

Verifica sui vettori (ArrayList)

Simulazione di una gara di Formula 1

FASE A

Viene messa a disposizione una classe **Pilota** le cui caratteristiche sono:

- Nome del pilota
- Scuderia di appartenenza
- Punti acquisiti

Si ipotizzi che un pilota è univoco (ovvero non ci sono 2 piloti con lo stesso nome)

La classe ha i seguenti metodi:

- il costruttore completo
- i metodi getter
- il `toString()` (DA COMPLETARE) che restituisce una stringa del tipo:
 - "Hamilton (25) Mercedes "

Dove "Hamilton" è il nome del pilota, 25 sono i punti finora acquisiti in campionato e "Mercedes" è la scuderia di appartenenza

FASE B

Implementare una classe **Gara** descritta nel seguente modo:

- Nome della gara
- Un flag che stabilisce se la gara è in corso;
- Un insieme di piloti che dovranno gareggiare



Prevedere i seguenti metodi/azioni

- Un costruttore (inizializza il nome della gara e crea l'insieme vuoto dei piloti) (IMPLEMENTATO)
- Un metodo **aggiungiPilota** che inserisce un `Pilota p` (parametro entrante) in gara.
- Un metodo **grigliaDiPartenza** che accetta
 - un array di piloti e un array di scuderie tipo:

```
String[] piloti = {"Leclerc", "Hamilton", "Verstappen", ... }  
String[] scuderie = {"Ferrari", "Mercedes", "Red Bull", ... }
```

Crea e inserisce in gara i piloti che andranno a gareggiare. Si assuma che la pole position (il primo in griglia) è in posizione 0 (ovviamente l'ultimo sarà in posizione `size()-1`).

Le informazioni necessarie per creare un oggetto di tipo `Pilota` saranno accessibili, con riferimento agli array sopra-descritti, a parità di indice; per esempio: se si volesse creare il pilota Leclerc che guida una Ferrari scriveremo:

```
Pilota p = new Pilota(piloti[0], scuderie[0], ... );
```

Si assuma, inoltre, che il punteggio di ciascun pilota sarà, inizialmente, un qualsiasi valore compreso tra 0 e 50;

- Un metodo **classifica** che visualizza a video la posizione (in tempo reale) di tutti i piloti in gara (GIA' IMPLEMENTATO)
- Un metodo **start** (IMPLEMENTATO) che dà inizio alla gara
- Un metodo privato **posCorrente** che accetta il nome di un pilota (stringa); restituisce la posizione corrente nell'ArrayList di quel pilota. Se il pilota non esiste restituisce -1.
- Un metodo **fuoriGara** che accetta il nome di un pilota (stringa); rimuove dalla gara quel pilota
- Un metodo **effettuaSorpasso** che accetta il nome di un pilota (stringa); ha l'effetto di far avanzare di una posizione quel pilota (attenzione, il pilota in testa alla gara non può effettuare sorpassi!)
- Un metodo **pitStop** che accetta il nome di un pilota (stringa); ha l'effetto di far perdere 3 posizioni a quel pilota (attenzione, il pilota in ultima, penultima e terzultima posizione non può perdere 3 posizioni!)
- Un metodo **stop** che termina la gara e visualizza a video (PARZIALMENTE IMPLEMENTATO):
 - Il vincitore
 - L'ultimo arrivato
 - La classifica di fine gara

FASE C

Implementare una classe **Formula1** che contiene il metodo main (parzialmente implementato):

- Che crea la gara di Imola
- Inizializza la griglia di partenza della gara di Imola
- Simula una gara che prevede il completamento di 5 giri
 - nel 1° giro: Alonso e Stroll effettuano un sorpasso, Bottas va fuori gara
 - nel 2° giro: Giovinazzi fa un incidente e va fuori gara; Hamilton effettua un sorpasso
 - nel 3° giro: non succede niente
 - nel 4° giro: Hamilton rompe la macchina e va fuori gara; Verstappen effettua un sorpasso e Alonso effettua il pit stop
 - nel 5° giro: Schumacher effettua un pit stop
 - In ogni giro visualizzare la classifica in tempo reale
- La gara finisce
- A gara terminata, prevedere l'aggiornamento del punteggio del pilota vincente che andrà incrementato di 10 punti.

Se tutto funziona, a fine gara dovresti ottenere la seguente classifica:

Vincitore: Leclerc (23) Ferrari

Ultimo: Schumacher (22) Haas

Posizioni in gara:

| | |
|---|--------------------------|
| 1 | Leclerc (23) Ferrari |
| 2 | Stroll (31) Aston Martin |
| 3 | Verstappen (33) Red Bull |
| 4 | Alonso (50) Alpine |
| 5 | Schumacher (22) Haas |