

Problema 8: Leer ficheros de texto

Datos personales	
Apellidos:	
Nombre:	

1 Leer de un fichero de texto

Descripción

Escribir un método estático llamado `leeDatosDeFich` que recibe como parámetro el nombre de un fichero y retorna un array con los 10 primeros números reales contenidos en dicho fichero.

El fichero tiene mezcladas palabras con números. Un ejemplo de fichero a leer podría ser:

```
hola 7.0 3.45 otra 2.1 dos palabras 56.7 -34.5 -4.2 pepe 20.8 -3.5 10.3 lolo -18.70 40.2
```

El método `leeDatosDeFich` deberá ignorar las palabras y meter en el array los primeros 10 números que encuentre. Así, para el caso anterior el método debería retornar el array:

```
{7.0, 3.45, 2.1, 56.7, -34.5, -4.2, 20.8, -3.5, 10.3, -18.70}
```

En el caso de que el fichero contenga menos de 10 números, el método deberá lanzar la excepción `NoSuchElementException` (lanzada automáticamente por los métodos de la clase `Scanner` cuando se trata de leer un dato una vez alcanzado el fin del fichero). Esta excepción es de las no comprobadas, pero es conveniente indicar que se puede lanzar con una cláusula `@throws` en el comentario de documentación.

Respuesta que se pide

<Poner aquí el código Java del método (con su comentario de documentación)>

2 Análisis de un fichero de texto

Descripción:

Escribir un método estático que recibe como parámetro un `String` que contiene el nombre de un fichero donde se encuentra un programa Java en código fuente. El método retorna un booleano indicando si hay llaves desaparejadas o no.

El fichero de texto se leerá línea a línea (métodos `hasNextLine()` y `nextLine()` de la clase `Scanner`). Para cada línea leída se recorrerán todos sus caracteres para analizar solo aquellos que son llaves: '{' o '}'.

Para hacer este método se creará un contador de llaves abiertas, que inicialmente es cero. Cuando durante el recorrido de los caracteres se encuentra una llave izquierda '{' el contador de llaves abiertas se incrementa en uno. Cuando se encuentra una llave derecha '}' el contador de llaves

Programación, Curso 2017-2018

abiertas se decremента. Las llaves están desaparejadas si encontramos una cualquiera de estas dos situaciones:

- al final del método el contador de llaves abiertas no es cero
- en cualquier momento el contador de llaves abiertas es negativo

Respuesta que se pide

<pon aquí el código Java del método (con su comentario de documentación)>

3 Crear una jerarquía de clases (opcional)

Nota: En este problema se evaluarán los ejercicios 1 y 2. Este tercer ejercicio se plantea como opcional, ya que aún no hemos visto la teoría que corresponde. Si te animas a leer los apuntes del capítulo 7 ([Herencia y polimorfismo](#)) y resuelves este tercer ejercicio podrás subir tu nota en este problema hasta 2 puntos sobre 10. Se puede llegar a obtener una nota de 12/10 en este problema, lo que ayudaría a subir la media de la nota de problemas.

Descripción

Se dispone de la clase abstracta descrita en este diagrama:

<i>AtraccionTuristica</i>
-int precio -String nombre -String localidad
+AtraccionTuristica(int precio, String nombre, String localidad) +boolean esGratis() +String getNombre() +String getLocalidad() +String toString()

Se desea extender la clase AtraccionTuristica para crear una jerarquía de clases:

- La clase Museo extiende a AtraccionTuristica y añade un atributo de tipo String que contendrá el tipo de museo (“Arte”, “Marítimo”, “Prehistoria”, etc.). El constructor recibe como parámetros los valores iniciales de todos los atributos (los heredados y el nuevo), invoca al constructor de la superclase y luego da valor al atributo nuevo. La clase proporciona un observador para el nuevo atributo.
- La clase abstracta Naturaleza extiende a AtraccionTuristica y no añade atributos ni métodos. El constructor recibe como parámetros los valores iniciales de todos los atributos heredados e invoca al constructor de la superclase.
- La clase Playa extiende a Naturaleza y añade un atributo de tipo int que indica la longitud de la playa en metros. El constructor recibe como parámetros los valores iniciales de todos los atributos (los heredados y el nuevo), invoca al constructor de la superclase y luego da valor al atributo nuevo. La clase proporciona un observador para el nuevo atributo.

Respuesta que se pide

<Poner una captura de pantalla del diagrama de clase esquemático que muestra el bluej>

Programación, Curso 2017-2018

<Poner aquí el código Java de las clases Museo, Naturaleza y Playa>