

Acceso a datos.

UT 1. Manejo de ficheros.

Práctica serialización.

Esta práctica no es entregable, si recomendable para poder hacer las entregables.

Nos han encargado crear una aplicación (realizada en el módulo PMDM) para el juego T-Rex de Google. Nuestro equipo tiene como objetivo, definir la estructura de las clases del juego y serializar y deserializar los mismos.

Se ha de poder serializar y deserializar en:

- JSON. Se recomienda la librería Gson.
- XML. Se recomienda usar JAXB.
- Binario.

El juego posee una lista de niveles, cada nivel posee:

- Tiempo del juego.
- Velocidad.
- Lista de cactus.
- Lista de pájaros.
- Música.

A su vez, un cactus posee:

- Coordenada.
- Tamaño.
- Nombre.

Y un pájaro:

- Tamaño.
- Nombre.
- Velocidad.
- Coordenada inicial.

Crear un proyecto con Gradle y añadir las librerías de JavaFX, las instrucciones se encuentran en <https://openjfx.io/openjfx-docs/#gradle>

Parte 1.

Definir la clase Size, con dos atributos:

- Width
- Heigh.

Definir la clase cactus y pájaro. No se usa herencia. Para las coordenadas usar la clase Point2D de JavaFX.

Parte 2.

Crear un método estático en el cactus y en pájaro para guardar, se le pasa el nombre del fichero a guardar, un cactus o pájaro y la extensión, que ha de ser xml, json o bin, en caso contrario ha de saltar una excepción.

Definir varios casos de prueba para probar la serialización de pájaros y cactus.

Ver los ficheros generados en binario, xml y json.

Parte 3.

Modificar el código anterior para hacer que no se almacenen los nombre del captus y el pájaro. Se han definir adaptadores para Gson y para JAXB.

Información de adaptadores en Gson:

<https://javadoc.io/doc/com.google.code.gson/gson/latest/com.google.gson/com/google/gson/TypeAdapter.html>

Información de adaptadores en JAXB.

https://eclipse.dev/eclipselink/documentation/2.6/moxy/advanced_concepts006.htm

Comprobar que realmente no se guarda el nombre.

Parte 4.

Crear el método estático load, para el cactus y el pájaro que recibe el nombre del fichero y la extensión y devuelve un cactus para la clase captus y un pájaro para la clase pájaro.

Crear casos de prueba para probar el código anterior.

Parte 5.

Crear ahora la clase juego con una lista de niveles.

La clase juego ha de tener los métodos estáticos load y save, de igual forma que el cactus y el pájaro (estos no se usan en la clase juego).

Definir casos de prueba y probar.



Financiado por
la Unión Europea
NextGenerationEU

