

Hoja de ejercicios 2 – Ficheros binarios.

SERIALIZACIÓN DE OBJETOS:

Serializar: proceso para que un objeto en memoria se transforme en una estructura que podamos almacenar en un fichero para que los datos sean persistentes.

Des serializar: proceso para que los datos de un fichero pasen a una estructura de objeto en memoria para poder manipularlo con facilidad desde un lenguaje de programación.

Cuando serializamos y desserializamos, es habitual usar estructuras complejas como los ArrayList o los HasMap.

ESTRUCTURAS DE DATOS:

[ArrayList](#).

Implementa el interfaz [List](#).

Es un array dimensionable, válido para todo tipo de datos que también admite el valor null. Tiene un método que permite modificar el tamaño de este.

Si pensamos en acceso concurrente de hilo, esta estructura de datos no está sincronizada.

Si necesito una estructura sincronizada, puedo usar la clase [Vector](#), que implementa un array de objetos accesible con un índice.

EJERCICIOS:

Vamos a trabajar con la clase Persona.

1. Crea un método para serializar un objeto de la clase Persona. Se trata de escribir en un fichero los datos de un objeto de tipo Producto.
2. Crea un método para leer un objeto de tipo Persona del fichero creado anteriormente.
3. Crea un método que escriba en un fichero binario un ArrayList de objetos de tipo Persona.
4. Crea un método que vuelque en un ArrayList los objetos almacenados en el fichero anterior.