttonAdd`.
- Добавьте `private XmlDocument` и строку, указывающую путь к XML-файлу, который мы будем использовать во всем коде. Напишите их прямо под объявлением класса.
- Вернитесь в конструктор и дважды щелкните кнопку «Записать», чтобы сгенерировать обработчик событий для ее события `Click`, а затем используйте следующий код.

```

private void button1_Click(object sender, EventArgs e)
{
    try
    {
        // Проверка введенных данных
        if (string.IsNullOrEmpty(textBoxName.Text) ||
            string.IsNullOrEmpty(textBoxAge.Text) ||
            string.IsNullOrEmpty(textBoxGender.Text))
        {
            MessageBox.Show("Пожалуйста, заполните все поля!",
                "Ошибка",
                MessageBoxButtons.OK,
                MessageBoxIcon.Warning);
            return;
        }

        doc = new XmlDocument();

        if (!File.Exists(PATH))
        {
            CreateNewXmlFile();
        }
        else
        {
            AppendToExistingXmlFile();
        }

        MessageBox.Show("Данные успешно добавлены в XML-файл!",
            "Успех",
            MessageBoxButtons.OK,
            MessageBoxIcon.Information);

        ClearFormFields();
    }
    catch (Exception ex)

```

```

-         {
-             MessageBox.Show($"Ошибка при сохранении данных:
{ex.Message}", "Ошибка",
-                             MessageBoxButtons.OK,
-                             MessageBoxIcon.Error);
-         }
-     }

-     private void CreateNewXmlFile()
-     {
-         // Создаем XML-декларацию
-         XmlDeclaration declaration = doc.CreateXmlDeclaration("1.0",
"utf-8", null);
-         doc.AppendChild(declaration);

-         // Добавляем комментарий
-         XmlComment comment = doc.CreateComment("Файл создан программой
XmlWriterApp");
-         doc.AppendChild(comment);

-         // Создаем корневой элемент
-         XmlElement root = doc.CreateElement("Persons");
-         doc.AppendChild(root);

-         // Добавляем первую запись
-         AddPersonToXml(root);

-         // Сохраняем документ
-         doc.Save(PATH);
-     }

-     private void AppendToExistingXmlFile()
-     {
-         // Загружаем существующий файл
-         doc.Load(PATH);

-         // Получаем корневой элемент
-         XmlElement root = doc.DocumentElement;

-         // Добавляем новую запись
-         AddPersonToXml(root);

-         // Сохраняем изменения
-         doc.Save(PATH);
-     }

-     private void AddPersonToXml(XmlElement root)
-     {
-         // Создаем элементы для новой записи
-         XmlElement person = doc.CreateElement("Person");
-         XmlAttribute nameAttr = doc.CreateAttribute("name");
-         XmlElement ageElement = doc.CreateElement("Age");
-         XmlElement genderElement = doc.CreateElement("Gender");

-         // Заполняем данные
-         nameAttr.Value = textBoxName.Text;
-         ageElement.InnerText = textBoxAge.Text;
-         genderElement.InnerText = textBoxGender.Text;

-         // Собираем структуру
-         person.Attributes.Append(nameAttr);
-         person.AppendChild(ageElement);

```

```

-         person.AppendChild(genderElement);
-         root.AppendChild(person);
-     }
-
-     private void ClearFormFields()
-     {
-         textBoxName.Text = string.Empty;
-         textBoxAge.Text = string.Empty;
-         textBoxGender.Text = string.Empty;
-         textBoxName.Focus();
-     }
- }

```

При нажатии кнопки **Добавить** создается новый `XmlDocument` . Он будет представлять XML-документ, который будет записан в файл, указанный в `PATH` . Теперь мы проверяем, был ли файл уже создан, с помощью метода `Exists` класса `System.IO.File` . Если он не существует, то создаем новый файл и добавляем первую запись. Если файл уже существует, то загружаем файл и добавляем новую запись.

После загрузки файла по указанному пути с помощью метода `Load` корневой элемент этого файла извлекается с помощью свойства `DocumentElement`. Теперь, когда у нас есть доступ к корневому элементу, мы можем добавить больше элементов `person`. Мы создали новый элемент `person`, используя введенные пользователем данные. Затем мы добавляем нового `person` в конец списка дочерних элементов корневого элемента. Затем мы обновляем XML-файл, снова вызывая метод `Save`. Последние строки кода просто показывают сообщение об успешном завершении, а затем очищают текстовые поля, чтобы можно было принимать новые входные данные.

Запустите приложение и введите данные не менее двух лиц. XML-файл будет расположен по пути, указанному переменной `PATH`. Вы можете изменить этот путь на другое местоположение. Вы увидите созданный XML-файл и сгенерированное содержимое.

```

<?xml version="1.0" encoding="utf-8"?>
<!--Файл создан программой XmlWriterApp-->
<Persons>
  <Person name="Иван Петров">
    <Age>35</Age>
    <Gender>Мужской</Gender>
  </Person>
  <Person name="Мария Сидорова">
    <Age>28</Age>
    <Gender>Женский</Gender>
  </Person>
  <Person name="Алексей Иванов">
    <Age>42</Age>
    <Gender>Мужской</Gender>
  </Person>
</Persons>

```

Теперь вы успешно сгенерировали XML-файл с использованием классов и методов XML DOM.

Самостоятельная работа. Добавить в разработанное приложение следующие возможности:

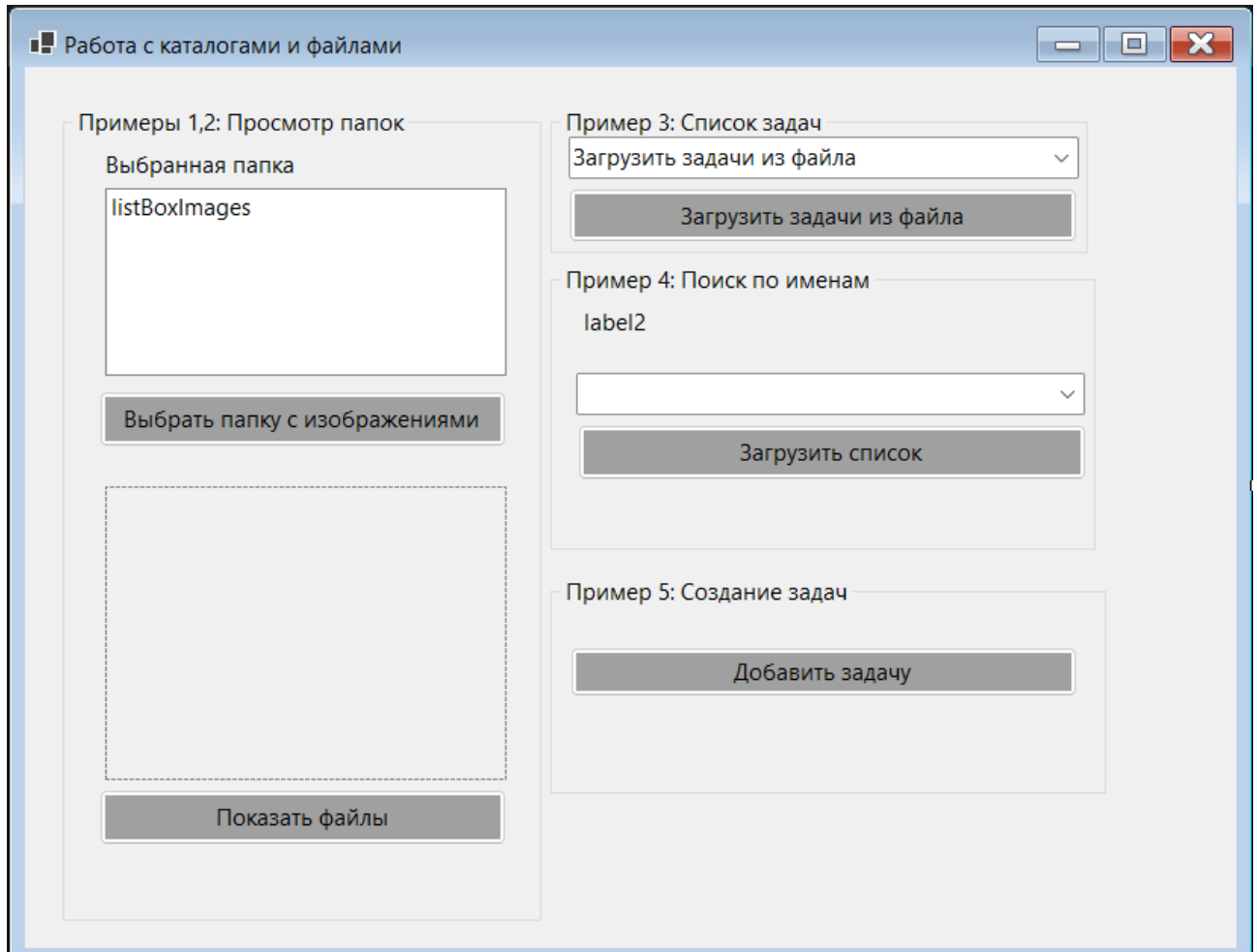
- Валидацию возраста (только числа)
- Выбор пола из выпадающего списка

- Редактирование и удаление записей
- Поиск по имени

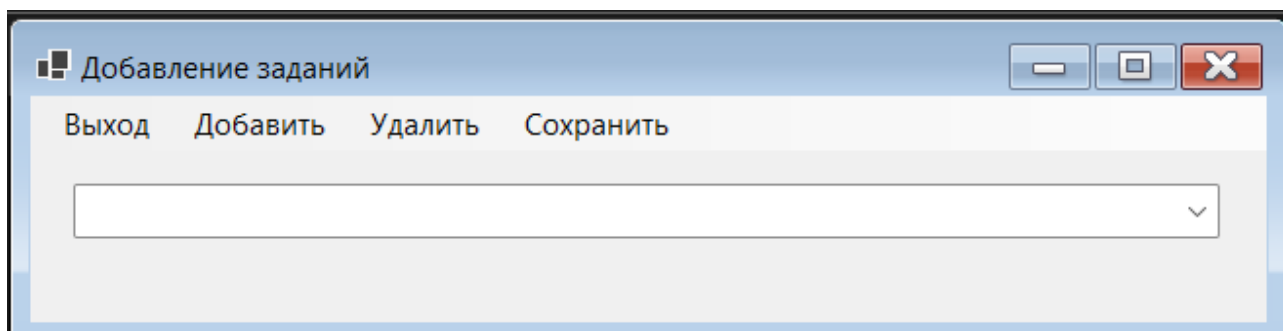
Текстовые файлы

Текстовые файлы предоставляют идеальное решение для чтения и записи данных, которые должны использоваться часто, но слишком объемны для ручного ввода при каждом запуске программы. Здесь вы узнаете, как читать и писать текстовые файлы с помощью .NET Framework и языка C#.

Задание №3. Создать приложение со следующими функциями:

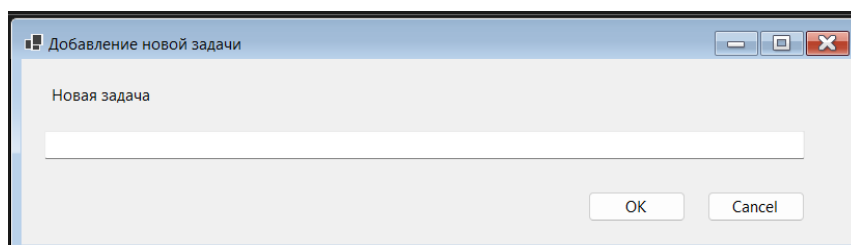


1. Отобразить имена изображений из указанного каталога в элементе управления ListBox, используя класс FolderBrowseDialog. Этот класс позволяет пользователю искать каталог, создавать каталог и выбирать каталог. Выбор каталога осуществляется с использованием древовидной структуры. Можно выбирать только файлы из файловой системы.
2. При выборе изображения из списка, отображать его в элемент управления PictureBox.
3. Считать из текстового файла задачи и загрузить их в элемент управления ComboBox.
4. Считать из файла имена людей и даты их дней рождения, загрузить имена в ComboBox. При выборе имени человека, показать в метке, сколько лет этому человеку исполнится в текущем году.
5. Разработать модуль для создания списка дел. Добавьте новую форму Windows Form в проект. Оформите ее как показано на рисунке:



Модуль содержит меню, состоящего из следующих команд: Выход, Добавить, Удалить, Сохранить:

- Выход – закрыть приложение.
- Добавить – для добавления элементов в элемент управления ComboBox. Добавьте в проект новую форму Windows, спроектируйте ее, как показано ниже:



- Удалить – удалить элемент из элемента управления ComboBox.
- Сохранить – сохранить все элементы из ComboBox в текстовый файл.

Задание №4. Для инициализации исходной матрицы необходимо использовать программу-генератор, разработанную в ходе выполнения учебной задачи. Целесообразно использовать ограничение на генерируемые числа, например, ограничить значения случайно генерируемых чисел величиной 100, 10 и т.д., чтобы вывод программы был более читабельным. Алгоритм решения должен работать для массивов любой величины, поэтому необходимо протестировать программу на различных входных данных.

Вариант	Задание
1	<p>Во входном файле содержится две строки: первая содержит одно целое число N (количество чисел во второй строке), вторая строка содержит N вещественных чисел (каждое от 0 до 10^5).</p> <p>В результате работы программы должен быть сформирован выходной файл, который содержит следующие значения:</p> <ol style="list-style-type: none"> 1. количество чисел, больших среднего арифметического исходных чисел; 2. сумма всех чисел, меньших среднего арифметического; 3. максимальное число.

2	<p>Во входном файле содержится две строки: первая содержит одно целое число N (количество чисел во второй строке), вторая строка содержит N вещественных чисел (каждое от 0 до 10^5).</p> <p>В результате работы программы должен быть сформирован выходной файл, который содержит следующие значения:</p> <ol style="list-style-type: none"> 1) произведение чисел, меньших 50 (если таких чисел нет, то выводится 0); 2) сумма всех чисел, больших среднего арифметического; 3) минимальное число.
3	<p>Во входном файле содержится две строки: первая содержит одно целое число N (количество чисел во второй строке), вторая строка содержит N вещественных чисел (каждое от 0 до 10^5).</p> <p>В результате работы программы должен быть сформирован выходной файл, который содержит следующие значения:</p> <ol style="list-style-type: none"> 1) произведение чисел, меньших 50 (если таких чисел нет, то выводится 0); 2) все числа, большие среднего арифметического через пробел.
4	<p>Во входном файле содержится две строки: первая содержит одно целое число N (количество чисел во второй строке), вторая строка содержит N вещественных чисел (каждое от 0 до 10^5).</p> <p>В результате работы программы должен быть сформирован выходной файл, который содержит следующие значения:</p> <ol style="list-style-type: none"> 1) среднее арифметическое максимального и минимального элементов; 2) среднее арифметическое всех элементов; 3) все числа, которые больше среднего арифметического максимального и минимального элементов.
5	<p>Во входном файле содержится две строки: первая содержит одно целое число N (количество чисел во второй строке), вторая строка содержит N вещественных чисел (каждое от 0 до 10^5).</p> <p>В результате работы программы должен быть сформирован выходной файл, который содержит следующие значения:</p> <ol style="list-style-type: none"> 1) максимальный элемент; 2) среднее арифметическое всех элементов; 3) модифицированный массив, в котором все элементы исходного, меньшие среднего арифметического, заменены на 0.
6	<p>Во входном файле содержится две строки: первая содержит одно целое число N (количество чисел во второй строке), вторая строка содержит N вещественных чисел (каждое от -10^5 до 10^5).</p> <p>В результате работы программы должен быть сформирован выходной файл, который содержит следующие значения:</p> <ol style="list-style-type: none"> 1) максимальный отрицательный элемент; 2) минимальный положительный элемент; 3) список элементов, не превышающих по модулю 50.

7	<p>Во входном файле содержится две строки: первая содержит одно целое число N (количество чисел во второй строке), вторая строка содержит N целых чисел (каждое от -10^5 до 10^5).</p> <p>В результате работы программы должен быть сформирован выходной файл, который содержит следующие значения:</p> <ol style="list-style-type: none"> 1) максимальный четный элемент; 2) минимальный нечетный элемент; 3) сумму элементов, которые делятся на 5.
8	<p>Во входном файле содержится две строки: первая содержит одно целое число N (количество чисел во второй строке), вторая строка содержит N вещественных чисел (каждое от -10^5 до 10^5).</p> <p>В результате работы программы должен быть сформирован выходной файл, который содержит следующие значения:</p> <ol style="list-style-type: none"> 1) среднее арифметическое всех элементов; 2) максимальный элемент; 3) модифицированный массив, в котором все элементы не превышающие по модулю 1000 заменены на 0.
9	<p>Во входном файле содержится две строки: первая содержит одно целое число N (количество чисел во второй строке), вторая строка содержит N целых чисел (каждое от -10^5 до 10^5).</p> <p>В результате работы программы должен быть сформирован выходной файл, который содержит следующие значения:</p> <ol style="list-style-type: none"> 1) максимальный элемент среди всех кратных 5; 2) минимальный элемент среди всех кратных 3; 3) сумму элементов, которые делятся на 10.
10	<p>Во входном файле содержится две строки: первая содержит одно целое число N (количество чисел во второй строке), вторая строка содержит N вещественных чисел (каждое от -10^5 до 10^5).</p> <p>В результате работы программы должен быть сформирован выходной файл, который содержит следующие значения:</p> <ol style="list-style-type: none"> 1) второй по величине элемент; 2) минимальный положительный элемент; 3) список элементов, не превышающих по модулю 1000.
11	<p>Во входном файле содержится две строки: первая содержит одно целое число N (количество чисел во второй строке), вторая строка содержит N целых чисел (каждое от -10^5 до 10^5).</p> <p>В результате работы программы должен быть сформирован выходной файл, который содержит следующие значения:</p> <ol style="list-style-type: none"> 1) сумму элементов, кратных 7; 2) произведение элементов, кратных 5; 3) список элементов кратных 10.

12	<p>Во входном файле содержится две строки: первая содержит одно целое число N (количество чисел во второй строке), вторая строка содержит N вещественных чисел (каждое от -10^5 до 10^5).</p> <p>В результате работы программы должен быть сформирован выходной файл, который содержит следующие значения:</p> <ol style="list-style-type: none"> 1) максимальный элемент, не превышающий 1000; 2) минимальный по модулю элемент; 3) список элементов, больших 5000 и меньших 7000.
13	<p>Во входном файле содержится две строки: первая содержит одно целое число N (количество чисел во второй строке), вторая строка содержит N целых чисел (каждое от -10^5 до 10^5).</p> <p>В результате работы программы должен быть сформирован выходной файл, который содержит следующие значения:</p> <ol style="list-style-type: none"> 1) количество элементов, кратных 5; 2) максимальный элемент; 3) список элементов, кратных 8.
14	<p>Во входном файле содержится две строки: первая содержит одно целое число N (количество чисел во второй строке), вторая строка содержит N вещественных чисел (каждое от -10^5 до 10^5).</p> <p>В результате работы программы должен быть сформирован выходной файл, который содержит следующие значения:</p> <ol style="list-style-type: none"> 1) сумму элементов, не превышающих по модулю 1000; 2) разность максимального и минимального элемента; 3) список отрицательных элементов.
15	<p>Во входном файле содержится две строки: первая содержит одно целое число N (количество чисел во второй строке), вторая строка содержит N целых чисел (каждое от -10^5 до 10^5).</p> <p>В результате работы программы должен быть сформирован выходной файл, который содержит следующие значения:</p> <ol style="list-style-type: none"> 1) максимальный четный элемент; 2) минимальный нечетный элемент; 3) модифицированный массив, в котором все элементы, меньшие среднего арифметического заменены на +.
16	<p>Во входном файле содержится две строки: первая содержит одно целое число N (количество чисел во второй строке), вторая строка содержит N вещественных чисел (каждое от -10^5 до 10^5).</p> <p>В результате работы программы должен быть сформирован выходной файл, который содержит следующие значения:</p> <ol style="list-style-type: none"> 1) модифицированный массив, в котором все отрицательные элементы заменены на «-», а положительные на «+»; 2) среднее арифметическое положительных элементов; 3) максимальный среди отрицательных элементов.

17	<p>Во входном файле содержится две строки: первая содержит одно целое число N (количество чисел во второй строке), вторая строка содержит N целых чисел (каждое от -10^5 до 10^5).</p> <p>В результате работы программы должен быть сформирован выходной файл, который содержит следующие значения:</p> <ol style="list-style-type: none"> 1) сумму элементов кратных 10; 2) произведение элементов, больших 0 и меньших 1000; 3) список элементов, кратных 11.
18	<p>Во входном файле содержится две строки: первая содержит одно целое число N (количество чисел во второй строке), вторая строка содержит N вещественных чисел (каждое от -10^5 до 10^5).</p> <p>В результате работы программы должен быть сформирован выходной файл, который содержит следующие значения:</p> <ol style="list-style-type: none"> 1) сумма элементов больших 1000 и меньших 5000; 2) сумма максимального и минимального элементов; 3) список элементов, не превышающих по модулю 1000.
19	<p>Во входном файле содержится две строки: первая содержит одно целое число N (количество чисел во второй строке), вторая строка содержит N целых чисел (каждое от -10^5 до 10^5).</p> <p>В результате работы программы должен быть сформирован выходной файл, который содержит следующие значения:</p> <ol style="list-style-type: none"> 1) максимальный нечетный элемент; 2) среднее арифметическое; 3) список четных элементов.
20	<p>Во входном файле содержится две строки: первая содержит одно целое число N (количество чисел во второй строке), вторая строка содержит N вещественных чисел (каждое от -10^5 до 10^5).</p> <p>В результате работы программы должен быть сформирован выходной файл, который содержит следующие значения:</p> <ol style="list-style-type: none"> 1) максимальный отрицательный элемент (если таких нет, то вывести No); 2) минимальный положительный элемент (если таких нет, то вывести No); 3) список элементов, превышающих по модулю 10000 (если таких нет, то вывести No).
21	<p>Во входном файле содержится две строки: первая содержит одно целое число N (количество чисел во второй строке), вторая строка содержит N целых чисел (каждое от -10^5 до 10^5).</p> <p>В результате работы программы должен быть сформирован выходной файл, который содержит следующие значения:</p> <ol style="list-style-type: none"> 1) максимальный элемент, кратный 5 (если таких нет, то вывести No); 2) список элементов, больших, чем среднее арифметическое положительных чисел; 3) максимум среди отрицательных чисел (если таких нет, то вывести No).

22	<p>Во входном файле содержится две строки: первая содержит одно целое число N (количество чисел во второй строке), вторая строка содержит N вещественных чисел (каждое от -10^5 до 10^5).</p> <p>В результате работы программы должен быть сформирован выходной файл, который содержит следующие значения:</p> <ol style="list-style-type: none"> 1) второй по величине элемент; 2) модифицированный массив, в котором все элементы, не превышающие по модулю 5000, заменены на «х»; 3) максимальный отрицательный элемент (если таких нет, то вывести No).
23	<p>Во входном файле содержится две строки: первая содержит одно целое число N (количество чисел во второй строке), вторая строка содержит N целых чисел (каждое от -10^5 до 10^5).</p> <p>В результате работы программы должен быть сформирован выходной файл, который содержит следующие значения:</p> <ol style="list-style-type: none"> 1) минимальный элемент кратный 8 (если таких нет, то вывести No); 2) максимальный элемент кратный 12 (если таких нет, то вывести No); 3) модифицированный массив, в котором все элементы, большие по модулю 15000, заменены на «+».
24	<p>Во входном файле содержится две строки: первая содержит одно целое число N (количество чисел во второй строке), вторая строка содержит N вещественных чисел (каждое от -10^5 до 10^5).</p> <p>В результате работы программы должен быть сформирован выходной файл, который содержит следующие значения:</p> <ol style="list-style-type: none"> 1) максимальный элемент; 2) список элементов, отстающих от максимального не более чем на 1000 (если таких нет, то вывести No); 3) модифицированный массив, в котором все элементы, большие по модулю 5000, заменены на «+», а меньшие на «-».
25	<p>Во входном файле содержится две строки: первая содержит одно целое число N (количество чисел во второй строке), вторая строка содержит N целых чисел (каждое от -10^5 до 10^5).</p> <p>В результате работы программы должен быть сформирован выходной файл, который содержит следующие значения:</p> <ol style="list-style-type: none"> 1) максимальный элемент, кратный 10 (если таких нет, то вывести No); 2) минимальный элемент, кратный 100 (если таких нет, то вывести No); 3) модифицированный массив, в котором все четные элементы, заменены на Ч, а нечетные – на нЧ.

26	<p>Во входном файле содержится две строки: первая содержит одно целое число N (количество чисел во второй строке), вторая строка содержит N вещественных чисел (каждое от -10^5 до 10^5).</p> <p>В результате работы программы должен быть сформирован выходной файл, который содержит следующие значения:</p> <ol style="list-style-type: none"> 1) среднее арифметическое отрицательных элементов; 2) среднее арифметическое положительных элементов; 3) список элементов, значения которых заключены между средними арифметическими, рассчитанными в заданиях 1) и 2).
27	<p>Во входном файле содержится две строки: первая содержит одно целое число N (количество чисел во второй строке), вторая строка содержит N целых чисел (каждое от -10^5 до 10^5).</p> <p>В результате работы программы должен быть сформирован выходной файл, который содержит следующие значения:</p> <ol style="list-style-type: none"> 1) количество элементов, кратных 10; 2) произведение элементов, кратных 100; 3) модифицированный массив, в котором все элементы кратные 5 заменены на 5, остальные – на x.
28	<p>Во входном файле содержится две строки: первая содержит одно целое число N (количество чисел во второй строке), вторая строка содержит N вещественных чисел (каждое от -10^5 до 10^5).</p> <p>В результате работы программы должен быть сформирован выходной файл, который содержит следующие значения:</p> <ol style="list-style-type: none"> 1) модуль максимального отрицательного элемента; 2) минимальный положительный элемент; 3) список элементов, лежащих между значениями, найденными в заданиях 1) и 2).
29	<p>Во входном файле содержится две строки: первая содержит одно целое число N (количество чисел во второй строке), вторая строка содержит N целых чисел (каждое от -10^5 до 10^5).</p> <p>В результате работы программы должен быть сформирован выходной файл, который содержит следующие значения:</p> <ol style="list-style-type: none"> 1) количество элементов, больших 1000; 2) минимальный элемент, кратный 7; 3) список элементов, лежащих между 0 и 1000.
30	<p>Во входном файле содержится две строки: первая содержит одно целое число N (количество чисел во второй строке), вторая строка содержит N вещественных чисел (каждое от -10^5 до 10^5).</p> <p>В результате работы программы должен быть сформирован выходной файл, который содержит следующие значения:</p> <ol style="list-style-type: none"> 1) среднее арифметическое отрицательных чисел (если таких нет, то 0); 2) минимальное положительное число; 3) список элементов, лежащих между значениями, найденными в задачах 1) и 2).

31	<p>Во входном файле содержится две строки: первая содержит одно целое число N (количество чисел во второй строке), вторая строка содержит N целых чисел (каждое от -10^5 до 10^5).</p> <p>В результате работы программы должен быть сформирован выходной файл, который содержит следующие значения:</p> <ol style="list-style-type: none"> 1) максимальный элемент кратный 10 (если нет, то 1000); 2) минимальный элемент кратный 7 (если нет, то -1000); 3) список элементов исходного массива, лежащих между значениями, найденными в задачах 1) и 2).
32	<p>Во входном файле содержится две строки: первая содержит одно целое число N (количество чисел во второй строке), вторая строка содержит N вещественных чисел (каждое от -10^5 до 10^5).</p> <p>В результате работы программы должен быть сформирован выходной файл, который содержит следующие значения:</p> <ol style="list-style-type: none"> 1) максимальный отрицательный элемент, среди меньших -1000 (если таких нет, то -1000); 2) минимальный положительный элемент, среди больших 1000 (если таких нет, то 1000); 3) список элементов исходного массива, лежащих между значениями, найденными в задачах 1) и 2).
33	<p>Во входном файле содержится две строки: первая содержит одно целое число N (количество чисел во второй строке), вторая строка содержит N целых чисел (каждое от -10^5 до 10^5).</p> <p>В результате работы программы должен быть сформирован выходной файл, который содержит следующие значения:</p> <ol style="list-style-type: none"> 1) максимальный четный элемент; 2) минимальный нечетный элемент; 3) список элементов исходного массива, лежащих между значениями, найденными в задачах 1) и 2).
34	<p>Во входном файле содержится две строки: первая содержит одно целое число N (количество чисел во второй строке), вторая строка содержит N вещественных чисел (каждое от -10^5 до 10^5).</p> <p>В результате работы программы должен быть сформирован выходной файл, который содержит следующие значения:</p> <ol style="list-style-type: none"> 1) количество элементов, больших 5000; 2) количество элементов, меньших -3000; 3) список элементов, не превышающих по модулю 1000.
35	<p>Во входном файле содержится две строки: первая содержит одно целое число N (количество чисел во второй строке), вторая строка содержит N целых чисел (каждое от -10^5 до 10^5).</p> <p>В результате работы программы должен быть сформирован выходной файл, который содержит следующие значения:</p> <ol style="list-style-type: none"> 1) среднее арифметическое отрицательных чисел; 2) максимальный элемент; 3) список элементов, кратных 50.

36	<p>Во входном файле содержится две строки: первая содержит одно целое число N (количество чисел во второй строке), вторая строка содержит N вещественных чисел (каждое от -10^5 до 10^5).</p> <p>В результате работы программы должен быть сформирован выходной файл, который содержит следующие значения:</p> <ol style="list-style-type: none"> 1) среднее арифметическое положительных чисел (если таких нет, то 10000); 2) среднее арифметическое отрицательных чисел (если таких нет, то 0); 3) модифицированный массив, в котором все элементы, лежащие между значениями, полученными в задачах 1) и 2) заменены на x, остальные – на y.
37	<p>Во входном файле содержится две строки: первая содержит одно целое число N (количество чисел во второй строке), вторая строка содержит N целых чисел (каждое от -10^5 до 10^5).</p> <p>В результате работы программы должен быть сформирован выходной файл, который содержит следующие значения:</p> <ol style="list-style-type: none"> 1) максимальный четный элемент; 2) минимальный элемент кратный 10; 3) модифицированный массив, в котором все элементы, кратные 5 заменены на 5, четные – на +, остальные – на x.
38	<p>Во входном файле содержится две строки: первая содержит одно целое число N (количество чисел во второй строке), вторая строка содержит N вещественных чисел (каждое от -10^5 до 10^5).</p> <p>В результате работы программы должен быть сформирован выходной файл, который содержит следующие значения:</p> <ol style="list-style-type: none"> 1) максимальный элемент; 2) среднее арифметическое всех элементов; 3) список элементов, не превышающих по модулю 1000.

39	<p>Во входном файле содержится две строки: первая содержит одно целое число N (количество чисел во второй строке), вторая строка содержит N целых чисел (каждое от -10^5 до 10^5).</p> <p>В результате работы программы должен быть сформирован выходной файл, который содержит следующие значения:</p> <ol style="list-style-type: none"> 1) максимальный четный элемент; 2) минимальный нечетный элемент; 3) модифицированный массив, в котором все элементы, кратные 10 заменены на 10, кратные 3 – на 3, остальные – на x.
----	--

40	<p>Во входном файле содержится две строки: первая содержит одно целое число N (количество чисел во второй строке), вторая строка содержит N вещественных чисел (каждое от -10^5 до 10^5).</p> <p>В результате работы программы должен быть сформирован выходной файл, который содержит следующие значения:</p> <ul style="list-style-type: none"> 1) максимальный элемент, не превышающий 1000 (если нет, то 1000); 2) минимальный отрицательный элемент, больший -1000 (если нет, то 0); <p>модифицированный массив, в котором все элементы, лежащие между значениями, полученными в задачах 1) и 2) заменены на x, остальные – на z.</p>
----	--