

#14 Сериализация

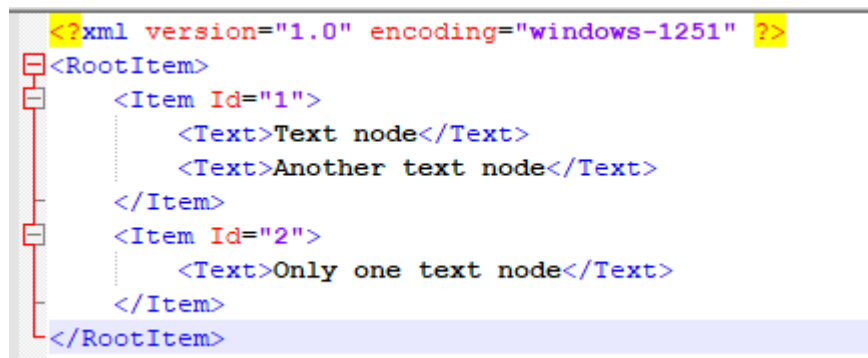
Задание:

1. Ознакомьтесь с базовыми концепциями сериализации/десериализации в .NET:
<https://docs.microsoft.com/en-us/dotnet/standard/serialization/>
2. Ознакомьтесь с 4-мя основными видами сериализации в .NET:
 - a. [JSON](#)
 - b. [Binary](#)
 - c. [XML](#)
3. Создайте класс **CustomSerializer**, который обеспечивает сериализацию и десериализацию любых объектов любых типов всеми вышеперечисленными способами. Интерфейс класса и параметры методов продумайте самостоятельно. Однако, методы **сериализации** должны *сохранять объект по определенному пути*, а методы **десериализации** – *считывать объекты по переданному пути*.
4. Продемонстрируйте работу ваших методов, используя класс из л.р. #10.
5. Запретите сериализацию одного из членов вашего класса и продемонстрируйте отсутствие данного элемента в результате работы сериализаторов.
6. Подключите nuget-пакет **Newtonsoft.JSON**:
<https://www.newtonsoft.com/json>
7. Ознакомьтесь с возможностями работы с JSON используя эту библиотеку и, в частности, возможность сериализации/десериализации, используя методы класса **JsonConvert**.
8. Добавьте в **CustomSerializer** методы **NewtonsoftSerialize** и **NewtonsoftDeserialize**, которые осуществляют сериализацию и десериализацию соответственно используя библиотеку **Newtonsoft**.
9. Познакомьтесь с понятием *LINQ to XML*:
<https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/concepts/linq/linq-to-xml-overview>
10. Продемонстрируйте использование этого языка на данном документе:
<https://raw.githubusercontent.com/demeshchik/oop-example/master/test.xml>

Выведите (по вариантам):

- a. 1, 6, 11, 16 – значение вложенного элемента *text* из всех элементов *dialog*, у которых есть атрибут *init*
 - b. 2, 7, 12, 17 – значение атрибута *id* всех дочерних элементов *phrase* в элементе *dialog*, который обладает *самым большим значением* атрибута *weight*
 - c. 3, 8, 13, 18 – значение атрибута *id* всех элементов *dialog*, у которых есть атрибут *weight*
 - d. 4, 9, 14, 19 – значение дочернего элемента *text* всех элементов *phrase*, которые являются «зависимыми» от элемента *phrase* с заданным значением атрибута *id*
- Note:** значение элемента *next* указывает на *id* элемента *phrase*, который «зависит» от текущего.
- e. 5, 10, 15, 20 – значение дочернего элемента *text* всех элементов *phrase*, у которых нет ни одного дочернего элемента *next*.

11. Используя **LINQ to XML**, реализуйте код создания следующего xml-документа:



```
<?xml version="1.0" encoding="windows-1251" ?>
<RootItem>
  <Item Id="1">
    <Text>Text node</Text>
    <Text>Another text node</Text>
  </Item>
  <Item Id="2">
    <Text>Only one text node</Text>
  </Item>
</RootItem>
```

Вопросы:

1. Что такое сериализация/десериализация?
2. Какие существуют форматы сериализации? Поясните структур каждого файла.
3. Есть ли разница между XML и SOAP? Если да, то в чем она заключается?
4. Для чего служат атрибуты **Serializable** / **NonSerialized**?
5. Для чего предназначен интерфейс **ISerializable**?
6. Для чего предназначена библиотека **Newtonsoft.JSON**?
7. Какие классы (в целом, не только библиотечные) вы использовали для осуществления сериализации/десериализации?