

Programación

PRACTICA 3

Boleto de euromillones

1. OBJETIVO

El objetivo de la práctica es la elaboración de un programa que permita confeccionar un boleto de la lotería “euromillones” de forma aleatoria, tal y como lo hacen los terminales que hay en las oficinas de lotería. Además, como ampliación, se propone que se pueda introducir la combinación ganadora y el programa compruebe si el boleto tiene o no premio. El objetivo es la utilización de:

- Clases.
- Relaciones TIENE – UN entre clases.

2. EL JUEGO

- 2.1.- FUNCIONAMIENTO DEL JUEGO

Para apostar a euromillones hay que elegir 5 números y 2 estrellas.

- Los números se eligen de entre 50 posibles (1 a 50).
- Las estrellas se eligen entre 12 posibles (1 a 12).

La combinación ganadora consta de 5 números y 2 estrellas.

Una vez publicada la combinación ganadora, la cantidad de números y estrellas acertados determinan el premio que se obtiene. A cada categoría le corresponde un importe:

Categoría del premio	Aciertos	
	Números	Estrellas
1ª	5	2
2ª	5	1
3ª	5	0
4ª	4	2
5ª	4	1
6ª	4	0
7ª	3	2
8ª	2	2
9ª	3	1
10ª	3	0
11ª	1	2
12ª	2	1
13ª	2	0

- 2.2.- ¿QUÉ ES UNA APUESTA?

Se denomina **apuesta** a la realización de un pronóstico de números y estrellas. Estará constituido por 5 valores (números) entre 1 y 50 **distintos entre sí**, y 2 valores (estrellas) entre 1 y 12 **distintos entre sí**.

- 2.3.- ¿QUÉ ES UN BOLETO?

Un boleto está constituido por un mínimo de 1 y un **máximo de 5 apuestas**.

En la siguiente imagen se muestra el aspecto del impreso que los usuarios rellenan para hacer sus pronósticos. En el se aprecia un boleto con capacidad para cinco apuestas:

3. EL PROGRAMA.

El objetivo del programa es generar boletos y mostrarlos por pantalla

El programa principal consistirá en un bucle en el que:

- Se pide al usuario el número de apuestas que quiere jugar en el boleto.
- Se genera un boleto con el número de apuestas indicado por el usuario. Si el número de apuestas supera al máximo, al crear el boleto se producirá una excepción, que se capturará convenientemente para volver a solicitar al usuario el número de apuestas que desea (Ver constructor de la clase Boleto más adelante).
- Se muestra por pantalla el boleto generado con el formato que se indica a continuación. Ten en cuenta, que los números y estrellas de una apuesta siempre se muestran con los valores ordenados de menor a mayor:

```

EUROMILLONES
3 apuesta(s)
=====
1: 08 12 21 33 34 - (09 11)
2: 06 07 25 43 44 - (01 03)
3: 02 13 15 23 24 - (02 06)
=====

```

- Si se realiza la ampliación, el programa dirá a continuación si el boleto generado tiene o no premio y qué premios tiene (puede tener más de uno si el boleto tiene varias apuestas)
- El programa finalizará cuando el número de apuestas indicado por el usuario sea cero.

4. CLASES A IMPLEMENTAR.

Para implementar el programa se desarrollarán las siguientes clases:

- LA CLASE APUESTA.

En la actualidad, una apuesta está formada por 5 números enteros cuyo valor está comprendido entre 1 y 50 y 2 estrellas cuyo valor está comprendido entre 1 y 12. Sin embargo, consideraremos la posibilidad de que estos valores (5, 50, 2, 12) varíen en el futuro por lo cual, al crear una Apuesta, estos valores se pasarán como parámetro al constructor

Atributos (privados):

- *numeros*: ArrayList de enteros.
- *estrellas*: ArrayList de enteros.

Métodos (al menos los siguientes):

- *public Apuesta (int numNumeros, int numEstrellas, int maxValorNumeros, int maxValorEstrellas)*: Crea una apuesta con números obtenidos aleatoriamente.
 - *numNumeros* y *numEstrellas* indican, respectivamente, de cuántos números y cuántas estrellas consta una apuesta. *maxValorNumeros* y *maxValorEstrellas* indican, respectivamente, cuál es el valor máximo de un número y cuál es el valor máximo de una estrella. (Actualmente en el juego son 50 y 12)
 - No puede haber dos números repetidos en la misma apuesta. Si se genera un número y ya está en la apuesta, este se generará de nuevo, hasta que se obtengan *numNumeros* números distintos.
 - No puede haber dos estrellas repetidas en la misma apuesta. Si se genera una estrella y ya está en la apuesta, ésta se generará de nuevo, hasta que se obtengan *numEstrellas* estrellas distintas.
 - Una vez generados los números y las estrellas correctamente, se han de ordenar de menor a mayor. Para ello puedes usar `Collections.sort`
- *public boolean equals (Object o)*: Devuelve true si dos apuestas son iguales. Dos apuestas son iguales si contienen los mismos números y las mismas estrellas.
- *public String toString()*: Devuelve un string con los números que forman la apuesta, todos en la misma línea, separados por espacios en blanco y las estrellas entre paréntesis. Los números de una sola cifra aparecerán rellenados con ceros a la izquierda para lo cual puedes usar el método `String.format` (similar a `System.out.format` en su funcionamiento):

```
08 12 21 33 34 - (09 11)
```
- **La clase no tiene métodos getter ni setter**

- LA CLASE BOLETO.

Un boleto está formado por un máximo de 5 apuestas pero contemplaremos la posibilidad de que el número máximo de apuestas varíe, para lo cual definiremos una constante en la clase Boleto para este valor:

Constantes:

- *MAXAPUESTAS* = 5

Atributos (privados):

- *apuestas*: ArrayList de objetos Apuesta

Métodos (al menos los siguientes):

- *public Boleto (int numApuestas, int numNumeros, int numEstrellas, int maxValorNumeros, int maxValorEstrellas) throws IllegalArgumentException:*
Crea un boleto formado por el número de apuestas que se indica como parámetro. Las apuestas se almacenarán en el ArrayList *apuestas*.
 - numApuestas indica cuántas apuestas tendrá el boleto.
 - numNumeros y numEstrellas indican, respectivamente, cuántos números y cuántas estrellas tendrá cada apuesta.
 - maxValorNumeros y maxValorEstrellas indican, respectivamente, cuál es el valor máximo de un número y cuál es el valor máximo de una estrella.
 - No puede haber dos apuestas iguales en el boleto. Si se crea una apuesta y ya hay otra igual en el boleto, se generará la apuesta de nuevo.
 - Se lanza la excepción si el número de apuestas con que se pide crear el boleto es incorrecto, es decir, si es menor que uno o si es mayor que la constante MAXAPUESTAS
- *public String toString():* Devuelve un String con la información del boleto tal como se ha indicado anteriormente, es decir:

```

                EUROMILLONES
3 apuesta(s)
=====
1: 08 12 21 33 34 - (09 11)
2: 06 07 25 43 44 - (01 03)
3: 02 13 15 23 24 - (02 06)
=====

```

- **La clase no tiene métodos getter ni setter**

- LA CLASE EUROMILLONES.

Esta será la clase de programa que contenga al método main. Su funcionamiento debe ser el que se ha explicado en el apartado 2.

Se definirán cuatro constantes que representen la cantidad de números y estrellas de que consta una apuesta y los valores máximos admitidos para un número y para una estrella. Estas cuatro constantes se usarán al llamar al constructor de Boleto.

5. AMPLIACIÓN

Aumentar la funcionalidad del programa para que permita comprobar qué premios tiene el boleto generado.

Para ello, el programa principal, tras generar cada boleto comprobará los premios que tiene comparándolo con la combinación ganadora, cuyos números se encuentran en un fichero.

El programa mostrará el número de apuestas con premio en cada categoría acertada. Por ejemplo:

```

PREMIOS:
=====
12ª categoría: 1
13ª categoría: 2
=====

```

o el texto "SIN PREMIOS" si el boleto no tiene aciertos.

Para llevar a cabo la ampliación:

- crea un fichero “ganadora.txt” al mismo nivel que la carpeta *src* del proyecto (fuera de ella). Escribe en él 7 números enteros separados por espacios o por saltos de línea. Esos 7 números representan la combinación ganadora. Los cinco primeros valores corresponden a los “números” y los dos siguientes a las “estrellas”
- añade a la clase Euromillones un método *public static void leerCombGanadora(String nombreFichero, ArrayList<Integer> números, ArrayList<Integer> estrellas)* que lea del fichero indicado los 7 valores que componen la combinación ganadora (5 números y 2 estrellas) y los almacene en los ArrayList que se pasan como parámetro.
- añade a la clase Apuesta un método *public int calcularPremio (ArrayList<Integer> números, ArrayList<Integer> estrellas)* que, *dados* los números y estrellas premiados, devuelva:
 - 0 si la apuesta no tiene premio.
 - 1 si tiene premio de la 1º categoría,
 - ...
 - 13 si tiene premio de la 13º categoría,

Para ello habrá que comparar los ArrayList recibidos con los ArrayList de números y estrellas que tiene la apuesta, determinar cuántos números y estrellas coinciden y comprobar, siguiendo la tabla de premios que aparece al principio al comienzo de este enunciado, qué categoría de premio tiene la apuesta actual.

- añade a la clase Boleto un método *int[] calcularPremios(ArrayList<Integer> números, ArrayList<Integer> estrellas)* que, *dados* los números y estrellas premiados, devuelva un array que contenga:
 - En la posición 1, el nº de apuestas con premio de 1ª categoría
 - En la posición 2, el nº de apuestas con premio de 2ª categoría
 - ...
 - En la posición 13, el nº de apuestas con premio de 13ª categoría
- En Euromillones.main, añade lo necesario para comprobar los premios del boleto generado y mostrar los premios por pantalla.

6. TRUCOS PARA COMPROBAR LA CORRECCIÓN DE LA PRÁCTICA.

- Si pones la constante MAXVALORNUMEROS = 4;
 - Si tratas de generar una apuesta el programa debería quedarse “colgado”, puesto que no será capaz de generar cinco números distintos.
- Si pones la constante MAXVALORNUMEROS = 5;
 - Si generas una apuesta, los valores generados deberían ser 1, 2, 3, 4 y 5 puesto que no puede haber valores repetidos en una misma apuesta.
 - Si tratas de generar más de una, el programa debería quedarse “colgado”, puesto que no será capaz de generar dos apuestas distintas en el mismo boleto.
- **Para la ampliación:** Comprobar si una apuesta tiene premio puede ser complicado porque la apuesta se genera aleatoriamente y esto dificulta la realización de pruebas.

Estaría bien poder controlar que números se generan aleatoriamente cuando se crea una apuesta. Esto se puede hacer utilizando la clase Random con una “semilla”.

- La clase Random permite especificar una “semilla”, de manera que los números que se generen sean siempre los mismos. Esto permite saber qué números son los que se van a generar y ponerlos en el fichero “ganadora.txt” para comprobar los premios. Si estás interesado en esto pregúntame y te doy más detalles.
- Para poder utilizar una semilla necesitas que el objeto Random se cree una sola vez en la clase Apuesta y se utilice siempre el mismo objeto a partir de su creación. ¿Como se consigue esto? En la clase Apuesta, añades un atributo:

```
private static Random r = new Random(semilla);
```

- La variable static r se crea una sola vez, y es compartida por todos los objetos de la clase Apuesta, de manera que todas las Apuesta utilizarán el mismo objeto para generar sus números.

7. ENTREGA DE LA PRÁCTICA:

- Se entregará el proyecto completo una vez comprimido en un archivo. El proyecto tendrá el nombre *practica3Euromillones.ApellidoNombre*, con la parte “ApellidoNombre” personalizada con tus datos
- Fecha límite de entrega: **Jueves, 17 de febrero**