Serializacón

La **serialización** en Java permite convertir un objeto en una secuencia de bytes para guardarlo en un archivo o enviarlo por la red. Luego, se puede reconstruir el objeto original mediante **deserialización**.

Ubicación en Java: `java.io.Serializable`

---

1. Habilitar la Serialización

Para que una clase sea **serializable**, debe implementar la interfaz `Serializable`.

```
import java.io.Serializable;

class Persona implements Serializable {
    private static final long serialVersionUID = 1L; // Identificador único
    String nombre;
    int edad;
    public Persona(String nombre, int edad) {
        this.nombre = nombre;
        this.edad = edad;
    }
}
```

`serialVersionUID`

- Se usa para garantizar que la versión de la clase coincide al deserializar.
- Si no se define manualmente, Java lo genera automáticamente.

---

2. Serializar un Objeto (Guardar en un Archivo)

Usamos `ObjectOutputStream` para escribir el objeto en un archivo.

```
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.ObjectOutputStream;

public class SerializarObjeto {
    public static void main(String[] args) {
        Persona persona = new Persona("Juan", 30);
        try (ObjectOutputStream salida = new ObjectOutputStream(new
FileOutputStream("persona.dat"))) {
```

```
            salida.writeObject(persona);
            System.out.println("Objeto serializado correctamente.");
        } catch (IOException e) {
        e.printStackTrace();
        }
    }
}
```

Explicación:
`FileOutputStream("persona.dat")` → Crea el archivo.
`ObjectOutputStream(salida)` → Permite escribir objetos en el archivo.
`writeObject(persona)` → Guarda el objeto en el archivo.

---

3. Deserializar un Objeto (Leer desde un Archivo)

Usamos `ObjectInputStream` para reconstruir el objeto desde el archivo.

```
import java.io.FileInputStream;
import java.io.IOException;
import java.io.ObjectInputStream;

public class DeserializarObjeto {
    public static void main(String[] args) {
        try (ObjectInputStream entrada = new ObjectInputStream(new
            FileInputStream("persona.dat"))) {
            Persona persona = (Persona) entrada.readObject();
            System.out.println("Nombre: " + persona.nombre);
            System.out.println("Edad: " + persona.edad);
        } catch (IOException | ClassNotFoundException e) {
        e.printStackTrace();
        }
    }
}
```

Explicación:
`FileInputStream("persona.dat")` → Abre el archivo.
`ObjectInputStream(entrada)` → Permite leer objetos.
`readObject()` → Carga el objeto y se castea a `Persona`.

---

4. Evitar Serialización de Campos (`transient`)

Si un campo es **sensible** y no debe serializarse (como una contraseña), usa `transient`.

```
import java.io.Serializable;

class Usuario implements Serializable {
    private static final long serialVersionUID = 1L;
    String nombre;
    transient String password; // No se serializa
    public Usuario(String nombre, String password) {
        this.nombre = nombre;
        this.password = password;
    }
}
```

Al deserializar, `password` será `null` o su valor por defecto.

---

5. Serializar una Lista de Objetos

Podemos serializar una lista completa en un solo archivo.

```
import java.io.*;
import java.util.ArrayList;
import java.util.List;

public class SerializarLista {
    public static void main(String[] args) {
        List<Persona> lista = new ArrayList<>();
        lista.add(new Persona("Ana", 25));
        lista.add(new Persona("Luis", 40));
        try (ObjectOutputStream salida = new ObjectOutputStream(new
            FileOutputStream("lista.dat"))) {
            salida.writeObject(lista);
            System.out.println("Lista serializada correctamente.");
        } catch (IOException e) {
        e.printStackTrace();
        }
    }
}
```

Para deserializar la lista:

```
try (ObjectInputStream entrada = new ObjectInputStream(new
FileInputStream("lista.dat"))) {
```

```
    List<Persona> lista = (List<Persona>) entrada.readObject();
    for (Persona p : lista) {
        System.out.println("Nombre: " + p.nombre + ", Edad: " + p.edad);
    }
} catch (IOException | ClassNotFoundException e) {
e.printStackTrace();
}
```