

Serialización y Persistencia de Objetos en Java

Dra. Maricela Claudia Bravo
Contreras

mari_clau_18@hotmail.com

Serialización y Persistencia de Objetos

Programación Orientada a Objetos

2

Serialización de objetos

- ▶ La serialización de un objeto consiste en obtener una secuencia de bytes que represente el estado de dicho objeto.
- ▶ Esta secuencia puede utilizarse de varias maneras:
 - ▶ enviarse a través de la red
 - ▶ guardarse en un archivo para su uso posterior
 - ▶ utilizarse para recomponer el objeto original.

Estado de un Objeto

- ▶ El estado de un objeto está dado por el estado de sus atributos.
- ▶ Serializar un objeto consiste en guardar el estado de sus atributos.
- ▶ Si el objeto a serializar tiene campos que a su vez son objetos, habrá que serializarlos primero.
- ▶ Es un proceso recursivo que implica la serialización de todo un árbol de objetos.
- ▶ También se almacena información relativa a dicho árbol, para poder llevar a cabo la reconstrucción del objeto serializado.

Modificador transient

- ▶ En ocasiones puede interesar que un atributo concreto de un objeto no sea serializado.
- ▶ Esto se puede conseguir utilizando el modificador *transient*, que informa a la *JVM* de que no interesa mantener el valor de ese atributo para serializarlo o hacerlo persistente.

Modificador transient

Ejemplo:

```
public class MiFecha
{
    protected int n;
    protected Date fecha;
    protected transient long s;
    . . .
}
```

En este ejemplo, los atributos *n* y *fecha* serán incluidos en la secuencia de bytes resultante de serializar un objeto de clase *MiFecha*.

El atributo *s* no será incluido, por tener el modificador `transient`.

Interface Serializable

- ▶ Un objeto serializable es un objeto que se puede convertir en una secuencia de bytes.
- ▶ Para que un objeto sea serializable, debe implementar la interfaz `java.io.Serializable`.
- ▶ Esta interfaz no define ningún método. Simplemente se usa para 'marcar' aquellas clases cuyas instancias pueden ser convertidas a secuencias de bytes (y posteriormente reconstruidas).

Interface Serializable

- ▶ Para serializar un objeto no hay más que declarar el objeto como serializable:

```
public class MiClase implements java.io.Serializable
```

- ▶ El sistema de ejecución de Java se encarga de hacer la serialización de forma automática

Ejemplo 1

```
public class Empleado implements java.io.Serializable
{
    public String nombre;
    public String direccion;
    public transient int SSN;
    public int numero;
    public void correo()
    {
        System.out.println("Enviar correo a" + nombre
            + " " + direccion);
    }
}
```

```
public class SerializeDemo
{
    public static void main(String [] args)
    {
        Empleado e = new Empleado();
        e.nombre = "Reyan Ali";
        e.direccion = "Phokka Kuan, Ambehta Peer";
        e.SSN = 11122333;
        e.numero = 101;
        try
        {
            FileOutputStream fileOut =
                new FileOutputStream("empleado.ser");
            ObjectOutputStream out = new ObjectOutputStream(fileOut);
            out.writeObject(e);
            out.close();
            fileOut.close();
            System.out.printf("Datos serializados en /empleado.ser");
        }catch(IOException i)
        {
            i.printStackTrace();
        }
    }
}}
```

```

public class DeserializeDemo{
    public static void main(String [] args)
    {
        Empleado e = null;
        try{
            FileInputStream fileIn = new FileInputStream("empleado.ser");
            ObjectInputStream in = new ObjectInputStream(fileIn);
            e = (Empleado) in.readObject();
            in.close();
            fileIn.close();
        }catch(IOException i)
        {
            i.printStackTrace();
            return;
        }catch(ClassNotFoundException c)
        {
            System.out.println("Clase Empleado no encontrada");
            c.printStackTrace();
            return;
        }
        System.out.println("Nombre: " + e.nombre);
        System.out.println("Direccion: " + e.direccion);
        System.out.println("SSN: " + e.SSN);
        System.out.println("Numero: " + e.numero);
    }
}

```

Tarea

- ▶ Modificar el programa de Serialización para que muestre un menú principal con las opciones de
 1. Deserializar (cargar datos desde un archivo)
 2. Registrar empleado
 3. Eliminar empleado
 4. Listar empleados
 5. Serializar y salir del programa
- ▶ Este programa deberá almacenar la lista de empleados en un arreglo de máximo 30 empleados.