

## PROGRAMACIÓN ORIENTADA A OBJETOS II

## Examen parcial 2

<b>Curso:</b>	Segundo	<b>Curso académico:</b>	2020/2021
<b>Fecha examen:</b>	05/05/21	<b>Duración examen:</b>	100 minutos
<b>Convocatoria:</b>	Primera	<b>Material permitido:</b>	TODO menos RTC.
<b>Instrucciones:</b>	Crear un proyecto llamado nombreApellidoParcial2. Utilizar un paquete con vuestro nombre y almacenar cada pregunta en un subpaquete diferente. Para entregarlo exportar el proyecto desde Eclipse como un .zip y subirlo a la PDU.		

- (2.5 puntos)** Crea una serie de clases capaz de almacenar la información relacionada con un Huerto Urbano compartido. En ellos, diferentes personas alquilan parcelas para poder cultivar sus propias plantas:
  - 1.1. Crea la clase Cultivo que contenga un nombre de cultivo, unas necesidades de agua (alta, media, baja) y una cantidad de plantas de ese cultivo.
  - 1.2. Crea la clase Parcela, que contenga un tamaño (en metros cuadrados), un Cliente a cargo de la Parcela y una colección de Cultivos. Haz un método añadir Cultivo que reciba como parámetro un Cultivo y lo añada a la colección.
  - 1.3. Crea la clase HuertoUrbano, que contenga un tamaño (en metros cuadrados) y una colección de Parcelas. Haz un método añadir parcela que reciba como parámetro una Parcela y la añada a la colección.
  - 1.4. Crea la clase Cliente, que contenga lo necesario para representar a un usuario del huerto urbano: nombre, apellido, teléfono, id de cliente. Haz que los clientes reciban ids consecutivos (1,2,3,...) conforme son creados y asegurando que nunca se repite ninguno.
- (2.5 puntos)** Comprueba que las clases anteriores funcionan, creando una clase HelperHuerto que contenga una función estática createHuerto que sea capaz de crear y poblar un objeto de tipo HuertoUrbano. La función debe recibir como parámetro de entrada el tamaño del huerto, el número de parcelas y un **Set** con nombres de plantas. Todas las parcelas serán del mismo tamaño, y todas las parcelas contendrán un cultivo cuyo nombre sea cada uno de los nombres del set. El resto de los valores se generan de forma aleatoria. Comprueba que funciona creando un huerto (5000 metros, 22 parcelas y un set de nombres de plantas con "Tomate", "Cebolla", "Lechuga", "Zanahoria" y "Albahaca") e imprimiéndolo por pantalla.
- (2.5 puntos)** Crea una clase Persistencia que contenga:
  - 3.1. un método estático toFile que reciba un HuertoUrbano y un nombre de fichero y sea capaz de persistir el HuertoUrbano a un fichero que pueda ser entendido por un humano (No puedes usar un ObjectOutputStream).
  - 3.2. un método estático fromFile que reciba un nombre de fichero (escrito con el método anterior) y sea capaz de leerlo y crear un objeto de tipo HuertoUrbano con la información contenida en el fichero.
  - 3.3. un método estático toJerarquia que reciba como parámetro un HuertoUrbano y genere una serie de ficheros y carpetas: como raíz una carpeta llamada

huerto\_XX (donde XX es el numero de metros del huerto). Dentro de la carpeta huerto, debes crear una carpeta por cada parcela donde el nombre sea parcela\_XX (donde XX corresponde al id del cliente a cargo de la parcela). Dentro de la carpeta parcela debes crear un archivo (vacío) por cada planta de cada cultivo con nombres consecutivos (Cebolla1,Cebolla2,Cebolla3,...).

3.4. un método main desde el que se pruebe el correcto funcionamiento de las funciones anteriores.

4. **(2.5 puntos)** Excepciones:

- 4.1. Crea una excepción checked llamada AguaInsuficienteException y un método en la clase HuertoUrbano llamado regar que reciba como parámetro de entrada un número de litros de agua. Cada cultivo con necesidades de agua altas consume 3 litros por planta, si las necesidades son medias 2 litros de agua por planta y si las necesidades son bajas 1 litro por planta. Cuando termine de regar todas las plantas de todos los cultivos de todas las parcelas del huerto imprimirá por pantalla un mensaje ("He regado todo"). Si se queda sin agua durante el riego haz que lance la excepción que has creado.
- 4.2. Crea una clase PruebaRiego con un método estático regarHuerto que reciba como parámetro de entrada un HuertoUrbano y una cantidad de litros. El método llamará a la función regar del huerto y le pasará el número de litros recibido como parámetro. En caso de que se produzca una Excepción debe capturarse adecuadamente, escribir por pantalla que no ha tenido suficiente agua y volver a llamar a la función regarHuerto incrementando la cantidad de agua en 10 litros. En caso de que no lance ninguna excepción, que imprima por pantalla el número de litros que ha usado para regar el huerto.
- 4.3. Crea una función main, crea un huerto (con el método de la pregunta 2) y llama a la función regarHuerto pasando el huerto y 10 litros de agua como parámetro.

```
No me ha llegado el agua... 1380
No me ha llegado el agua... 1390
No me ha llegado el agua... 1400
No me ha llegado el agua... 1410
No me ha llegado el agua... 1420
No me ha llegado el agua... 1430
No me ha llegado el agua... 1440
No me ha llegado el agua... 1450
No me ha llegado el agua... 1460
He regado todo!
Bienn regado con 1470 litros
```