

SERIALIZACIÓN DE OBJETOS

Java ha añadido una interesante faceta al lenguaje denominada serialización de objetos que permite convertir cualquier objeto cuya clase implemente el interface *Serializable* en una secuencia de bytes que pueden ser posteriormente leídos para restaurar el objeto original. Esta característica se mantiene incluso a través de la red, por lo que podemos crear un objeto en un ordenador que corra bajo Windows 95/98, serializarlo y enviarlo a través de la red a una estación de trabajo que corra bajo UNIX donde será correctamente reconstruido. No tenemos que preocuparnos, en absoluto, de las diferentes representaciones de datos en los distintos ordenadores.

La serialización es una característica añadida al lenguaje Java para dar soporte a

- La invocación remota de objetos (RMI)
- La persistencia

La invocación remota de objetos permite a los objetos que viven en otros ordenadores comportarse como si vivieran en nuestra propia máquina. La serialización es necesaria para transportar los argumentos y los valores de retorno.

El interface *Serializable*

Un objeto se puede serializar si implementa el [interface](#) *Serializable*. Este interface no declara ninguna función miembro, se trata de un interface vacío.

```
import java.io.*;
public interface Serializable{
}
```

Para hacer una clase serializable simplemente ha de implementar el interface *Serializable*,

Lectura/escritura

Escribir objetos al flujo de salida *ObjectOutputStream* es muy simple y requiere los siguientes pasos:

1. Creamos un objeto de la clase *Lista*

```
Lista lista1= new Lista(new int[]{12, 15, 11, 4, 32});
```

2. Creamos un [flujo de salida a disco](#), pasándole el nombre del archivo en disco o un objeto de la clase *File*.

```
FileOutputStream fileOut=new FileOutputStream("media.obj");
```

3. El flujo de salida *ObjectOutputStream* es el que procesa los datos y se ha de vincular a un objeto *fileOut* de la clase *FileOutputStream*.

```
ObjectOutputStream salida=new ObjectOutputStream(fileOut);
```

o en una sólo línea

```
ObjectOutputStream salida=new ObjectOutputStream(new  
FileOutputStream("media.obj"));
```

4. El método *writeObject* escribe los objetos al flujo de salida y los guarda en un archivo en disco. Por ejemplo, un string y un objeto de la clase *Lista*.

```
salida.writeObject("guardar este string y un objeto\n");  
salida.writeObject(lista1);
```

5. Finalmente, se cierran los flujos

```
salida.close();
```

```
Lista lista1= new Lista(new int[]{12, 15, 11, 4, 32});  
ObjectOutputStream salida=new ObjectOutputStream(new  
FileOutputStream("media.obj"));  
salida.writeObject("guardar este string y un objeto\n");  
salida.writeObject(lista1);  
salida.close();
```

El proceso de lectura es paralelo al proceso de escritura, por lo que leer objetos del flujo de entrada *ObjectInputStream* es muy simple y requiere los siguientes pasos.

1. Creamos un [flujo de entrada a disco](#), pasándole el nombre del archivo en disco o un objeto de la clase [File](#).

```
FileInputStream fileIn=new FileInputStream("media.obj");
```

2. El flujo de entrada *ObjectInputStream* es el que procesa los datos y se ha de vincular a un objeto *fileIn* de la clase *FileInputStream*.

```
ObjectInputStream entrada=new ObjectInputStream(fileIn);
```

o en una sólo línea

```
ObjectInputStream entrada=new ObjectInputStream(new  
FileInputStream("media.obj"));
```

3. El método *readObject* lee los objetos del flujo de entrada, en el mismo orden en el que ha sido escritos. Primero un string y luego, un objeto de la clase *Lista*.

```
String str=(String)entrada.readObject();  
Lista obj1=(Lista)entrada.readObject();
```

4. Se realizan tareas con dichos objetos, por ejemplo, desde el objeto *obj1* de la clase *Lista* se llama a la función miembro *valorMedio*, para hallar el valor medio del array de datos, o se muestran en la pantalla

```
System.out.println("Valor medio "+obj1.valorMedio());  
System.out.println("-----");  
System.out.println(str+obj1);
```

5. Finalmente, se cierra los flujos

```
entrada.close();
```

```
ObjectInputStream entrada=new ObjectInputStream(new  
FileInputStream("media.obj"));  
String str=(String)entrada.readObject();  
Lista obj1=(Lista)entrada.readObject();  
System.out.println("Valor medio "+obj1.valorMedio());  
System.out.println("-----");  
System.out.println(str+obj1);  
System.out.println("-----");  
entrada.close();
```