using System;

using System.Collections.Generic;

using System.IO;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Xml;

using System.Xml.Serialization;

using System.Runtime.Serialization.Formatters.Binary;

//using System.Runtime.Serialization.Formatters.Soap;

//using System.Runtime.Serialization.Json;

namespace Practice\_Exam\_2

{

public interface IComparable

{

int CompareTo(User user1, User user2);

}

[Serializable]

public class User:IComparable

{

//private string mail { get; set; }

public string mail { get; set; }

public int password { get; set; }

public string status;

public string Status

{

get

{

return status;

}

set

{

if (value == "signin" || value == "signout")

{

status = value;

}

else

{

throw new Exception();

}

}

}

public int CompareTo(User user1, User user2)

{

if (user1.mail.Length == user2.mail.Length)

{

return 0;

}

else if (user1.mail.Length > user2.mail.Length)

{

return 1;

}

else

{

return -1;

}

}

public User(string mail, int password,string st)

{

this.mail = mail;

this.password = password;

this.Status = st;

}

public User()

{

}

public override string ToString()

{

return ("мэйл обьекта " + mail + "пароль обьекта " + password + "статус " + status);

}

public override int GetHashCode()

{

return (password\*2);

}

public override bool Equals(object obj)

{

if (obj == null)

{

return false;

}

User m = obj as User;

if (m as User == null)

{

return false;

}

else

{

return m.mail == this.mail && m.password == this.password && m.status == this.status;

}

}

}

[Serializable]

public class WedNet<T> where T : User

{

LinkedList<T> linkedlist = new LinkedList<T>();

public void Add(T obj)

{

linkedlist.AddFirst(obj);

}

public void Delete(T obj)

{

linkedlist.Remove(obj);

}

}

class Program

{

static void Main(string[] args)

{

try

{

User user1 = new User("qazxsw@mail.ru", 1111, "signisdn");

User user2 = new User("qazxsw@bk.com", 2222, "signout");

Console.WriteLine(user1.ToString());

Console.WriteLine(user2.ToString());

Console.WriteLine(user1.Equals(user1));

Console.WriteLine(user1.Equals(user2));

Console.WriteLine(user1.GetHashCode());

Console.WriteLine(user2.CompareTo(user1, user2));

Console.WriteLine("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

WedNet<User> users = new WedNet<User>();

users.Add(user1);

users.Add(user2);

Console.WriteLine("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

var binFormatted = new BinaryFormatter();

using (var file = new FileStream("file.bin", FileMode.OpenOrCreate))

{

binFormatted.Serialize(file, users);

}

//var soapFormatted = new SoapFormatter();

//using (var file = new FileStream("file.soap", FileMode.OpenOrCreate))

//{

// soapFormatted.Serialize(file, users);

//}

Console.WriteLine("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

var XMLFormatted = new XmlSerializer(typeof(WedNet<User>));

using (var file = new FileStream("file.xml", FileMode.OpenOrCreate))

{

XMLFormatted.Serialize(file, users);

Console.WriteLine("Объект сериализован");

}

Console.WriteLine("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

//var jsonFormatted = new DataContractJsonSerializer(typeof(List<User>));

//using (var file = new FileStream("file.json", FileMode.Create))

//{

// jsonFormatted.WriteObject(file, users);

//}

//using (var file = new FileStream("file.json", FileMode.OpenOrCreate))

//{

// var list = jsonFormatted.ReadObject(file) as List<User>;

// foreach (User i in list)

// {

// Console.WriteLine(i);

// }

//}

}

catch (Exception e)

{

Console.WriteLine("{0} Exception caught.", e);

}

}

}

}