

PRO-UT1-UT2. Prueba extraordinaria de recuperación. Curso 2021-22

La prueba

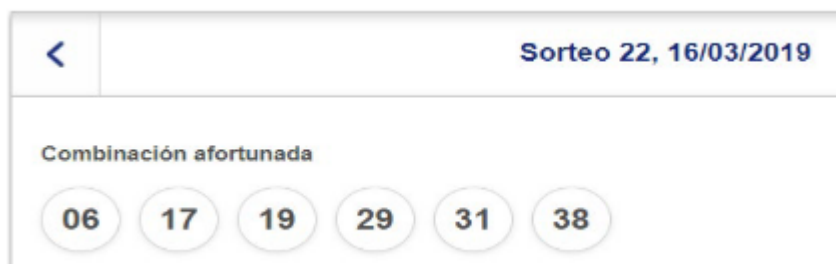
- Se realizará en 3 sesiones de clase
- Puedes utilizar todos los documentos del apartado **Contenidos** del aula virtual para la UT1 y UT2
- Una vez dispones del enunciado de la prueba y de los contenidos debes desconectar el cable de red del equipo.
- Cuando termines conecta el cable de red para poder entregar los archivos con el código fuente de la actividad.
- Si no tienes claro lo que se pide o necesitas alguna aclaración pregunta al profesor.

Práctica a realizar

1. Requisitos

Se desea implementar el sistema del sorteo de la Primitiva y del Euromillon.

El boleto de la primitiva está formado por 6 valores del 1-49.



El boleto de euromillon esta formado por 5 valores del 1-50 y dos estrellas formadas por números del 1-12



El sistema permitirá generar boletos de Primitiva y de Euromillón y almacenarlos.

Una vez generados los boletos de los participantes se generará una **combinación ganadora** de Primitiva y de Euromillon.

Después de generar la combinación ganadora se mostrarán los **resultados** del sorteo y **estadísticas** del mismo. Esta última parte solo se realizará para la lotería primitiva.

2. Implementación

Clase **Loteria**

Se creará una clase **abstracta Loteria** que contenga los atributos y métodos **comunes** de las clases **Primitiva** y **Euromillon**. Aparte del constructor, al menos deberá incluir los métodos:

- `_generar_boleto()`: genera los valores del boleto
- `get_boleto()`: devuelve lista con los valores generados.
- `__str__()`: devuelve cadena de texto con la información del boleto

Clase **Primitiva**

Se creará la clase **Primitiva** que heredará de **Loteria** e implementará los métodos abstractos de la misma teniendo en cuenta que:

- Al crear un objeto de la clase **Primitiva** se generarán los números del mismo.
- La combinación estará formada por 6 números del 1 al 49 **no repetidos** y **almacenados en orden**.
- Implementará, al menor, todos los métodos de la clase **Loteria**

Clase **Euromillon**

Se creará la clase **Euromillon** que heredará de **Loteria** e implementará los métodos abstractos de la misma teniendo en cuenta que:

- Al crear un objeto de la clase **Euromillon** se generarán los números del mismo. La combinación estará formada por 5 números del 1 al 50 **no repetidos** y dos estrellas del 1-12 que **no se repiten** entre ellas. Se almacenarán en **dos listas** con los números en **orden**
- Implementará, al menor, todos los métodos de la clase **Loteria**

Clase **Registro**

La clase **Registro** almacenará en una lista todos los boletos (objetos) de la primitiva y en otra lista todos los boletos (objetos) de euromillon que se generen.

- Incluirá, al menos, los métodos necesarios para añadir boletos a las listas correspondientes.

Clase **Sorteo**

- En el momento de crear un objeto de la clase **Sorteo** se le pasará un objeto de la clase **Registro** con los boletos de primitiva y euromillon generados y generará una combinación ganadora de la primitiva.
- Incluirá método `__str__()` que mostrará las combinaciones ganadoras de primitiva y euromillon.
- Incluirá un método `mostrar_resultados_primitiva()` que mostrará por pantalla:
 - La combinación ganadora
 - Cada uno de los boletos de la primitiva que participan en el sorteo y el número de aciertos del mismo

De forma similar a:



```

Combinacion ganadora: 3, 23, 30, 38, 44, 49
Boletos          Aciertos
11, 13, 23, 29, 35, 42    1
3, 28, 29, 45, 46, 49    2
1, 14, 20, 22, 24, 27    0
2, 20, 26, 27, 31, 49    1
...
3, 6, 11, 17, 22, 29     1

```

- Incluirá un método `mostrar_estadisticas_primitiva()` que mostrará por pantalla:
 - El número total de boletos registrados
 - El número de boletos con `n` aciertos.

De forma similar a:

```

Participan 100 boletos en el sorteo.
Los resultados son:
0 aciertos: 55
1 aciertos: 33
2 aciertos: 10
3 aciertos: 2
4 aciertos: 0
5 aciertos: 0
6 aciertos: 0

```

Programa principal

- Se generarán 100 boletos de primitiva y 100 de euromillon
- Se guardarán en un objeto de la clase `Registro`
- Se generará un objeto de la clase `Sorteo` al que se le pasará el registro de boletos y generará las combinaciones ganadoras de primitiva y euromillón.
- Se mostrarán los números del sorteo de la Primitiva y del Euromillón.
- Se mostrarán los resultados del sorteo de la primitiva (`mostrar_resultados_primitiva()`)
- Se mostrarán las estadísticas del sorteo de la primitiva (`mostrar_estadisticas_primitiva()`)

3. Calificación de los apartados

1. Crear las clases `Loteria`, `Primitiva`, `Registro`, `Sorteo` y cualquier otra adicional que necesites.(0,5p)
2. Definir las relaciones de herencia entre las clases de forma óptima(0,5p)
3. Realizar método que genera los valores de un boleto de la primitiva heredado de la clase `Loteria`(1p)
4. Realizar método que genera los valores de un boleto del euromillon heredado de la clase `Loteria`(1,25p)
5. Realizar un método que muestra los números de un boleto de la primitiva heredado de la clase `Loteria`.(0,25p)
6. Realizar método que devuelve los números de un boleto de la primitiva heredado de la clase `Loteria`.(0,25p)

7. Realizar un método que muestra los números y estrellas de un boleto del euromillón heredado de la clase Loteria.(0,25p)
8. Realizar un método que devuelve los números y estrellas de un boleto del euromillón heredado de la clase Loteria.(0,5p)
9. Realizar en la clase Registro un método que permite añadir un boleto de la primitiva.(0,25p)
10. Realizar en la clase Registro un método que permite añadir un boleto del euromillon.(0,25p)
11. Generar constructor de la clase Sorteo al que se le pasa el registro de boletos de primitiva y euromillon y genera una combinación ganadora para ambos sorteos que almacenará en el propio objeto. (1p)
12. Crear método que muestra los resultados del sorteo de la primitiva y del euromillon. (0,5p)
13. Crear método `mostrar_resultados_primitiva()` de acuerdo a las especificaciones.(1p)
14. Crear método `mostrar_estadisticas_primitiva()` de acuerdo a las especificaciones.(1p)
15. Crear programa principal de acuerdo con las especificaciones del apartado `programa principal` del apartado `implementación` (1,5p)

4. Criterios de calificación

- No se corregirá el supuesto si este tiene errores sintácticos.
- Al calificar cada apartado se tendrá en cuenta:
 - Debe realizar exactamente lo que se pide.
 - Se usa el nombre de las clases y métodos propuestos.
 - En las relaciones de herencia se define en la clase madre todo el código común y se aprovecha en todas las clases hijas (usando `super()` si es posible)
 - No se debe acceder al atributo de una clase desde fuera de la clase. Se generarán los métodos para hacerlo si son necesarios.
 - Los únicos métodos públicos en las clases serán aquellos que son accedidos de forma externa (desde otros objetos o desde el programa principal). Salvo en método `__str__()`
 - Se refactoriza/reutiliza código en la medida de lo posible.
 - Los nombres de identificadores de variables, constantes, atributos, métodos y clases deben expresar correctamente la información que almacenan o la funcionalidad que realizan.

Calma y mucha suerte