Nombre: Álvaro Giménez Gorrís

Login: agimenezgor

Ejercicio 1 – Haciendo una clase en Java (6 puntos)

f) **Teoría**: en la clase `TankCheck` (i.e. fichero `TankCheck.java`) descomenta la línea 65:

```
tank.height = 30.31;
```

Verás que Eclipse compila en tiempo real y te da un mensaje de error. ¿Por qué?

Porque el atributo `Height` está declarado como privado, por lo que no se puede acceder a el desde fuera de la clase `Tank`, ya que es una restricción de acceso impuesta por el código de la clase.

g) **Teoría**: ¿por qué crees que es más adecuado utilizar los métodos de tipo *setter* dentro del constructor en vez de acceder directamente a los diferentes atributos para asignarles los valores?

Porque los métodos setter analizan las precondiciones que debe de cumplir el atributo en cuestión antes de asignarle el valor de entrada. De esta manera, nos aseguramos la escalabilidad y el correcto mantenimiento del código.

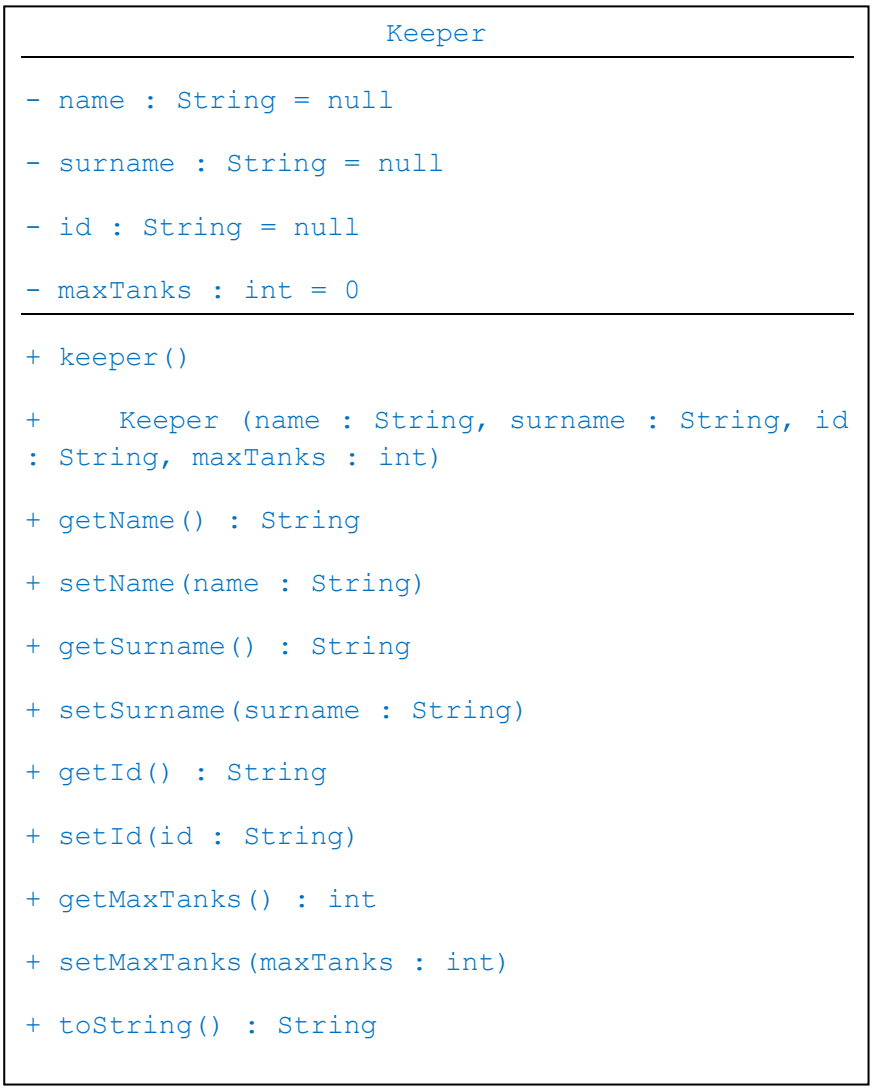
i) **Teoría**: dibuja el diagrama UML de la clase `Tank` una vez introducidos todos los cambios pedidos hasta ahora en este ejercicio.

```
Tank
-------------------------------------------------------------------
- name : String = "Default"
- description : String = "Tank Description"
- length : double = 50.25
- height : double = 10.55
- width : double = 100.232
- imageBackground : String = "./"
- temperature : double = 15
- ph : int = 7
-------------------------------------------------------------------
+ tank()
+ tank(name : String, description : String, length : double, height : double, width : double, imageBackground : String, temperature : double, ph : int)
+ getName(): String
+ setName(name : String)
+ getDescription(): String
+ setDescription(description : String)
+ getImageBackground() : String
+ setImageBackground(imageBackground : String)
+ getLength() : double
+ setLength(length : double) : boolean
+ getHeight() : double
+ setHeight(height : double) : boolean
+ getWidth() : double
+ getWidth(width : double) : boolean
+ getTemperature() : double
+ setTemperature(temperature : double)
+ getPh() : int
+ setPh(ph : int) : boolean
```

Para responder el resto de los apartados de este ejercicio debes incluir en el fichero `loginUOC_PRAC1.zip` el proyecto `PRAC1_ex1` con los ficheros que se te piden en el enunciado.

Ejercicio 4 – Diseño orientado a objetos (2 puntos)

a) Hacer el diagrama de clases UML de la clase `Keeper`, indicando el nivel de acceso/visibilidad de los atributos y métodos, así como los tipos de los atributos y su valor de inicialización (si se indica en el enunciado), los tipos de los parámetros de los métodos y de los valores de retorno de los métodos.



b) Explicación de las decisiones tomadas en el diseño de la clase `Keeper` en el apartado (a).

Declaramos los atributos como privados y los métodos como públicos. En este caso, no ha hecho falta declarar ningún método de tipo privado.

Declaramos los atributos con el valor inicial `Null` para los valores de tipo `String` y `0` para los valores de tipo `int`.

Declaramos los métodos constructores de la clase:

- **Uno con el nombre de la clase y que no contiene ningún tipo de entrada para que, al instanciar el objeto lo inicie a `null` y a `0`.**
- **Sobrecargamos el método anterior, pero, en este caso, le otorgamos tantos argumentos como atributos tiene la clase, para poder iniciar el objeto con los atributos deseados.**

Incluimos todos los métodos `getter` y `setter` de los atributos de la clase:

- **Los métodos `getter` sin argumentos de entrada y que retornan el valor del atributo (`String` o `int` en este caso).**
- **Los métodos setter con el argumento necesario en cada caso (`String` o `int` en este caso) y si retornar nada.**