1. Что такое сериализация, десериализация?

**Сериализация** представляет процесс преобразования какого-либо объекта в поток байтов. После преобразования мы можем этот поток байтов или записать на диск или сохранить его временно в памяти. А при необходимости можно выполнить обратный процесс - **десериализацию**, то есть получить из потока байтов ранее сохраненный объект.

2. Какие существуют форматы сериализации? Поясните структуру для каждого формата. Какие классы для работы с ними существуют в .NET?

Бинарный, SOAP, xml, JSON

Для каждого формата предусмотрен свой класс: для сериализации в бинарный формат - класс BinaryFormatter, для формата SOAP - класс SoapFormatter, для xml - XmlSerializer, для json - DataContractJsonSerializer.

3. Какие классы существуют в пространстве имен System.Xml?

Много разных.

4. Что такое XPath? Приведите пример.

Язык запросов к элементам XML-документа. В XPath существует семь видов узлов: элемент, атрибут, текст, пространство имён, инструкции обработки, комментарии и узлы документа. XML-документы обрабатываются в виде деревьев узлов. Верхний элемент дерева называется корневым элементом. Посмотрите на следующий документ XML:

5. Какие возможности дает LINQ to Xml. Приведите примеры

Еще один подход к работе с Xml представляет технология LINQ to XML. Вся функциональность LINQ to XML содержится в пространстве именSystem.Xml.Linq. Рассмотрим основные классы этого пространства имен:

* *XAttribute: представляет атрибут xml-элемента*
* *XComment: представляет комментарий*
* *XDocument: представляет весь xml-документ*
* *XElement: представляет отдельный xml-элемент*

Ключевым классом является XElement, который позволяет получать вложенные элементы и управлять ими. Среди его методов можно отметить следующие:

* *Add(): добавляет новый атрибут или элемент*
* *Attributes(): возвращает коллекцию атрибутов для данного элемента*
* *Elements(): возвращает все дочерние элементы данного элемента*
* *Remove(): удаляет данный элемент из родительского объекта*
* *RemoveAll(): удаляет все дочерние элементы и атрибуты у данного элемента*

