Agenda anual

Dispones de la implementación de un sistema para gestionar una agenda correspondiente al año 2017. La agenda se compone de semanas en las cuales se pueden registrar citas los días laborables. Las citas tienen una duración (en horas) y una descripción. La clase que se utiliza para diseñar las citas está correctamente implementada. Los desarrolladores de las clases asociadas a los días y las semanas las han implementado pero no las han testeado. Está es tu labor: diseñar un conjunto de tests que permita chequear ambas implementaciones, detectando todos los errores que presentan. Primero debes asegurarte de que la clase DiaBreak es correcta de acuerdo a la especificación y después hacer lo mismo con la clase SemanaBreak. En caso de que detectes errores debes corregir el código hasta conseguir que pase todos los tests diseñados.

Los requisitos de nuestra implementación para la clase DiaBreak son los siguientes:

- Los días se identifican por el número del día del año (1-365).
- Cada día almacena las citas correspondientes a cada hora disponible para tal efecto: primera cita a las 9:00 y última cita a las 17:00 de la tarde.
- Una cita puede tener una duración superior a una hora, por lo que puede estar asignada a varias horas consecutivas de un día.
- La clase presenta varios métodos:

1. `int buscaSlot(int duracion):` devuelve la primera hora disponible para una cita de la duración que recibe como parámetro. En caso de de no encontrar un hueco, devolverá -1.
2. `boolean asignarCita(int hora, Cita cita):` asigna la cita a las horas necesarias para la cita que se recibe como parámetro empezando en la hora indicada si hay hueco libre. Devuelve un boolean que indica si la operación se ha podido realizar.
3. `Cita getCita(int hora):` devuelve la Cita asignada a la hora que se recibe como parámetro. En caso de estar libre esa hora, devuelve null.
4. `String muestraCita(int hora):` devuelve un String que describe la Cita asociada a la la hora indicada como parámetro

    a. "No existe" si no hay Cita asignada a la hora recibida como parámetro.
    b. "Hora no valida" si la hora indicada no es asignable a una cita
    c. "hora:00 Descripción de la Cita"
5. `boolean validaHora(int hora):` indica si la hora recibida como parámetro es asignable para una Cita o no.

6. `public boolean huecoLibre(int hora, int duracion)` indica si esta libre el hueco que abarca desde la hora indicada tantas horas como duración tiene la cita.

Los requisitos de nuestra implementación para la clase SemanaBreak son los siguientes:

- Una semana se identifica por el número de semana del año (0-52).
- Cada semana almacena los días disponibles para asignación de citas (lunes-viernes) esa semana.
- La clase presenta varios métodos:

1. `String mostrarCita(int hora, int diaSemana):` devuelve un String que describe la Cita asociada al día de la semana y la hora indicadas como parámetros

   a. “No existe” si no hay Cita asignada a la hora recibida como parámetro.
   b. “Hora no valida” si la hora indicada no es valida
   c. “hora:00 Descripción de la Cita”

2. `DiaBreak getDia(int diaSemana):` devuelve el Dia correspondiente al día de la semana que se recibe como parámetro. En caso de recibir 6 o 7 (sábado o domingo), devuelve null.

3. `public int getNumeroSemana(): devuelve el número de semana.`

4. `public String primerHueco(int duracion).` Devuelve un String que indica el nombre del primer día de la semana con disponibilidad y la hora de inicio de la cita. En otro caso devuelve "No hay disponibilidad".

5. `public String diaSemana(int dia).` Devuelve el nombre del día de la semana (Lunes, Martes…Viernes). En otro caso muestra “No citable”.