**Student.cs**

using System.Xml.Serialization;

using System.Xml;

using System.Xml.Linq;

using System.Runtime.Serialization.Formatters.Binary;

using System.Runtime.Serialization.Formatters.Soap;

using Newtonsoft.Json;

//using System.Runtime.Serialization.Json;

namespace ConsoleApp1

{

[Serializable]

public class Student: Person

{

public bool pass;

public Student(bool \_pass,string \_name,int \_age):base(\_name,\_age)

{

pass = \_pass;

}

public Student()

{ }}}

**Person.cs**

using System.Xml.Serialization;

using System.Xml;

using System.Xml.Linq;

using System.Runtime.Serialization.Formatters.Binary;

using System.Runtime.Serialization.Formatters.Soap;

using Newtonsoft.Json;

//using System.Runtime.Serialization.Json;

namespace ConsoleApp1

{

[Serializable]

public class Person

{

public string name;

public int age;

public Person(string \_name, int \_age)

{

name = \_name;

age = \_age;

}

public Person()

{ }}}

using System.Runtime.Serialization.Formatters.Binary;

using System.Runtime.Serialization.Formatters.Soap;

//using Newtonsoft.Json;

//using Newtonsoft.Json.Linq;

using System.Xml.Serialization;

using System.Xml;

using System.Xml.Linq;

using System.Runtime.Serialization.Json;

using System.Collections.Generic;

using System.Linq;

using System.Text.Json.Serialization;

using System.Text.Json;

namespace ConsoleApp1

{

class Program

{

static void Main(string[] args)

{

//Serialization/Deserialization

Student student1 = new Student(true,"Nastia",18);

Student student2 = new Student(false, "Maks", 20);

Student student3 = new Student(true, "Katia", 19);

Student[] massiv = new Student[] { student1, student2, student3 };

//Binary

BinaryFormatter binaryFormatter = new BinaryFormatter();

using (FileStream fileStream = new FileStream("binary.dat",FileMode.OpenOrCreate))

{

binaryFormatter.Serialize(fileStream,student1);

Console.WriteLine("Binary Serialization");

}

using(FileStream fileStream = new FileStream("binary.dat", FileMode.OpenOrCreate))

{

Student newStudent= (Student)binaryFormatter.Deserialize(fileStream);

Console.WriteLine($"{newStudent.name} - {newStudent.pass} ");

Console.WriteLine("Binary Deserialization");

}

//Soap

SoapFormatter soapFormatter = new SoapFormatter();

using (FileStream filestream = new FileStream("soap.dat", FileMode.OpenOrCreate))

{

soapFormatter.Serialize(filestream,student2);

Console.WriteLine("Soap Serialization");

}

using (FileStream fileStream = new FileStream("soap.dat",FileMode.OpenOrCreate))

{

Student newStudent2 = (Student)soapFormatter.Deserialize(fileStream);

Console.WriteLine($"{newStudent2.name} - {newStudent2.pass} ");

Console.WriteLine("Soap Deserialization");

}

//Xml

XmlSerializer xmlSerializer = new XmlSerializer(typeof(Student));

using (FileStream fileStream = new FileStream("Xml.xml",FileMode.OpenOrCreate))

{

xmlSerializer.Serialize(fileStream,student3);

Console.WriteLine("Xml Serialization");

}

using (FileStream fileStream = new FileStream("Xml.xml", FileMode.OpenOrCreate))

{

Student newStudent3 = (Student)xmlSerializer.Deserialize(fileStream);

Console.WriteLine($"{newStudent3.name} - {newStudent3.pass} ");

Console.WriteLine("Xml Deserialization");

}

//Json

DataContractJsonSerializer json = new DataContractJsonSerializer(typeof(Student));

using (Stream fileStream = new FileStream("Json.json", FileMode.Create, FileAccess.Write, FileShare.None))

{

json.WriteObject(fileStream, student2);

Console.WriteLine("Json Serialization");

}

using (Stream fileStream = File.OpenRead("Json.json"))

{

Student studentUsing = (Student)json.ReadObject(fileStream);

Console.WriteLine(studentUsing.ToString());

Console.WriteLine("Json Deserialization");

}

using (Stream stream = new FileStream("newJson.json", FileMode.Create, FileAccess.Write, FileShare.None))

{

foreach(var i in massiv)

{

json.WriteObject(stream,i);

}

Console.WriteLine("json");

}

BinaryFormatter formatter = new BinaryFormatter();

using (FileStream fs = new FileStream("people.dat", FileMode.OpenOrCreate))

{

formatter.Serialize(fs, massiv);

Console.WriteLine("Binary massiv Serialization");

}

using (FileStream fs = new FileStream("people.dat", FileMode.OpenOrCreate))

{

Student[] deserilizePeople = (Student[])formatter.Deserialize(fs);

foreach (Person p in deserilizePeople)

{

Console.WriteLine($"Name: {p.name} -Age: {p.age}");

}

Console.WriteLine("Binary massiv Desrialization");

}

//3

XmlDocument xmlDocument = new XmlDocument();

xmlDocument.Load("myCollection.xml");

XmlElement xRoot = xmlDocument.DocumentElement;

XmlNodeList nodes = xRoot.SelectNodes("\*");

foreach(XmlNode node in nodes)

{

var NameNode = node.SelectSingleNode("name");

Console.WriteLine(NameNode.InnerXml);

}

foreach (XmlNode node in nodes)

{

var NameNode = node.SelectSingleNode("age");

Console.WriteLine(NameNode.InnerXml);

}

//4

XDocument xdoc = XDocument.Load("myCollection.xml");

var items = from xe in xdoc.Element("ArrayOfStudent").Elements("Student")

where xe.Element("age").Value == "18"

select new Student

{

name = xe.Element("name").Value,

age = Convert.ToInt32(xe.Element("age").Value)

};

foreach (var item in items)

Console.WriteLine($"{item.name} - {item.age}");}}}