

## IIP Primer Parcial - ETSInf

16 de Noviembre de 2015. Duración: 1 hora y 30 minutos.

Se desea representar un termostato que controle la temperatura (medida en grados centígrados) de un dispositivo térmico instalado en una casa. Cada termostato se define en base a cuatro datos: su identificador (nombre de la zona de la casa donde se sitúa), su modo (frío para refrigeración, calor para calefacción), la temperatura actual de la zona y la temperatura de confort que desea el usuario. La WWF (*World Wide Fund for Nature*) recomienda como temperaturas ideales 25° mínimo en verano y 20° máximo en invierno.

1. 7 puntos Se pide implementar la clase **Termostato**, para lo que se debe:

a) (0.5 puntos) Definir los siguientes atributos de clase públicos y constantes de tipo entero:

- **FRIO**, con valor 0 que representa el modo refrigeración;
- **CALOR**, con valor 1 que representa el modo calefacción;
- **T\_IDEAL\_FRIO**, con valor 25 que representa la temperatura mínima recomendada en modo refrigeración;
- **T\_IDEAL\_CALOR**, con valor 20 que representa la temperatura máxima recomendada en modo calefacción.

Estos atributos deberán ser utilizados siempre que se requiera (tanto en la clase **Termostato** como en la clase **GestorTermostato**).

b) (0.5 puntos) Definir los atributos de instancia privados **nombre** (**String**), **modo** (**int**), **tConfort** (**int**) y **tActual** (**double**).

c) (1.5 puntos) Implementar dos constructores:

- Un constructor general con los parámetros apropiados para inicializar todos los atributos de instancia. Suponed que los datos son correctos.
- Un constructor por defecto que crea un **Termostato** en modo **FRIO**, con nombre "zona de estar", temperatura de confort la ideal en refrigeración según WWF y como temperatura actual un valor **double** aleatorio en el intervalo [20.0, 40.0[. Para la generación de este valor aleatorio se deberá utilizar el método **generarAleatorio** que se pide diseñar en el apartado siguiente.

d) (1 punto) Escribir el método privado estático **generarAleatorio** que, dados dos valores **double** x e y, devuelva un número también **double** en el intervalo [x, y[ generado de forma aleatoria.

e) (0.5 puntos) Escribir el método consultor y el método modificador del atributo **modo**. Suponed que el valor del parámetro del modificador es correcto.

f) (1 punto) Escribir el método **equals** (que sobrescribe el de **Object**) para comprobar si un termostato es igual a otro, i.e. si otro es también un termostato y tienen el mismo nombre, mismo modo y temperatura de confort y la diferencia en valor absoluto de temperatura actual es inferior a 1.0 grado.

g) (1 punto) Escribir el método **toString** (que sobrescribe el de **Object**) para que el resultado tenga un formato como el mostrado en los siguientes ejemplos:

"zona de estar, modo refrigeracion, TConfort = 25° y TActual = 27.5°"

"dormitorio, modo calefaccion, TConfort = 20° y TActual = 18.5°".

h) (1 punto) Escribir un método `diferenciaConIdeal()` que devuelva un entero que debe ser:

- 0 si la temperatura de confort es adecuada al modo, es decir, si es mayor o igual que la ideal en modo `FRIJO` o menor o igual en modo `CALOR`,
- la diferencia en valor absoluto entre las temperaturas de confort e ideal, en otro caso.

Por ejemplo:

- Si el termostato está en modo refrigeración y la temperatura de confort es 22°, el resultado del método debe ser 3; si la temperatura de confort fuera 25° o más, el resultado debería ser 0.
- Si el termostato está en modo calefacción y la temperatura de confort es 22°, el resultado del método debe ser 2; sin embargo, si la temperatura de confort fuera 20° o menos, el resultado debería ser 0.

2. 3 puntos Se pide completar el programa `GestorTermostato` para crear un termostato y aconsejar al usuario sobre incrementar o decrementar la temperatura de confort para cumplir con las normas de eficiencia energética de la WWF. Para ello, se debe:

a) (1.5 puntos) Implementar el método con perfil `public static String consejo(Termostato t)` que compruebe la diferencia de temperatura de confort con respecto a la ideal del termostato `t` (método `diferenciaConIdeal`) y devuelva un mensaje en el que se indique si se ha de subir o bajar la temperatura de confort y en cuántos grados o si la temperatura es adecuada.

Por ejemplo:

- Si el termostato está en modo refrigeración y la temperatura de confort es 22° el consejo sería "Grados a aumentar: 3"; si la temperatura de confort fuera 25° o más, el consejo sería "La temperatura es adecuada".
- Si el termostato está en modo calefacción y la temperatura de confort es 22° el consejo sería "Grados a disminuir: 2"; si la temperatura de confort fuera 20° o menos, el consejo sería "La temperatura es adecuada".

b) (1.5 puntos) Completar el método `main` para que, una vez leídos desde teclado los valores de modo y temperaturas con las instrucciones que ya figuran (por simplificar se suponen correctos), realice las siguientes acciones:

- (0.5 puntos) Crear un `Termostato t` para el "dormitorio ppal" con los valores de modo y temperaturas leídos.
- (0.5 puntos) Mostrar por pantalla el termostato creado.
- (0.5 puntos) Mostrar por pantalla el consejo sobre eficiencia energética de este termostato.