



1 Preliminares

- Este ejercicio debe estar implementado 24 horas antes de la siguiente clase de laboratorio.
- Importa el proyecto `greenhouse_skeleton`
- Renombra el proyecto dentro del Eclipse (usando Refactor- Rename) como sigue: **apellido1_apellido2_session6_task_greenhouse**, en letras minúsculas y sin acentos
- **Añade el diagrama de clase UML** al directorio raíz del proyecto exportado.

2 Planteamiento del problema

Mantener las plantas sanas requiere el mejor entorno de cultivo posible. Pero mantenerse al tanto de todos los cambios ambientales y estados o fallos del equipo puede ser un desafío.

Por eso, se nos ocurrió la idea de construir **jardín inteligente**, una aplicación para regular la temperatura y la humedad en un invernadero.



Existen sensores de temperatura y de humedad.

Cada sensor de temperatura leerá la temperatura, mientras que cada sensor de humedad leerá la humedad del invernadero de manera periódica. Los sensores proporcionan información para los controladores. Entonces, estos deciden si es necesario abrir o cerrar puertas para mitigar el calor / frío así como actuar en el sistema de riego, para incrementar o decrementar la humedad y asegurar niveles adecuados de temperatura y humedad en el jardín. Además, necesitamos comprobar el estado de los dispositivos electrónicos en el invernadero (puertas y sensores automáticos).

3 Proyecto de partida

Main Crea 10 puertas, 10 sensores de temperatura y un controlador de invernadero. A continuación, arranca el controlador.

GreenhouseController Es un panel de control central que monitoriza cada controlador del invernadero. Por el momento, solo será el controlador de temperatura.

Tiene método `start()` que ejecuta un ciclo para evaluar regularmente las condiciones ambientales.

Una vez que se realiza la lectura de temperatura, se pone a dormir durante unos segundos antes de obtener la siguiente lectura.

TemperatureController. Descrito más abajo

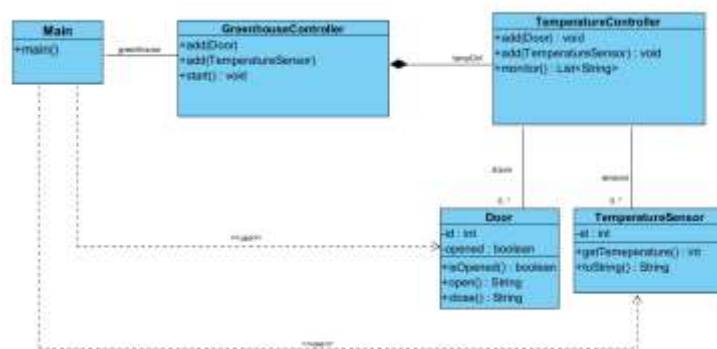


Door Cada puerta está completamente identificada por un número. También hay un campo (boolean opened) que indica si está abierta o no. Los métodos open y close se ejecutarán para abrir o cerrar la puerta respectivamente y cambiarán el valor del campo opened y devolverán un mensaje.

Sensor Class TemperatureSensor tiene un identificador y un método getTemperature que devuelve la temperatura medida por el sensor (valor aleatorio entre 5 y 40 grados).

4 Ejercicio 1: Implementar una versión en funcionamiento del invernadero

Estará formada por varios sensores de temperatura, varias puertas, y un controlador de temperatura.



4.1 Sensores de temperatura

Cada sensor de temperatura puede ser leído invocando el método `getTemperature()` que devuelve la medida de la temperatura en ese momento. Esta operación está simulada devolviendo un número aleatorio entre 5 y 40 grados.

4.2 Controlador de temperatura

Existe una lista de **sensores** de temperatura y una lista de puertas (**actuadores**) que mantendrán la temperatura del invernadero dentro de un rango a partir de la recopilación de datos de los sensores, calculando la temperatura media y, si el valor cae fuera de los límites del rango (mínimo 22 grados, máximo 25 grados) informando al jardinero de cuántas puertas deben ser abiertas o cerradas.

Implementa `TemperatureController::monitor()` de forma que

1. Calcula la temperatura media entre todos los sensores
2. Cuando excede por encima del umbral (hace demasiado calor), el jardinero debe abrir varias puertas para enfriar el ambiente (si es posible).

El mensaje: `Current temp is <t>. It is too hot`



deberá ser devuelto al `GreenhouseController` para informar al jardinero y simular la operación del jardinero invocando al método `door.open()`.

3. Cuando la temperatura media está por debajo del umbral (hace demasiado frío), el jardinero debe cerrar varias puertas para calentar el ambiente (si es posible).

El mensaje `Current temp is <t>. It si too cold`

deberá ser devuelta al `GreenhouseController` para informar al jardinero y simular la operación del jardinero invocando el método `door.close()`.

4. Si la temperatura está dentro del rango seguro, el jardinero es informado, pero no necesita hacer nada. Devuelve al `GreenhouseController` el mensaje `Current temp is <t>. It is right`

4.3 Calentar (o enfriar)

Cuando hace frío, intenta calentar el invernadero cerrando puertas.

- Calcula cuántas puertas deben ser cerradas (El 10% de las puertas existentes por cada grado por debajo del mínimo o fracción) e informa al jardinero enviando al `GreenhouseController` el mensaje `x doors must be closed`
Y por cada puerta abierta que debe ser cerrada (para simular la operación del jardinero) imprime el mensaje `Please, close door x`
- Si no hay suficientes puertas para cerrar, imprime `WARNING: No enough doors could been close. X more doors should be closed.`

5 Ejercicio 2: Otras puertas para el invernadero

Se añadirán 10 nuevas puertas al invernadero. Esta vez, se instalarán puertas automáticas que pueden operar sin intervención humana. Los mensajes serán diferentes:

En lugar de: `Please, open door x`, será: `Automatic door <id> is opening` o en

lugar de: `Please, clase door x`, será: `Automatic door <id> is closing.`

6 Scanner de dispositivos

Para asegurar un correcto funcionamiento de los dispositivos electrónicos, tenemos un scanner que comprueba (checks) si todos están funcionando apropiadamente, reportando los fallos de los sensores dañados y devolviendo un mensaje para informar al jardinero. Esta comprobación también se realiza periódicamente.

6.1 Implementar una versión en funcionamiento de DeviceScanner

`DeviceScanner` está instalado para controlar los dispositivos electrónicos (puertas automáticas y sensores). Contiene:



- Una lista de dispositivos electrónicos
Los dispositivos electrónicos en el invernadero son sensores de temperatura, sensores de humedad, y puertas automáticas. El scanner los comprueba y reporta sus fallos.
- Método `scan()` para chequear el estado del dispositivo y retornar mensajes para informar al jardinero sobre la operación
- `DeviceScanner` debe también ser invocado en `GreenhouseController::start()`
- Si cualquier dispositivo falla, `DeviceScanner::scan` retorna mensajes de advertencia:

```
WARNING: <device toString()> is not in a good condition
```


Donde `toString` debe retornar `[TemperatureSensor] id 0 [AutomaticDoor] id`
- Los dispositivos deben fallar con una probabilidad del 0,5% de las veces (`Random.nextDouble()>0,005`)

7 Control de humedad

El invernadero está también equipado con sensores de humedad, equipamiento de riego y un controlador de humedad para mantener la humedad dentro de una zona de confort. Se debe ser comprobar regularmente la humedad media, como la temperatura, y debe ser mantenida entre el 40% y el 60%.

El controlador de humedad lee valores de los sensores de humedad para calcular la humedad media en el invernadero y ajustar la dosis de riego actuando sobre el sistema de riego.

- Los sensores de humedad devuelven un valor aleatorio entre 0% y 100%
- El sistema de riego está destinado a regular la cantidad de agua que se aplica a la tierra. Tiene cuatro posiciones: OFF, LOW, MEDIUM and HIGH.
- El valor inicial del sistema de riego está OFF.
- Cada vez que se cambia el nivel del sistema de riego, debe devolverse el siguiente mensaje `Irrigator set to OFF/LOW/MEDIUM/HIGH`
- Cada vez que se cambia el nivel del sistema de riego, debe devolverse el siguiente mensaje `Irrigator set to OFF/LOW/MEDIUM/HIGH`
- Para calcular si el nivel de riego debe aumentar o disminuir, debe ser calculada la humedad media.
 - Informar al jardinero de los valores de humedad devolviendo
`Current humidity is <valor aquí>`
 - Si la humedad promedio está en el nivel de confort, no se necesita ninguna acción. Informar al jardinero: `Humidity is right`
 - Si la humedad promedio es demasiado alta (20% por encima del límite superior), entonces, coloca a OFF el sistema de riego e informar al jardinero.
`Current humidity: <valor aquí>`
`It is too damp.`
`Irrigator system is <valor original> and is set to <nuevo valor>`
 - Si la humedad media es demasiado baja (20 % por debajo del límite inferior), se coloca el sistema de riego a HIGH (posición máxima) y se informa al jardinero
`Current humidity:<valor aquí>.`
`It is too dry.`



- Irrigator system is <valor original> and is set to <nuevo valor>
• En otro caso, simplemente sube o baja el sistema de riego un nivel y devuelve los mensajes
Current humidity: <valor aquí>.
It is a bit too <damp/dry>.
Irrigator system is <valor original> and is set to <nuevo valor>

8 Arranque

Crea un invernadero con 10 puertas manuales, 10 puertas automáticas, 10 sensores de temperatura y 10 sensores de humedad. La temperatura y la humedad se controlan periódicamente. El escáner también verifica correctamente el funcionamiento de dispositivos electrónicos con la misma frecuencia.

9 Ejemplo de salida

El siguiente texto corresponde a una ejecución del invernadero. Por favor considera solo el formato, ya que los valores y el orden de las operaciones probablemente serán diferentes.

```
Average temperature is 23.1. It is right
Current humidity is 59,30.
It is right
Average temperature is 17,20. It is too cold
10 doors must be closed
Please, close door 0
Door 0 is closing
Please, close door 1
Door 1 is closing
WARNING. No enough doors could been closed. 6 more doors should be closed
Current humidity is 42,60.
It is right
WARNING: Device [HumiditySensor] 4 is not in a good condition
WARNING: Device [AUTOMATIC DOOR] 6 is not in a good condition
```