

## ESTRUCTURA DE DADES I ALGORISMES

Exercici d'avaluació inicial



Implementarem una aplicació per a jugar al Trivial. L'aplicació es desenvoluparà en parelles. Podeu treballar amb Eclipse o amb IntelliJ Idea.

### **Primer pas. Crear la classe Pregunta.**

La classe Pregunta tindrà els següents atributs. Un atribut de tipus String per a emmagatzemar una afirmació, un atribut de tipus boolean per a registrar si l'afirmació és una certa o falsa i un nombre enter de l'1 al 5, per a especificar la dificultat de la pregunta. Per exemple, {"La capital de França és París", True, 2}.

### **Segon pas. Crear una classe Partida que tingui el mètode main.**

La classe Partida inicialitza un ArrayList, o bé un array estàtic (depenent del que coneixeu de les assignatures prèvies) amb 10 preguntes. Us les podeu inventar vosaltres, o us podeu inspirar investigant en Internet.

Documentació:

[Java ArrayList](#)

<https://docs.oracle.com/javase/tutorial/java/nutsandbolts/arrays.html>

### **Tercer pas. Implementar el motor del joc.**

Quan el joc comença, li preguntem a l'usuari el seu nom. En una primera versió, el joc és només per a un jugador.

Documentació: [Scanner \(Java SE 16 & JDK 16\)](#)

Tutorial per llegir de teclat amb Scanner: [Java Scanner \(With Examples\)](#)

Després, remenem les preguntes. Podeu investigar la API de la classe Collections. Hi ha un mètode específic per assolir aquesta funcionalitat.

Documentació per remenar, en el cas que utilitzeu un array static:

<https://docs.oracle.com/javase/tutorial/displayCode.html?code=https://docs.oracle.com/javase/tutorial/collections/interfaces/examples/Shuffle.java>

En cada ronda, mostrarem l'afirmació de la següent pregunta a l'usuari. L'usuari respon veritable o fals. Si l'usuari ha encertat la resposta correcta, llavors incrementem en una variable la puntuació de l'usuari, sumant la puntuació assignada a aquesta pregunta.

El joc finalitza quan s'han consumit totes les preguntes del ArrayList o del array estàtic.

### **Quart pas. Mostrar estadístiques del joc.**

Al final del joc, el programa mostrarà unes estadístiques sobre el rendiment del jugador.

1. Total de respostes encertades
2. Total de puntuació obtinguda
3. Tant per cent de respostes encertades respecte del total

**Es recomana utilitzar un disseny modular. Cada pas representa, com a mínim, un mètode de la classe Partida.**

### **Funcionalitats Extres**

1. Ordenar les preguntes per dificultat (en comptes de random). Per a això, fareu que la classe Pregunta implementi la interfície Comparable.

Documentació:

[Comparable \(Java SE 16 & JDK 16\)](#)

[Collections \(Java SE 16 & JDK 16\)](#)

[Guide to Implementing the compareTo Method | Baeldung](#)

[Java67: 3 ways to sort a List in Java 8 and 11 - Example Tutorial](#)

2. Permetre que el trivial sigui multijugador. Crear la classe Jugador. Cada jugador tindrà un nom i una puntuació. Al final del joc es mostraran les estadístiques de cada jugador. Ordenar els jugadors per la seva puntuació.