**1. Что такое сериализация, десериализация?**

**Сериализация** – это процесс преобразования объекта в формат, который может быть сохранен или передан (например, в файл, поток или по сети). Основная цель – сохранить состояние объекта для последующего восстановления.

**Десериализация** – это обратный процесс восстановления объекта из его сериализованного представления.

**Пример:**

using System;

using System.IO;

using System.Runtime.Serialization.Formatters.Binary;

[Serializable]

public class Person

{

public string Name { get; set; }

public int Age { get; set; }

}

class Program

{

static void Main()

{

Person person = new Person { Name = "John", Age = 30 };

BinaryFormatter formatter = new BinaryFormatter();

using (FileStream stream = new FileStream("person.dat", FileMode.Create))

{

formatter.Serialize(stream, person);

}

using (FileStream stream = new FileStream("person.dat", FileMode.Open))

{

Person deserializedPerson = (Person)formatter.Deserialize(stream);

Console.WriteLine($"Name: {deserializedPerson.Name}, Age: {deserializedPerson.Age}");

}

}

}

**2. Форматы сериализации и классы .NET**

1. **Binary (бинарная сериализация)**
   * Формат: Двоичный
   * Структура: Уплотненные байты, включающие метаинформацию о типе и данные объекта.
   * Класс: BinaryFormatter (устарел, рекомендуется System.Text.Json или XmlSerializer).
2. **XML (сериализация в XML)**
   * Формат: XML-документ
   * Структура: Элементы и атрибуты, представляющие свойства объекта.
   * Класс: XmlSerializer
3. **JSON (сериализация в JSON)**
   * Формат: Текстовый (JavaScript Object Notation)
   * Структура: Пары ключ-значение.
   * Класс: JsonSerializer (из System.Text.Json)
4. **SOAP (Simple Object Access Protocol)**
   * Формат: XML с дополнительными правилами.
   * Структура: XML-документ с пространствами имен.
   * Класс: SoapFormatter (устарел).

**3. Классы в пространстве имен System.Xml**

* XmlDocument – предоставляет DOM-модель для работы с XML.
* XmlReader – потоковое чтение XML.
* XmlWriter – потоковая запись XML.
* XmlSerializer – сериализация и десериализация объектов в/из XML.

**4. Атрибуты для настройки XML сериализации**

* [XmlRoot] – задает корневой элемент XML.
* [XmlElement] – задает имя элемента.
* [XmlAttribute] – сериализует свойство как атрибут XML.
* [XmlIgnore] – исключает свойство из сериализации.

**Пример:**

public class Book

{

[XmlAttribute("title")]

public string Title { get; set; }

[XmlElement("author")]

public string Author { get; set; }

[XmlIgnore]

public int InternalId { get; set; }

}

**5. Отличия BinaryFormatter и SoapFormatter**

* **BinaryFormatter**
  + Формат: Бинарный
  + Быстрее, меньше по размеру.
  + Платформозависимый.
* **SoapFormatter**
  + Формат: XML (SOAP)
  + Читаемый, кроссплатформенный.
  + Медленнее и занимает больше места.

**6. Сериализация контрактов данных**

**Контракт данных** – это соглашение о структуре сериализуемого типа, используемое в WCF (Windows Communication Foundation).

* [DataContract] – отмечает класс как сериализуемый.
* [DataMember] – отмечает свойства или поля для сериализации.

**Пример:**

[DataContract]

public class Order

{

[DataMember]

public int OrderId { get; set; }

[DataMember]

public string CustomerName { get; set; }

}

**7. Атрибуты [OnSerializing], [OnSerialized], [OnDeserializing], [OnDeserialized]**

* [OnSerializing] – вызывается перед сериализацией.
* [OnSerialized] – вызывается после сериализации.
* [OnDeserializing] – вызывается перед десериализацией.
* [OnDeserialized] – вызывается после десериализации.

**Пример:**

[Serializable]

public class Product

{

public string Name { get; set; }

[OnDeserialized]

private void OnDeserializedMethod(StreamingContext context)

{

Console.WriteLine("Object deserialized");

}

}

**8. XPath и пример**

**XPath** – язык запросов для поиска информации в XML-документе.

**Пример:**

XmlDocument doc = new XmlDocument();

doc.Load("books.xml");

XmlNode node = doc.SelectSingleNode("/catalog/book[1]/title");

Console.WriteLine(node.InnerText);

**9. Возможности LINQ to XML и примеры**

**LINQ to XML** – технология для работы с XML с помощью LINQ-запросов.

**Пример:**

XDocument doc = XDocument.Load("books.xml");

var titles = from book in doc.Descendants("book")

select book.Element("title").Value;

foreach (var title in titles)

{

Console.WriteLine(title);

}