



Ejercicio Obligatorio. Sesión 02

Este ejercicio debe estar implementado **48 horas antes** de la siguiente clase de laboratorio.

Es sumamente importante **seguir las normas de denominación de los proyectos**. No olvides comprobar los nombres antes de entregarlos

Ejercicio 1. Proyecto Analyzer

Cargue el proyecto *nombre_apellido1_apellido2_session02_analyzer* realizado en clase, realice una copia y renombre el mismo como:

nombre_apellido1_apellido2_session02_task_analyzer

Complete las pruebas de los métodos de la clase *WordAnalyzer.java* (que no se hayan completado en el laboratorio).

Ejercicio 2. Proyecto SeatManager

Cree un proyecto *nombre_apellido1_apellido2_session02_task_seatmanager* con una clase **SeatManager** que contenga un array bidimensional de personas (cada persona posee el nombre y la edad). Puede usarse la clase *Person* de la sesión anterior. El tamaño máximo del array es de 10 filas y 4 columnas y se deben implementar las siguientes operaciones:

- a. ***bookSeat***: recibe como parámetro un objeto de tipo *Person*, un número de fila y uno de columna. Coloca al objeto *Person* mencionado en la posición del array identificada por los parámetros fila y columna, si la posición no está ocupada o está ocupada por un adulto no jubilado. El método retorna *true* si sienta con éxito a la persona y *false*, en otro caso.
- b. ***findPassengersByAge***: recibe dos parámetros, edad mínima y edad máxima. Retorna un *ArrayList* con las personas sentadas en las filas pares y cuya edad esté comprendida en el intervalo *[edad mínima, edad máxima]*.

Hay que incluir el control de parámetros, lanzando una excepción cuando no sean válidos. Utilice los métodos de control de *ArgumentsCheck* del proyecto util.

Pruebe los métodos *bookSeat* y *findPassengersByAge* utilizando JUnit, implementando las pruebas de funcionalidad y las pruebas de robustez.

Realice las pruebas partiendo del siguiente estado inicial.:



```
// creamos pasajeros
child1 = new Person("Maria", 2);
adult1 = new Person("Marta", 24);
retired1 = new Person("Jose", 78);
child2 = new Person("Pablo", 7);
adult2 = new Person("Andres", 20);
retired2 = new Person("juan", 65);

childToSeat = new Person("Ines", 12);
adultToSeat = new Person("Elsa", 22);
retiredToSeat = new Person("Ramon", 82);

// se les asigna un asiento
// a 2 niños, 2 adultos y 2 retirados

seatManager.bookSeat(child1, 0, 1);
seatManager.bookSeat(adult1, 1, 3);
seatManager.bookSeat(retired1, 6, 0);
seatManager.bookSeat(child2, 5, 3);
seatManager.bookSeat(adult2, 4, 2);
seatManager.bookSeat(retired2, 9, 3);
```

NOTA. Utilice `@Before` para incluir el estado inicial. Tenga en cuenta que el método `setUp` de `@Before` se ejecuta cada vez que se ejecuta un método de prueba

Ejercicio 3. Proyecto 2048

Renombre el proyecto 2048 implementado en la sesión 1 como:

`nombre_apellido1_apellido2__session02_task_game`

Añada al proyecto 2048 los siguientes métodos:

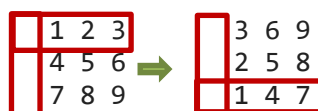
1. Método ***public void moveRight()*** compacta a la derecha los números de cada fila y a continuación suma de dos en dos los números consecutivos iguales dejando el resultado en la casilla derecha. Finalmente, se vuelve a compactar.

Ejemplo: Dada una fila 2 2 0 2 2 el resultado de `moveRight()` sería 0 0 0 4 4

```
2 2 0 2 2 -> compacta a la derecha -> 0 2 2 2 2
0 2 2 2 2 -> suma consecutivos -> 0 0 4 0 4
0 0 4 0 4 -> compacta a la derecha -> 0 0 0 4 4
```

2. Método ***public void moveLeft()*** Análogo a `moveRight` pero a la izquierda. Para realizar esta operación, en lugar de usar el método de mover a la izquierda se deben hacer rotaciones del tablero y se reutiliza siempre el movimiento a la derecha. En este caso sería: rotar, rotar, mover a la derecha, rotar, rotar

Para rotar el tablero, se debe rotar a la izquierda (contrario a las agujas del reloj)



Crea un método privado ***private void rotateBoard()*** para esta operación.

3. Método ***public void moveUp()***. Similar al anterior.
4. Método ***public void moveDown()***. Similar al anterior.



5. **Genere la documentación** con JavaDoc. El código del proyecto debe incluir, comentarios de documentación y comentarios de implementación.
6. **Realice las pruebas** de los métodos indicados con JUnit.