

Сериализация

- Сериализация – это преобразование данных в непрерывный поток байт для записи в файл или передачи по сети или Интернету
- Платформа NET поддерживает автоматическую сериализацию/десериализацию в распространенные форматы (bin, xml, soap, json)
- Для сериализации используются объекты, реализующие интерфейс **IFormatter**.

Методы интерфейса: **Serialize, Deserialize**

Автоматическая сериализация

BIN

- mscorlib.dll
- Namespace System.Runtime.Serialization.Formatters.Binary
- BinaryFormatter

XML

- System.Xml.dll
- Namespace System.Xml.Serialization
- XmlSerializer

SOAP

- System.Runtime.Serialization.Formatters.Soap.dll
- Namespace System.Runtime.Serialization.Formatters.Soap
- SoapFormatter

JSON

- System.Runtime.Serialization.dll
- namespace System.Runtime.Serialization.Json
- DataContractJsonSerializer

Binary-сериализация

Сериализуемый тип, а также вложенные элементы составные элементы типа, должны быть помечены атрибутом **[Serializable]**

Основные действия для сериализации

1. Открываем файловый поток

```
var stream = new FileStream("persons.bin", FileMode.Create);
```

2. Создаем объект сериализации

```
IFormatter fmt = new BinaryFormatter();
```

3. Выполняем запись объекта (например, объект типа List<Person>)

```
fmt.Serialize (stream, lst);
```

4. Закрываем файловый поток

```
stream.Close();
```

Binary-десериализация

Основные действия для десериализации:

1. Открытие файлового потока

```
Stream stream = new FileStream(fileName, FileMode.Open);
```

2. Создание объекта сериализации

```
IFormatter fmt = new BinaryFormatter();
```

3. Десериализация и преобразование к типу

```
list = (List<Person>) fmt.Deserialize(stream);
```

4. Закрытие файлового потока

```
stream.Close();
```

XML-сериализация

- Сериализация осуществляется для открытых полей и свойств
- Сериализуемый объект и вложенные элементы объекта должны быть открытыми, содержать конструктор по умолчанию
- С помощью атрибутов можно управлять структурой XML-документа
[XmlIgnore], [XmlAttribute], [XmlElement]
- Стандартная сериализация поддерживается для примитивных типов, массивов, списков. Но не поддерживается для объектов Dictionary, Tuple, KeyValuePair

Xml-сериализация

1. Открытие файлового потока

```
Stream sr = new FileStream("data.xml", FileMode.Create);
```

2. Создание объекта сериализации с указанием типа данных

```
XmlSerializer xmlSer = new  
    XmlSerializer(typeof(List<Person>));
```

3. Выполнение сериализации

```
xmlSer.Serialize(sr, myList);
```

4. Закрытие файлового потока

```
sr.Close();
```

Xml-десериализация

1. Открытие файлового потока

```
Stream sr = new FileStream("data.xml", FileMode.Open);
```

2. Создание объекта сериализации с указанием типа данных

```
XmlSerializer xmlSer = new  
    XmlSerializer(typeof(List<Person>));
```

3. Выполнение десериализации и приведение типа

```
List<Person> myList = (List<Person>)xmlSer.Deserialize(sr);
```

4. Закрытие файлового потока

```
sr.Close();
```

XML

XmlReader
XmlWriter

XmlSerializer

LINQ to XML

Linq to XML

```
XElement x = new XElement("Faculty",  
    from st in f.Students select new XElement("Student",  
        new XAttribute("ID", st.ID),  
        new XElement("LastName", st.LastName),  
        new XElement("FirstName", st.FirstName)),  
    from t in f.Teachers  
        select new XElement("Teacher",  
            new XElement("Name", t.Name),  
            new XElement("Subject", t.Subject)  
        )  
);  
x.Save(file);
```