# Java Quiz Reader

Progetto di Programmazione Avanzata

**Cristiano Gambirasio**

A.A.2022/2023

# 1 Introduzione

Il progetto sviluppato con Java consiste in un lettore di "quiz" scritti su file .txt che seguono uno standard inventato per questo progetto.

Il lettore permetterà di giocare ai quiz in modalità Singleplayer e Multiplayer, alla fine di ogni partita restituirà un piccolo report sul quiz appena svolto.

Durante la scrittura del codice sono stati usati i costrutti Java studiati durante il corso: **collezioni, varargs, record, ereditarietà, interfacce, overriding**

# 2 Funzionamento dell'applicazione

Appena avviata l'applicazione comparirà un menu che permette di scegliere tra partita singleplayer, partita multiplayer e la chiusura dell'applicazione.
Tutti gli input sono controllati dal software, quindi se l'utente inserisce un valore minore di zero, maggiore di 2 o non intero, verrà richiesto di inserire nuovamente una risposta. Allo stesso modo tutti i prossimi input presenti nell'applicazione.

## 2.1 Partita Singleplayer

Una volta selezionata una partita Singleplayer verrà richiesto di inserire il nome di uno dei quiz tra quelli che vengono elencati.
Una volta fatta la scelta inizierà il gioco: verranno mostrate tutte le domande una ad una, l'utente dovrà selezionare un numero tra 1 e 4 per rispondere.
Quando l'utente avrà risposto a tutte le domande, l'applicazione mostrerà i risultati del quiz.
Premendo "invio" ci si ritroverà al menù iniziale.

## 2.2 Partita Multiplayer

Anche selezionando una partita Multiplayer verrà richiesto di inserire che quiz fare. In più verrà richiesto inserire il numero di giocatori (tra 1 e 4).
In questa modalità ogni domanda verrà mostrata più volte in modo da permettere a tutti i giocatori di rispondere.
Alla fine del quiz vengono mostrati i risultati di tutti i giocatori.
Premendo "invio" ci si ritroverà al menù iniziale.

# 3 Formato dei quiz

I quiz letti dall'applicazione sono formati da directory contenenti:

- Un file *config.txt* contenente un intero pari al numero di domandi presenti nel quiz.
- Un certo numero di file .txt (pari al valore contenuto in *config.txt*), nominati con interi crescenti (*1.txt,2.txt,3.txt...*) contenenti 5 righe di testo ciascuno. Le righe contengono:
  1. La domanda
  2. La risposta giusta
  3. (fino alla quinta riga) Le tre risposte errate

I quiz sono contenuti in una cartella (Quiz) presente nella cartella principale del progetto

# 4 Struttura del codice

## 4.1 Package utils

Questo package contiene 3 classi:
ConfigReader e QuestionReader servono per interfacciarsi e leggere i file .txt, ConsoleCleaner è un oggetto che permette di cancellare la console.

## 4.2 Package partite

Contiene tutte le classi e interfacce che descrivono le modalità di partita:

- Partita) È una **classe astratta** che descrive una partita in generale. Oltre a contenere le informazioni riguardanti il path del quiz e il numero di domande, implementa l'interfaccia Staratable.
- Startable) È un'**interfaccia** che impone di avere il metodo *start* che restituisce un oggetto di tipo Report.
- Singleplayer-Multiplayer) Sono classi che **estendono** la classe astratta Partita. Le due classi fanno **override** del metodo start in modo che ognuna implementi lo svolgimento della partita nel modo giusto.

## 4.3 Package gameReport

Contiene i vari tipi di Report restituiti alla fine del quiz.

- Report) È una **classe astratta** che rappresenta un report in generale, impone la presenza del metodo *print*.

- ReportRecord) È un **Record** che prende come parametri il numero di risposte giuste, il nome del giocatore e una serie di risposte date passate come **varargs** di stringhe.
- ReportSP-ReportMP) Rappresentano i due tipi di report disponibili ed estendono la classe Report. Uno conterrà un solo ReportRecord, mentre l'altro un **ArrayList** di ReportRecord. Entrambi implementano il metodo print facendo **override** del metodo della superclasse.

## 4.4 Main

Nel main (*QuizGame.java*):

- Viene gestito l'inserimento in input dei parametri per giocare (modalità, quiz, numero di giocatori). L'inserimento viene fatto tramite il metodo statico safeInput che itera la richiesta di input fino a quando non viene inserito un valore valido.
  Questo metodo è definito in due modi (**overload**), in modo da gestire sia l'inserimento di interi che di stringhe.
  Per mostrare all'utente l'elenco di quiz disponibili si estrae uno **stream** di path che vengono poi inseriti in una **collezione** di tipo ArrayList.
- Viene inizializzata l'istanza "partita" corretta (Singleplayer o Multiplayer)
- Si esegue il metodo *start* della partita (**override**), il quale restituirà il report da stampare.