

Examen final 26 de Enero de 2021	
Grado en Ingeniería Informática Doble Grado en Ingeniería Informática y Administración de Empresas	

LEE **CUIDADOSAMENTE** ESTAS INSTRUCCIONES ANTES DE EMPEZAR EL EXAMEN. EN CASO DE NO SEGUIRLAS SE APLICARÁ UNA **PENALIZACIÓN** MÍNIMA DE 1 PUNTO:

- **Lee** el examen entero antes de empezar a escribir código.
- Solo se responderán preguntas durante los primeros **30 minutos**. Después toma las decisiones que consideres adecuadas y justifícalas en el examen.
- No respondas en esta hoja, utiliza las hojas en blanco.
- Esta hoja no se debe entregar, puedes anotar en ella lo que consideres oportuno.
- No se admitirán exámenes escritos **a lápiz** ni con **bolígrafo rojo**.
- No olvides poner tu **nombre** en todas las páginas.
- Si no resuelves los problemas en orden utiliza una cara/hoja para cada problema y ponlas en el **orden correcto** al final. También puedes usar una hoja para cada clase, pero los métodos deben estar en el orden en que aparecen en el enunciado.
- Puedes usar cualquier **material impreso**. Los dispositivos **electrónicos** están **prohibidos**.
- El tiempo máximo para el examen es de **3 horas**
- Se calificará con **0 puntos** una sección si:
 - Se usan funciones, librerías o partes del lenguaje no vistas durante el curso.
 - Se usa break en un bucle o una función.
 - Se usa cualquier método de lista excepto: append, insert y remove (también se puede usar la función del)
 - Se usa una variable no declarada o que está declarada en otro método o clase.
 - Se usan variables globales.
 - Se alteran los parámetros de los métodos o no se usan los parámetros en el método.
 - Se cambia la cabecera de cualquiera de los métodos propuestos.
 - Se crean atributos no pedidos.

Problema 1 (10 Puntos).- La gran nevada de 2021 en Madrid ha sido la mayor en más de un siglo y ha demostrado que la ciudad no está preparada para este tipo de situaciones. El objetivo de este ejercicio es ayudar al alcalde a afrontar esta situación si vuelve a ocurrir en el futuro, implementando un sistema automático de limpieza de calles con quitanieves. Los conceptos más importantes del problema son los siguientes:

- Una Ciudad tiene Calles y Quitanieves que pueden limpiar los Copos de nieve
- Las Calles pueden ser horizontales o verticales, tienen una coordenada inicial (x,y), una longitud y una temperatura.
- Los Quitanieves se usan para limpiar los Copos de nieve de las calles. Están en una coordenada determinada y tienen una cierta cantidad de gasolina.
- Los Copos de nieve caen en una coordenada determinada (x,y) que debe pertenecer a una calle. Si la temperatura de la calle está por encima de 2 grados se fundirán, si no, se quedarán en la calle.

Considerando esta información, se pide crear un programa como se muestra a continuación:

- a) (1 punto) Crear la clase Copo con atributos temperatura, x e y (temperatura debe ser privado). Crear un método init con los parámetros adecuados. Todos los atributos se deben declarar dentro del init y exclusivamente ahí. Crear propiedades y setters para asegurar que tanto x como y son >= 0. Crear también una propiedad de solo lectura, fundido, que devuelve si un copo de nieve está fundido o no.
- b) (1 punto) Crear la clase Calle con atributos horizontal o vertical, x_inicial, y_inicial, longitud, temperatura y una lista de Copos de nieve que inicialmente estará vacía. No hace

falta crear propiedades. Crear un método `init` con los parámetros adecuados. Todos los atributos se deben declarar dentro del `init` y exclusivamente ahí. Crear también un método `str` que devuelva (para una calle vertical H-`<y_inicial>` se debe cambiar por V-`<x_inicial>`): "Calle H-`<y_inicial>`, va de (`x_inicial`, `y_inicial`) a (`x_final`, `y_final`). Temperatura: `<temperatura>`", donde `<elemento>` debe ser cambiado por su valor. Al final del examen se muestra un ejemplo de la salida.

- c) (0,5 puntos) Crear la clase `Quitanieves` con atributos `x`, `y`, `gasolina`. Al igual que antes todos los atributos deben declararse dentro del `init`. No hacen falta propiedades.
- d) (0,5 puntos) Crear la clase `Ciudad`. Tiene dos atributos para guardar su ancho y su alto que deben ser números enteros mayores que 5 (usar propiedades y setters). También tiene una tupla de objetos `Calle` y otra de objetos `Quitanieves`, ambas inicializadas a `None`. Al igual que antes todos los atributos deben declararse dentro del `init`.
- e) (1 punto) Implementar el método `crear_calles(self)` que crea las calles de una ciudad y rellena la tupla correspondiente. Debe crear `alto` calles horizontales y `ancho` calles verticales. La primera calle horizontal estará en la posición (0,0), la segunda en la (0,1) y así sucesivamente. Las calles verticales estarán en la (0,0), (1,0), etc. Su longitud será un número aleatorio en el rango [5, `ancho`] para las horizontales y [5, `alto`] para las verticales. Su `temperatura` será un valor aleatorio entre [-5, 4]. Especificar en qué clase se encuentra.
- f) (0,5 puntos) Implementar el método `crear_quitanieves(self)` que crea la tupla de `quitanieves` de la ciudad. Se debe crear una `quitanieves` al inicio de cada calle. Su `gasolina` inicial estará en el rango [1, 20]. Especificar en qué clase se encuentra.
- g) (1,5 puntos) Crear el método `nevar(self, numero: int)` que crea `numero` copos de nieve distribuidos aleatoriamente en las calles de la ciudad: su `temperatura` debe ser la de la calle en la que están, sus coordenadas deben ser un valor aleatorio dentro de los límites de la calle. Especificar en qué clase se encuentra.
- h) (1 punto) Crear el método `necesita_limpieza(self) -> bool` que devuelve si una calle necesita ser limpiada. Una calle necesita limpieza si en cualquiera de sus posiciones tiene al menos dos copos de nieve no fundidos. Especificar en qué clase se encuentra.
- i) (0,5 puntos) Implementar el método `calles_a_limpiar(self) -> str` que devuelve una cadena de texto con las calles que se deben limpiar (hay un ejemplo de salida al final del examen). Especificar en qué clase se encuentra.
- j) (2 puntos) Crear el método `pasar_quitanieves(self)` que limpia todas las calles de la ciudad que lo necesiten. Cada `quitanieves` se moverá por su calle paso a paso, eliminando de la lista de copos de nieve los que se encuentren en esa posición. Cada vez que se mueva se decrementará la `gasolina` en una unidad. Si se queda sin `gasolina` dejará de limpiar. Especificar en qué clase se encuentra.
- k) (0,5 puntos) Crear un programa principal con una ciudad de 20x20 y 100 copos de nieve. A continuación se deben limpiar las calles e imprimir las que no se pudo limpiar porque la `quitanieves` se quedó sin `gasolina`. Ejemplo de ejecución:

Calles a limpiar:

Calle H-15, va de (0,15) a (15,15). Temperatura: -1

Calle V-2, va de (2,0) a (2,17). Temperatura: -3

Calle V-3, va de (3,0) a (3,5). Temperatura: -1

Calle V-5, va de (5,0) a (5,16). Temperatura: -4

Calle V-11, va de (11,0) a (11,12). Temperatura: -4

Calles sin limpiar:

Calle H-15, va de (0,15) a (15,15). Temperatura: -1