

Segundo Parcial

14 de octubre 2022 – 8hs

Las implementaciones se evalúan en términos de
correctitud, eficiencia y legibilidad

Dado el siguiente diagrama que modela las temperaturas mínimas en una región con n estaciones meteorológicas para un período de m días.

TemperaturasRegion
<<Atributo de Instancia>> real [][] temperaturas
<<Constructor>> TemperaturasRegion(n, m : entero)
<<Comandos>> establecerTemp(e : entero, d : entero, t :real)
<<Consultas>> obtenerTemp(e :entero, d :entero):real cantDias():entero cantEstaciones():entero estacionCreciente(e :entero):boolean cantEstacionesCrecientes():entero clone():TemperaturasRegion
Cada fila denota las temperaturas de una estación meteorológica. Cada columna mantiene las temperaturas de un día del período. Desde la clase cliente los días se denotan a partir de 1 y las estaciones a partir de 0.

- **TemperaturasRegion(n, m : entero)** requiere n y m mayores a 1.
- **establecerTemp(e : entero, d : entero, t :real)** requiere $0 \leq e < \text{cantEstaciones}()$ y $1 \leq d \leq \text{cantDias}()$
- **obtenerTemp(e :entero, d :entero):real** requiere $0 \leq e < \text{cantEstaciones}()$ y $1 \leq d \leq \text{cantDias}()$
- **estacionCreciente(e :entero):boolean** requiere $0 \leq e < \text{cantEstaciones}()$ y retorna verdadero si en la estación que corresponde a la fila e las temperaturas están ordenadas en forma creciente.
- **cantEstacionesCrecientes():entero** cuenta la cantidad de estaciones que tienen las temperaturas ordenadas en forma creciente.
- **clone():TemperaturasRegion** retorna un nuevo objeto de clase TemperaturasRegion que es un clone del objeto que recibe el mensaje.

a) Implemente la clase TemperaturasRegion.

b) Descargue el tester del aula virtual de Moodle, complete con dos casos de prueba significativos para verificar el método estacionCreciente, uno que compute false y otro que compute true.

c) Comprima el proyecto y envíenlo a la cuenta ipoo.dic@gmail.com