

Programación Orientada a Objetos
Curso 2019/2020
Entrega 1 – Ejercicio 5

1. Funcionalidad.

Queremos desarrollar una aplicación para jugar a una versión simplificada del juego del **Bingo de 30 bolas**, más conocido como “**bingo rápido**”. Los jugadores del bingo disponen de cartones de 9 casillas, cada casilla contiene un número del 1 al 30. Gana el jugador que consigue “tachar” todos los números de su cartón.

La clase **Cartón** representa el cartón que utilizan los jugadores del bingo. Las propiedades que caracterizan a un cartón son (consultables):

- *números*: lista que almacena los números del cartón.
- *tamaño*: número que representa la cantidad de números del cartón (propiedad calculada).

Los números se establecen en la construcción y no podrán ser modificados. Por tanto, la clase solo declara un constructor.

La clase **Bingo** se caracteriza por las siguientes propiedades (todas consultables):

- *bombo*: lista de enteros con las bolas del bingo que todavía no han sido extraídas.
- *bolas extraídas*: lista de enteros que representa las bolas que han salido del bombo.
- *cartones*: lista de los cartones que están jugando.
- *cartones ganadores*: lista de cartones que han obtenido bingo.
- *finalizado*: valor booleano que indica si el bingo ha finalizado. Es una propiedad calculada. Un bingo ha finalizado si la lista *bombo* está vacía o la lista *cartones ganadores* tiene algún elemento.

Nótese que, dadas las restricciones del juego que se han indicado en la introducción, la clase debe declarar y hacer uso de dos constantes:

- *bola mayor*: número máximo de bolas del juego (30).
- *tamaño del cartón*: número de casillas del cartón (9).

La clase ofrece un constructor sin argumentos. En la construcción se inicializa la lista *bombo* con todas las bolas (desde 1 hasta *bola mayor*). El resto de listas quedarán vacías.

La funcionalidad de un bingo es la siguiente:

- **Comprobar cartón**: esta operación comprueba si el cartón establecido como parámetro es un “bingo”. Para ello consulta si todos los números del cartón están contenidos en la lista *bolas extraídas*. La operación retorna el valor booleano verdadero en caso de que sea bingo, falso en caso contrario.

- Generar un cartón. Esta operación tiene como objetivo generar un nuevo cartón y retornarlo. Los pasos que realiza son:
 - Construye una lista con todas las posibles bolas del bingo (números desde 1 hasta *bola mayor*). A esta lista la denominamos *todos los números*.
 - Mezcla el contenido de esa lista utilizando el método `static java.util.Collections.shuffle`.
 - Construye una nueva lista vacía. Denominamos a esta lista *números cartón*.
 - Realiza un bucle desde 1 hasta *tamaño del cartón* donde en cada paso:
 - Saca el primer elemento de la lista *todos los números*.
 - Introduce el elemento en la lista *números cartón*.
 - Construye un cartón con la lista *números cartón*.
 - Añade el cartón a la lista *cartones* del bingo.
 - Por último, retorna el cartón.
- Extraer una bola. Esta operación se encarga de extraer una bola del bingo. La bola extraída será el valor devuelto por la operación. Los pasos que realiza son:
 - Si el bingo está finalizado, la operación retorna el valor 0 como marca de error.
 - Utiliza el método `java.util.Collections.shuffle` para mezclar la lista del *bombo*.
 - Extrae el primer número de la lista *bombo* (lo obtiene y lo borra). Este número corresponde con la bola extraída.
 - Añade el número a la lista *bolas extraídas*.
 - Recorre la lista de cartones y comprueba si alguno tiene bingo. En caso afirmativo, añade el cartón a la lista de *cartones ganadores*.
 - Por último, retorna la bola extraída.

2. Programa

Implementa un programa con la siguiente funcionalidad:

- Declara y construye un Bingo.
- Crea tres cartones y muestra los números en la consola.
- Mientras el bingo no esté finalizado:
 - Extrae una bola y muéstrala por la consola.
- Si hay cartones ganadores, muestra los números del cartón por la consola.