



Más sobre la práctica 1 (revisada)

---



# La clase Cell

---

- Atributos: un entero para representar el valor contenido en la baldosa
- Métodos:
  - ...
  - **boolean doMerge(neighbour)** fusiona, si es posible, **this** y **neighbour**, y deja a 0 el valor de **neighbour**



# La clase Board

- **MoveResults executeMove(dir)** se puede definir con la ayuda de cuatro métodos privados **MoveResults executeMoveUp()**, **MoveResults executeMoveDown()**, ... que desplazan y fusionan columnas (up y down) o filas (left y right) según sea **dir**.
- Los métodos privados anteriores se pueden definir todos, a su vez, en términos de un método privado único **Cell[] desplazaYFusiona(Cell[])**
- Este método **desplazaYFusiona(Cell[])** desplaza en una dirección fija, por ejemplo, hacia la derecha, de manera que, cuando lo quiera usar **executeMoveUp()**, por ejemplo, le tiene que pasar una columna de células en un array de células, colocando la primera célula de la columna en la componente última del array, la segunda en la penúltima, etc.
- El método **desplazaYFusiona(Cell[])** primero desplaza células hacia la derecha, luego fusiona de derecha a izquierda (puede que produciendo ceros) y, por último, vuelve a desplazar hacia la derecha.



# La clase Board

---

Para definir el método **toString()** de esta clase **no** es necesario que utilices los métodos de **MyStringUtils**. Si los quieres usar –no es obligatorio- tienes que incorporar esta clase a tu proyecto, como una clase más.

- Para conocer lo que hacen los métodos **format()** y **substring()** de **String** puedes consultar la documentación de la clase **String** que ofrece **Oracle**.
- La clase **Board** necesita otros métodos: (1) **freeCells()**, que proporciona un array con las **posiciones** de las células libres que hay en el tablero. Sobre este array se elegirán aleatoriamente las iniciales y la nueva tras cada jugada que haya supuesto movimiento de células; (2) **fullBoard()**, para averiguar si el tablero está lleno; (3) **canMerge()**, para averiguar si hay dos células contiguas fusionables (en fila o columna) en el tablero. Estos dos últimos métodos se usan para saber si la partida ha terminado.



# La clase Game

---

- Para determinar las células iniciales, el juego le pide a su tablero las posiciones de las células libres, las baraja y después va eligiendo aleatoriamente posiciones.
- De forma análoga se determina la posición de la nueva célula después de que una jugada haya producido movimiento de células.
- Para barajar **se debe hacer** lo que hacen los tres métodos de la clase **ArrayAsList** del anexo. **No es necesario implementar la clase ArrayAsList**, pero **sí es necesario seguir lo que hacen los métodos shuffle() y choice()**.
- Otros métodos de la clase **Game** son **toString()** e **isFinished()**, este último para averiguar si el juego ha acabado.



# La clase Controller

---

- Para leer las jugadas del usuario se usa un objeto **sc** de la clase **Scanner**, que se construye así:  
**Scanner sc=new Scanner(System.in);**
- Este objeto lee las líneas que le suministra el usuario así:  
**String s=sc.nextLine();**
- El string resultante de la operación anterior es el que se trata y analiza con los métodos **trim()** y **split()** de la clase **String**.
- El método **run()** es un while que, mientras el juego no finalice, lee una orden del usuario, la analiza y le manda al juego que la ejecute.



# La clase Game2048

---

- Contiene el método **main()**
- Crea un controlador (que crea un juego -que, a su vez, crea un tablero-, y que crea un scanner) con el que invoca **run()**
- El tamaño de un tablero se lee con la instrucción  
**int dim = Integer.parseInt(args[0]);**
- Análogamente se lee el número de baldosas iniciales de **args[1]**
- Opcionalmente, la semilla se lee de **args[2]**  
**long seed = Long.parseLong(args[2]);**  
o se crea directamente con  
**long seed = new Random().nextInt(1000);**
- Recuerda que si **Random rand**, entonces  
**rand.nextInt(100);**  
genera un número aleatorio entero entre 0 y 99, ambos incluidos