UNIVERSIDAD NACIONAL DEL SUR
Departamento de Ciencias e Ingeniería de la Computación
Introducción a la Programación Orientada a Objetos

Segundo Parcial

Tema 1

12 de Octubre 2021

Lea atentamente cada uno de los **4 ejercicios** antes de comenzar a resolverlo.
Recuerde que las soluciones se evalúan en términos de
correctitud, eficiencia y legibilidad.

Para comenzar descargue el paquete de Bluej necesario para el parcial en el sitio:
https://cs.uns.edu.ar/~mmaisonnave/resources/IPOO/SegundoParcial.zip

1. Dada la implementación de `Termostato` y parte de la implementación de `ArregloTermostatos`, complete el código de la clase `ArregloTermostatos` agregando los métodos `copy` y `hayNDesreguladosConsecutivos` de acuerdo al siguiente diagrama:

Termostato
<<Atributos de Instancia>> sensado, panel: entero
<<Constructor>> Termostato(s, p: entero) *<<Comandos>>* establecerSensado(s: entero) establecerPanel(p: entero) copy(t:Termostato) *<<Consultas>>* obtenerSensado(): entero obtenerPanel(): entero regulado():boolean equals(t:Termostato):boolean

ArregloTermostatos
<<Atributos de Instancia>> arreglo:Termostato[]
<<Constructor>> ArregloTermostatos(tamanio:entero) *<<Comandos>>* establecerTermostato(i:entero, t:Termostato) **copy(at: ArregloTermostatos)** *<<Consultas>>* obtenerTermostato(i:entero):Termostato tamanioArreglo ():entero cantTermostatos():entero **hayNDesreguladosConsecutivos(n:entero):boolean**

La clase `ArregloTermostatos` encapsula a un arreglo de tamaño mayor a 1.

- **`copy(at: ArregloTermostatos)`** si **`at`** ligado y ambos arreglos tienen el mismo tamaño, realiza copia superficial, en caso contrario no tiene efecto.

- **`hayNDesreguladosConsecutivos(n:entero):boolean`** devuelve true si el arreglo contiene **exactamente** **`n`** termostatos desregulados en posiciones consecutivas. Requiere **`n`** mayor a cero y todas las posiciones del arreglo ligadas.

2. Implemente una clase tester para verificar el servicio `hayNDesreguladosConsecutivos` con valores fijos para al menos 3 casos de prueba significativos.

3. En el mismo proyecto BlueJ implemente la clase `TermostatosEdificio` de acuerdo al siguiente diagrama:

TermostatosEdificio
<<Atributos de Instancia>> edificio:Termostato[][]
<<Constructor>> TermostatosEdificio(cant_pisos:entero, cant_sectores:entero) *<<Comandos>>* establecerTermostato(p:entero, s:entero, t:Termostato) *<<Consultas>>* cantidadSectores():entero cantidadPisos():entero cantTermostatos(p:entero):entero obtenerTermostato(p:entero,s:entero):Termostato obtenerPisoMayorSensado(): entero entrepisos(p1,p2:entero):boolean pisosEquivalentes(p:entero,te:TermostatosEdificio):boolean
Las filas del atributo edificio representan los pisos de un edificio y las columnas representan los sectores en los que se divide cada piso. Todos los pisos tienen la misma cantidad de sectores. Al menos un sector de cada piso tiene asignado un termostato cuando se ejecuta cualquiera de las consultas. *El piso 1 corresponde a la fila 0 y el primer sector 1 a la columna 0, es decir, desde las clases clientes el primer piso es el 1 y el primer sector es el 1.*

La clase `TermostatosEdificio` encapsula un arreglo de dos dimensiones.
- **`TermostatosEdificio(cant_pisos:entero, cant_sectores:entero)`** crea un arreglo de **`cant_pisos`** filas y **`cant_sectores`** columnas. Requiere ambos parámetros mayores a 4.

- **`establecerTermostato(p:entero, s:entero, t:Termostato)`** si **`p`** y **`s`** son valores válidos, asigna el termostato **`t`** a la posición del arreglo que corresponde al piso y sector.

- **`obtenerTermostato(p:entero,s:entero):Termostato`** devuelve el termostato en el piso **`p`** y sector **`s`**. Requiere que **`p`** y **`s`** sean valores válidos.

- **`obtenerPisoMayorSensado(): entero`** Retorna el piso en el que se registra el mayor valor sensado. Si el mayor valor de sensado se repite en dos o más pisos, retorna el piso más alto en el cual se registra dicho valor.

- **`entrepisos(p1,p2:entero):boolean`** Retorna true si solo si entre los pisos **`p1`** y **`p2`** (incluyendo ambos), todos los termostatos están regulados. Requiere p2 mayor a p1.

- **`pisosEquivalentes(p:entero,te:TermostatosEdificio):boolean`** Retorna true si y solo sí el edificio que recibe el mensaje y el que pasa como parámetro tienen la misma cantidad de sectores y el piso **`p`** del edificio que recibe el mensaje y el que pasa como parámetro tienen termostatos equivalentes en exactamente los mismos sectores. Requiere **`te`** ligado.

- **`cantidadSectores()`** y **`cantidadPisos()`** devuelven la cantidad de sectores y de pisos del edificio.

- **`cantTermostatos(p:entero):entero`** retorna la cantidad de sectores que tienen asignado un termostato en el piso **`p`**.

4. Comprima todo el proyecto de BlueJ en un archivo que incluya sus nombres y apellidos (ejemplo: Juan_Manuel_Perez.zip) y subalo a la plataforma moodle.