

SERIALIZACIÓN

Programación

Facultad de Estadística e Informática

Noviembre 2017

INTRODUCCIÓN

- La serialización de objetos es la obtención de una secuencia de bytes que representa el estado de un objeto
- La secuencia de bytes puede ser utilizada para ser distintos de flujos de entrada y salida, por ejemplo:
 - Ser almacenado o leído en un archivo
 - Ser enviado o recibido por la red (sockets)
- La representación del objeto con una secuencia de bytes es utilizada para recomponer el objeto original

ESTADO DE UN OBJETO

- El estado de un objeto es establecida por los valores de sus atributos que lo definen
- Por lo tanto, al ser serializado un objeto, lo que se almacena y recupera son los valores del objeto en memoria
- Si un objeto tiene atributos que son objetos, será necesario serializarlos previamente, siendo un proceso recursivo que requiere la serialización de todo un grafo de objetos, incluyendo meta-datos para su recuperación

EJEMPLO EN JAVA (2/2)

- Para la serialización de un objeto en java se deberá implementar la interfaz `java.io.Serializable`
- Esta interfaz no define ningún método, solo es utilizada para identificar las clases que permiten instanciar objetos que puedan ser convertidos a una secuencia de bytes (serializables) , y reconstruidos posteriormente
- En Java, clases comunes como `String`, o `ArrayList` ya implementan la clase `java.io.Serializable`, de modo que atributos de este tipo ya pueden ser serializados y reconstruidos

EJEMPLO EN JAVA (2/2)

```
• public class Persona implements java.io.Serializable{  
    private String nombre;  
    private int edad;  
    public Persona(String nombre, int edad){  
        this.nombre = nombre;  
        this.edad = edad;  
    }  
}
```

• Si se desea que un atributo no sea serializado se utiliza el modificador **transient**. Por ejemplo, si en la clase Persona no se requiere serializar edad se utilizaría la declaración:

```
...  
    private transient int edad;  
...
```

ALMACENANDO Y RECUPERANDO OBJETOS

```
.....

FileOutputStream fos = new FileOutputStream("archivo.bin");

FileInputStream fis = new FileInputStream("archivo.bin");

ObjectOutputStream out = new ObjectOutputStream(fos);

ObjectInputStream in = new ObjectInputStream(fis);

Persona p1 = new Persona("Luis", 10);

Persona p2 = new Persona("Juan", 11);

// Escribir el objeto en el fichero

out.writeObject(p1);

out.writeObject(p2);

. . .

// Leer el objeto del archivo (con un mismo orden)

p1 = (Persona)in.readObject();

p2 = (Persona)in.readObject();

...
```

ENVIANDO Y RECIBIENDO OBJETOS POR LA RED (CLIENTE)

```
.....  
Socket socketCliente = new Socket("127.0.0.1", 9090);  
ObjectOutputStream out = new ObjectOutputStream(socketCliente.getOutputStream());  
ObjectInputStream in = new ObjectInputStream(socketCliente.getInputStream());  
Persona p1 = new Persona("Luis", 10);  
Persona p2 = new Persona("Juan", 11);  
// Escribir el objeto en la red  
out.writeObject(p1);  
out.writeObject(p2);  
.  
.br/>.  
// Leer el objeto de la red (un objeto de la clase RespuestaPersona que envía el servidor)  
RespuestaPersona rp = (RespuestaPersona)in.readObject();  
...
```