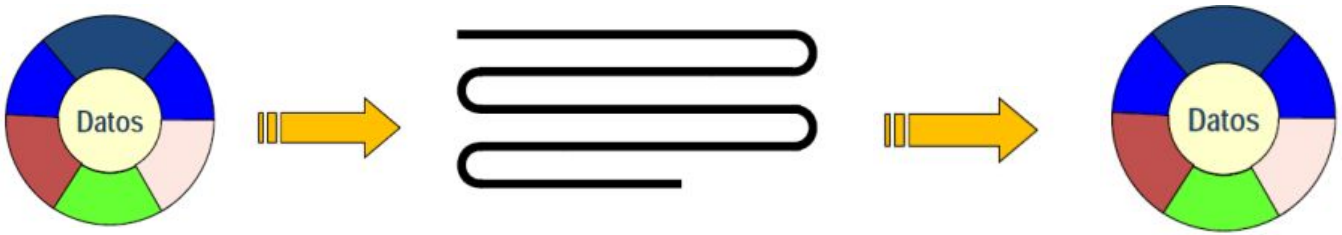
Serialización, persistencia de objetos

Serialización

Se trata de “traducir” el estado de un objeto a una sucesión lineal de bytes.



Permite convertir cualquier objeto que implemente la interfaz Serializable en una secuencia de bytes que pueda ser utilizada posteriormente para reconstruir el objeto original

Principios de la serialización

Esta secuencia de bits puede guardarse en un fichero, o enviarse a otra máquina virtual…

Los objetos mantienen la referencia a otros objetos. Estos deben ser serializables para poder ser almacenados y recuperados para mantener las relaciones originales. En el caso de que no lo sean se lanzará una excepción del tipo NotSerializableException

Para reconstruir un objeto (o conjunto de objetos) serializado, es necesario que la clase (o clases) estén accesibles con el fin de identificarlas y verificarlas antes de restaurar el contenido en una nueva instancia.

Cómo serializar

Un objeto se puede serializar si implementa la intrefaz Serializable. Esta no declara ningún método, se trata de una interfaz vacía.

El motivo es que la serialización de objetos puede suponer una brecha de seguridad, al almacenarse de una manera “legible” valores de campos de un objeto que deberían estar protegidos. Los programadores de Java pusieron esta pequeña “traba” para asegurarse de que el programador que hiciera uso de esta capacidad supiera lo que hacía.

Flujos para entrada y salida de objetos

Las clases para la lectura y escritura de objetos y sus métodos son:

ObjectInputStream y su método Object ObjectInputStream.readObject()

ObjectOutputStream y su método void ObjectOutputStream.writeObject()

Obviamente, para utilizarlos en nuestro programa deberemos realizar los casteos correspondientes, ya que los objetos se devuelven como Object