

Programación, juegos con dados: Risco

en Java y Python 3

Índice

Introducción del proyecto.	2
Planteamiento del proyecto.	3
Clases	3
Dados	3
Jugador.	4
Partida.	4
TestPartida.	4
Teclado	4
Menú.	4
Objetivo del proyecto.	5

1.Introducción del proyecto.

El proyecto se basa en el juego de dados Risco. Para comprender el proyecto comenzaremos explicando las reglas del juego.

El número de jugadores es ilimitado, se indicará al inicio de la partida.

Utilizamos tres dados durante 13 rondas.

El juego consiste en rellenar la tabla que veremos a continuación y ganará el que más puntos obtenga. La puntuación que se indica es la máxima que se puede obtener no siempre se obtendrá esa.

	Jugador 1	Jugador 2	Jugador 3	Etc.	Máximo puntaje de la categoría
Risco					50
Trece					26
Escalera mayor					20
Escalera menor					20
Escalera par					20
Escalera impar					20
Tres iguales					25
Seis					18
Cinco					15
Cuatro					12
Tres					9
Dos					6
As					3
Total					244

Cada participante tira dos veces por vuelta, puedes decidir cuántos dados quieres cambiar y cuáles por ejemplo cambiar solo el dado 1, el dado 1 y 3, todos o ninguno. Hay trece vueltas por jugador y en cada vuelta se deben anotar puntos en una categoría. Una vez que se haya anotado puntos en una de ellas, no podrá volver a hacerlo.

Después de las trece vueltas, cada uno tendrá puntaje en las trece categorías o juegos (aunque tenga cero en algunos de ellos). Se computan los totales y el que obtenga mayor cantidad de puntos es el ganador.

Puntuación:

Para los números se multiplica el total de dados con el número elegido por dicho número. Así, tres 6 valdrán 18 puntos en el seis; dos 5 valdrán 10 en el cinco.

- Risco: (cualquier par más un tercer dado que sumen 13 puntos, por ejemplo dos 5 y un 3) vale 50 puntos.
- Trece: (tres dados cualesquiera que sumen 13) vale 26 puntos.
- Escalera mayor: (4-5-6) vale 20 puntos.
- Escalera menor: (1-2-3) vale 20 puntos.
- Escalera par: (2-4-6) vale 20 puntos.
- Escalera impar: (1-3-5) vale 20 puntos.
- Tres iguales: vale 25 puntos.

2. Planteamiento del proyecto.

Comenzamos dividiendo el proyecto en las clases que necesitaremos. En este caso vemos 3 objetos principales:

- Datos.
- Jugador.
- Partida.

Para la comunicación con el usuario usaremos la clase TestPartida.

Por otro lado para la entrada y salida de datos crearemos la clase Teclado y para la gestión del menú la clase Menu.

3. Clases

a. Datos

Se encarga de generar el trío de dados del jugador de forma aleatoria. Además de cambiar los dados que indique el jugador si lo solicita. Sus atributos son:

- d1 (Int) → dado 1
- d2 (int) → dado 2
- d3.(int) → dado 3

b. Jugador.

Contiene la información (nombre, puntuaciones...) de cada jugador. Sus atributos son:

- Nombre (String) → Nombre del jugador
- nj (int) → Número de jugador, sirve como identificación
- p (ArrayList<Integer>) → Array de enteros, contiene todos los puntos que tiene el jugador.
- dadosJugador (Dados) → Son los dados del jugador.

Además de la información del jugador, en esta clase creamos el método encargado de exportar los datos del jugador a un archivo txt que se llamara *risco_<nombre del jugador>.txt*. Esta función nos permitirá guardar los datos de la partida(su puntuación, puesto en el que ha quedado, contra cuantos jugadores jugaba, etc)

c. Partida.

Es la clase que se encarga de la ejecución de la partida. Sus atributos son:

- ronda (int) → Ronda por la que va la partida
- nJugadores (int) → Número de jugadores que hay en la partida
- jugadores (ArrayList<Jugador>) → Array que contiene todos los jugadores que están jugando la partida.

d. TestPartida.

La clase TestPartida se encarga de la interacción con el usuario, consta del método Main, ya que es la que se ejecuta de cara a los jugadores reales.

e. Teclado

Se encarga de capturar los datos que se introducen por teclado.

f. Menú.

Clase encargada de gestionar el menú, mostrarlo y recoger la opción elegida. Sus atributos son:

- Título (String) → titulo del menu
- opciones(String[]) → Las opciones del menú
- numOpciones(int) → número de opciones

4. Objetivo del proyecto.

- Entender el concepto de la programación orientada a objetos y aplicarlo en Java y Python.
- Aprender a trabajar con ficheros.
- Aprender el uso de excepciones.
- Aprender a documentar y generar documentación en Python (PyDoc) y Java (JavaDoc).
- Uso de arrays, ArrayLists de Java y listas de Python.
- Mejorar la limpieza de código.
- Aprender a trabajar en equipo.
- Uso de Git y GitHub para el desarrollo del proyecto entre todos los programadores.
- Uso de expresiones regulares en Java y Python.