На сегодняшний день XML является одним из распространенных стандартов документов, который позволяет в удобной форме сохранять сложные по структуре данные. Поэтому разработчики платформы .NET включили в фреймворк широкие возможности для работы с XML.

Прежде чем перейти непосредственно к работе с XML-файлами, сначала рассмотрим, что представляет собой xml-документ и как он может хранить объекты, используемые в программе на c#.

Например, у нас есть следующий класс:

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | class User  {      public string Name { get; set; }      public int Age { get; set; }      public string Company { get; set; }  } |

В программе на C# мы можем создать список объектов класса User:

|  |  |
| --- | --- |
| 1  2  3 | User user1 = new User { Name = "Bill Gates", Age = 48, Company = "Microsoft" };  User user2 = new User { Name = "Larry Page", Age = 42, Company = "Google" };  List<User> users = new List<User> { user1, user2 }; |

Чтобы сохранить список в формате xml мы могли бы использовать следующий xml-файл:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11 | <?xml version="1.0" encoding="utf-8" ?>  <users>    <user name="Bill Gates">      <company>Microsoft</company>      <age>48</age>    </user>    <user name="Larry Page">      <company>Google</company>      <age>48</age>    </user>  </users> |

XML-документ объявляет строка <?xml version="1.0" encoding="utf-8" ?>. Она задает версию (1.0) и кодировку (utf-8) xml. Далее идет собственно содержимое документа.

XML-документ должен иметь один единственный корневой элемент, внутрь которого помещаются все остальные элементы. В данном случае таким элементом является элемент <users>. Внутри корневого элемента <users> задан набор элементов <user>. Вне корневого элемента мы не можем разместить элементы user.

Каждый элемент определяется с помощью открывающего и закрывающего тегов, например, <user> и </user>, внутри которых помещается значение или содержимое элементов. Также элемент может иметь сокращенное объявление: <user /> - в конце элемента помещается слеш.

Элемент может иметь вложенные элементы и атрибуты. В данном случае каждый элемент user имеет два вложенных элемента company и age и атрибут name.

Атрибуты определяются в теле элемента и имеют следующую форму: название="значение". Например, <user name="Bill Gates">, в данном случае атрибут называется name и имеет значение Bill Gates

Внутри простых элементов помещается их значение. Например, <company>Google</company> - элемент company имеет значение Google.

Названия элементов являются регистрозависимыми, поэтому <company> и <COMPANY> будут представлять разные элементы.

Таким образом, весь список Users из кода C# сопоставляется с корневым элементом <users>, каждый объект User - с элементом <user>, а каждое свойство объекта User - с атрибутом или вложенным элементом элемента <user>

Что использовать для свойств - вложенные элементы или атрибуты? Это вопрос предпочтений - мы можем использовать как атрибуты, так и вложенные элементы. Так, в предыдущем примере вполне можно использовать вместо атрибута вложенный элемент:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13 | <?xml version="1.0" encoding="utf-8" ?>  <users>    <user>      <name>Bill Gates</name>      <company>Microsoft</company>      <age>48</age>    </user>    <user>      <name>Larry Page</name>      <company>Google</company>      <age>48</age>    </user>  </users> |

Теперь рассмотрим основные подходы для работы с XML, которые имеются в C#.