

Serialización de objetos

Java facilita el almacenamiento y transmisión del estado de un objeto mediante un mecanismo conocido con el nombre de serialización.

La **serialización** de un objeto consiste en generar una secuencia de bytes lista para su almacenamiento o transmisión. Después, mediante la **deserialización**, el estado original del objeto se puede reconstruir.

Para que un objeto sea serializable, ha de implementar la interfaz `java.io.Serializable` (que lo único que hace es marcar el objeto como serializable, sin que tengamos que implementar ningún método).

```
import java.io.Serializable;
import java.util.Date;

public class Contacto implements Serializable
{
    private String nombre;
    private String telefono;
    private String email;
    private String direccion;
    private Date nacimiento;
    private int grupo;
    private double deuda;
    ...
}
```

- Para que un objeto sea serializable, todas sus variables de instancia han de ser serializables.
- Todos los tipos primitivos en Java son serializables por defecto (igual que los arrays y otros muchos tipos estándar).

Serialización

```
import java.io.*;

public class ContactoOutput
{
    private FileOutputStream file;
    private ObjectOutputStream output;

    // Abrir el fichero

    public void abrir()
        throws IOException
    {
        file = new FileOutputStream( "clientes.ser" );
        output = new ObjectOutputStream(file);
    }

    // Cerrar el fichero

    public void cerrar()
        throws IOException
    {
        if (output!=null)
            output.close();
    }

    // Escribir en el fichero

    public void escribir (Contacto contacto)
        throws IOException
    {
        if (output!=null)
            output.writeObject(contacto);
    }
}
```

Deserialización

```
import java.io.*;

public class ContactoInput
{
    private FileInputStream file;
    private ObjectInputStream input;

    public void abrir()
        throws IOException
    {
        file = new FileInputStream( "clientes.ser" );
        input = new ObjectInputStream (file);
    }

    public void cerrar()
        throws IOException
    {
        if (input!=null )
            input.close();
    }

    public Contacto leer ()
        throws IOException, ClassNotFoundException
    {
        Contacto contacto = null;

        if (input!=null) {

            try {
                contacto = (Contacto) input.readObject();
            } catch (EOFException eof) {
                // Fin del fichero
            }
        }

        return contacto;
    }
}
```

Uso

Escritura de datos

```
ContactoOutput salida;  
  
salida = new ContactoOutput();  
  
salida.abrir();  
salida.escribir( contacto1 );  
salida.escribir( contacto2 );  
salida.cerrar();
```

Lectura de datos

```
Contacto          contacto;  
ContactoInput    entrada;  
  
entrada = new ContactoInput();  
  
entrada.abrir();  
  
do {  
    contacto = entrada.leer();  
    System.out.println(contacto);  
} while (contacto!=null);  
  
entrada.cerrar();
```

NOTA: El fichero con los objetos serializados `contactos.ser` almacena los datos en un formato propio de Java, por lo que no se puede leer fácilmente con un simple editor de texto (ni editar).