

# Práctica 1

## “Juego RULETA DE LA FORTUNA”

### 1.1. Objetivos

El objetivo es que el alumno construya un programa para jugar al juego de la ruleta de la fortuna.

- Mediante esta práctica se desea afianzar los conocimientos básicos de POO del alumno:
  - o Identificación de clases, atributos y métodos.
  - o Identificar el ámbito de clases para cada uno de ellos.
  - o Diferencia entre variables de clase y variables instancia.
  - o Relación de contención entre objetos.

Quedan fuera de esta práctica conceptos más avanzados de POO como son herencia y polimorfismo (estos conceptos serán abordados en la práctica 2 de la asignatura).

### 1.2. Planteamiento:

Se trata de realizar un programa para jugar al juego de LA RULETA DE LA FORTUNA. El juego consiste en un panel con una frase con los caracteres ocultos. El jugador irá tirando a la ruleta de tal forma que le saldrá una puntuación (10, 20, 30, 40 y 0), irá eligiendo consonantes hasta conseguir acertar la frase o agotar el número máximo de intentos. Se seguirán estas premisas:

- Si la consonante está en la frase, entonces se mostrarán esas consonantes en la frase oculta y se sumará a la puntuación del jugador la puntuación por el número de consonantes encontradas.
- Si esa consonante no está en la frase no pasa nada, básicamente ha perdido un intento.
- Si sale el 0 pierde la puntuación acumulada. Es como la bancarrota en la ruleta original.
- Antes de volver a tirar la ruleta, el jugador podrá comprar una vocal por el importe de 30 puntos por vocal elegida. Evidentemente, si no tiene saldo de puntos no podrá comprar.
- En el juego original hay varios jugadores, en nuestro caso lo haremos solo con un jugador.
- Existirán 3 niveles de juego:
  - o Novato: 10 intentos y frases fáciles.
  - o Medio: 8 intentos y frases de dificultad media.
  - o Experto: 5 intentos y frases de dificultad elevada.
- Existirán ya una serie de frases por defecto para cada nivel de juego, al menos 3 de cada tipo.
- El jugador podrá jugar 1 o n partidas que elegirá al inicio del juego.

Para ello se solicitarán por pantalla los siguientes datos:

- Datos jugador
  - o Alias.
  - o Password.
  - o Nombre.
  - o Fecha de nacimiento.
  - o Número de partidas a jugar.
  - o Modo de juego

### 1.3. Definiciones:

A continuación, se detallan los actores principales que aparecen en el programa:

**Jugador:** Se trata de la persona que quiere jugar, almacenaremos su alias, password, nombre, y su fecha de nacimiento, siguiendo estas premisas:

- Alias, password y fecha de nacimiento son obligatorios.
- El password debe tener al menos longitud 6.
- La fecha de nacimiento no puede ser mayor de la actual.

**Partida:** Estará definido por los datos de la partida del jugador, en concreto tendrá:

- El número de partidas a jugar
- Jugador que está jugando
- Puntuación acumulada de las partidas y el juego

Consideraciones adicionales:

- Si no se especifican número de partidas se realizarán por defecto 3.
- La puntuación por defecto es 0.

**JuegoRuletaFortuna:** Estará definido por los datos necesarios para realizar el juego de la ruleta de la fortuna, modo de juego, puntuación del juego en curso, frase a adivinar, letras acertadas y letras no acertadas, siguiendo estas premisas:

- Existirán 3 niveles de juego:
  - o Novato: 10 intentos y frases fáciles.
  - o Medio: 8 intentos y frases de dificultad media.
  - o Experto: 5 intentos y frases de dificultad elevada.
- Si no se especifica nivel de juego por defecto es Novato.
- Tendrá un método llamado comprobarEdad para validar si el juego es válido para el usuario que quiere jugar. Por defecto, no pueden jugar los menores de 10 años.
- Tendrá un método jugar que tendrá toda la lógica de cada jugada:
  - o Mostrará los datos para realizar el juego; frase encriptada, vocales elegidas, consonantes elegidas, puntuación...
  - o Solicitará al usuario si quiere comprar vocal (siempre que tenga puntos suficientes)

- Tirará la ruleta para obtener la puntuación (no requiere interacción del usuario)
- Solicita al usuario una consonante
- Valida si existe y calcula la puntuación a acumular, si procede.
- Tendrá un método llamado `esGanador` y `esPerdedor` para saber si gana o no y otro llamado `puedeSeguirJugando` para validar si ha agotado el número de intentos en función del nivel.
- Tendrá un método llamado `mostrarInfoResultado` con el resultado de cada jugada.

**Frase:** Esta clase contendrá todas las frases en función de la dificultades fácil, media y elevada, siguiendo estas premisas:

- Tendrá un método llamado `dameFrase` que devolverá la frase aleatoria entre las almacenadas, con la que jugar en función del nivel de juego.

**Carácter:** Esta clase se encargará de toda la gestión de los caracteres: validación de consonantes, de vocales, etc..., siguiendo estas premisas:

- Tendrá un método `esConsonante` y otro llamado `esVocal` para validar que los datos del usuario concuerdan con las normas del juego.

**Tirada:** Esta clase se encargará de obtener aleatoriamente la puntuación de la tirada, los valores posibles son (10, 20, 30, 40 y 0):

- Tendrá un método `tirar` que devolverá la puntuación por la que se juega.

Para poder jugar a la aplicación el usuario deberá estar dado de alta o entrar como invitado. Por tanto, el usuario deberá registrarse o bien entrar como invitado con alias "invitado" y password "invitado". En caso de no hacerlo así se le denegará el acceso. Por tanto, se almacenará un listado de usuarios almacenados en la aplicación y se validará que o bien se entra con el usuario invitado o el usuario se ha registrado previamente.

### 1.4. Consideraciones a tener en cuenta:

En el caso de variables que solo admiten x valores se pueden utilizar enumeraciones, como por ejemplo la siguiente:

```
public enum Nivel
{
    NOVATO, MEDIO, EXTERTO
}
```

Para almacenar varios valores de un mismo tipo se recomienda utilizar la clase `ArrayList`, que permite agregar y eliminar elementos de forma dinámica. También podemos utilizar cualquier otra clase de tipo colección que cumpla con los requerimientos solicitados. No se recomienda la utilización de arrays clásicos por ser más costosos de implementar y no aportar valor añadido para el conocimiento de la asignatura.

## 1.5. Parte persistencia (2 puntos)

Para obtener la puntuación máxima de la práctica se deberá implementar persistencia en fichero:

- Persistencia de jugadores, de tal forma que los jugadores no se perderán entre una ejecución y otra del programa.
- Persistencia de las partidas jugadas: Una vez terminada una tanda de partidas se solicitará al usuario si quiere guardar la información de las partidas jugadas.

Para la persistencia en fichero se utilizará una librería externa que serialice a XML o JSON, como por ejemplo GSON de Google, que persiste objetos de Java a formato JSON de forma transparente para el usuario.

*// Serialization*

```
Gson gson = new Gson();  
String objJSONSerialized = gson.toJson(partida);
```

*// Deserialization*

```
Gson gson = new Gson();  
return gson.fromJson(cadena.split("\\n")[0], Partida.class);
```

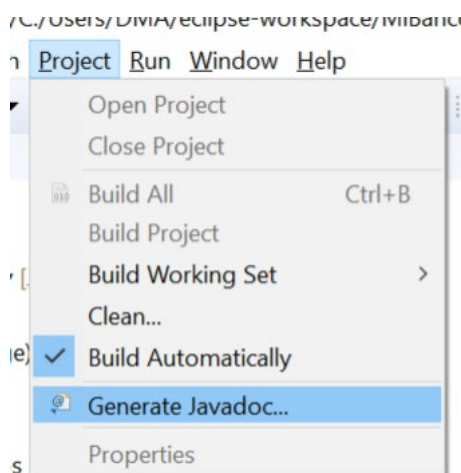
Para leer y escribir en un fichero solicitar al profesor las indicaciones correspondientes.

## 1.6. Evaluación

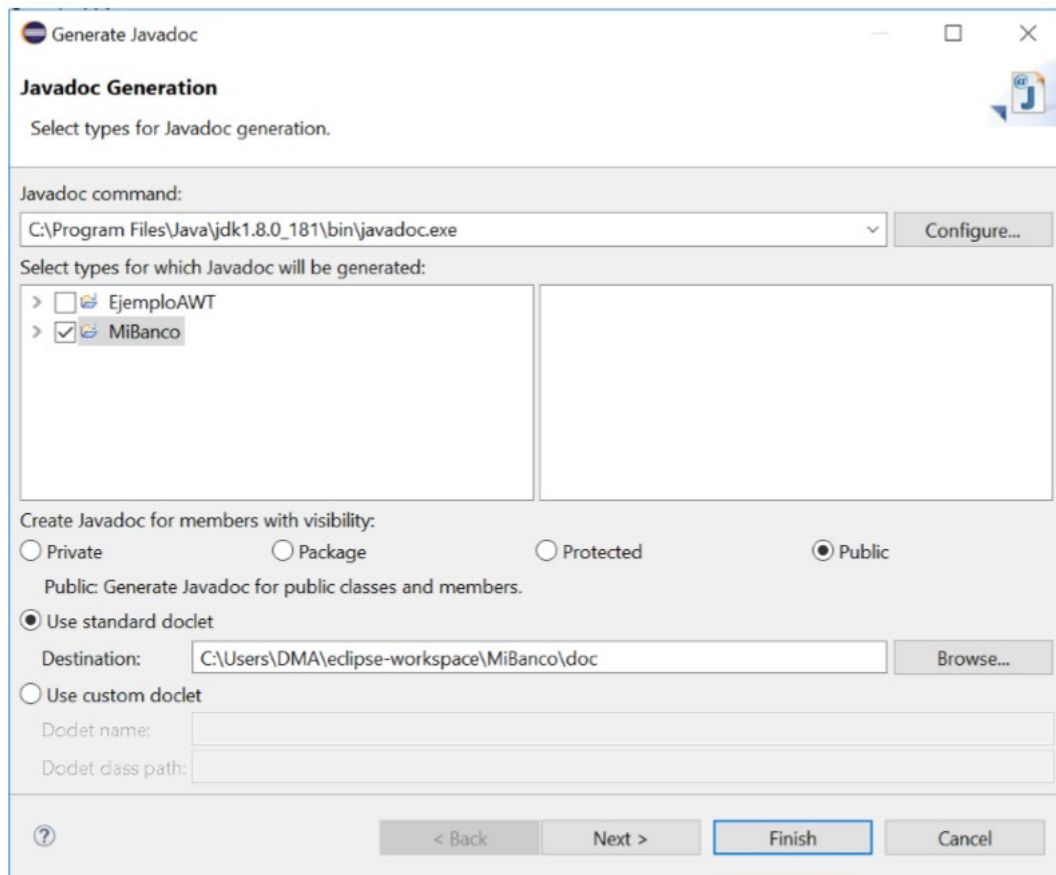
Más importante que terminar la práctica completamente, es que lo que se implemente esté realizado correctamente conforme al paradigma de POO y sea coherente.

Se valorará muy positivamente la limpieza del código, la documentación de este y la realización de comentarios siempre que sean útiles.

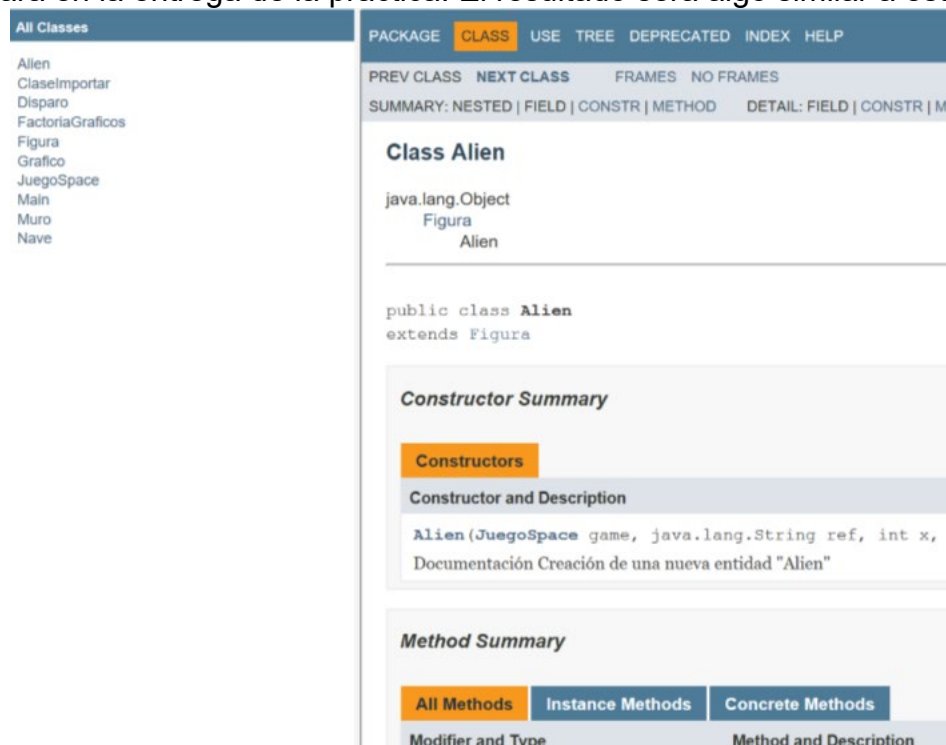
Es obligatoria la realización de la documentación Javadoc. Para ello documentaremos correctamente el código y al finalizar la práctica la generaremos con la opción Project / Generate Javadoc:



En la pantalla siguiente se informará la ruta del ejecutable javadoc que se distribuye con el JDK de Java.



El resto de los campos se dejará por defecto y la documentación resultado se adjuntará en la entrega de la práctica. El resultado será algo similar a esto:



### **1.7. Normas de entrega:**

La práctica deberá entregarse antes del **14 de noviembre de 2021** a las **23:59** mediante la opción de entrega Práctica 1 que aparecerá en el **Moodle** de la asignatura.