

Non è consentito usare libri o appunti.

- **[12 punti]** Implementare opportunamente in Java la seguente gerarchia di classi:
 - una classe `Autovettura` che modella le autovetture che partecipano a campionati di automobilismo. Ogni autovettura è caratterizzata da `nome`, `numCavalli`, `tempoImpegnato`.
 - e tre sottoclassi
 - i. `Benzina` che fornisce il seguente metodo:
 - `getTempoGiro(int km, int numeroCurve, int numeroRettilinei)` se l'autovettura è accesa restituisce il tempo impiegato in minuti dall'autovettura per effettuare un giro in base al numero di curve e di rettilinei di un giro del tracciato utilizzando la seguente formula:
$$km * [(numeroCurve * 30 + numeroRettilinei * 30) / numCavalli]$$

Se l'autovettura è spenta viene lanciata l'eccezione controllata `AutoSpentaException`.

- ii. `Ibride` che fornisce il seguente metodo
 - `getTempoGiro(int km, int numeroCurve, int numeroRettilinei)` se l'autovettura è accesa restituisce il tempo impiegato in minuti dall'autovettura per effettuare un giro in base al numero di curve e di rettilinei di un giro del tracciato utilizzando la seguente formula:
$$km * [(numeroCurve * 20 + numeroRettilinei * 40) / numCavalli]$$

Se l'autovettura è spenta viene lanciata l'eccezione controllata `AutoSpentaException`.

- i. `Elettriche` che fornisce il seguente metodo
 - `getTempoGiro(int km, int numeroCurve, int numeroRettilinei)` se l'autovettura è accesa restituisce il tempo impiegato in minuti dall'autovettura per effettuare un giro in base al numero di curve e di rettilinei di un giro del tracciato utilizzando la seguente formula:
$$km * [(numeroCurve * 15 + numeroRettilinei * 45) / numCavalli]$$

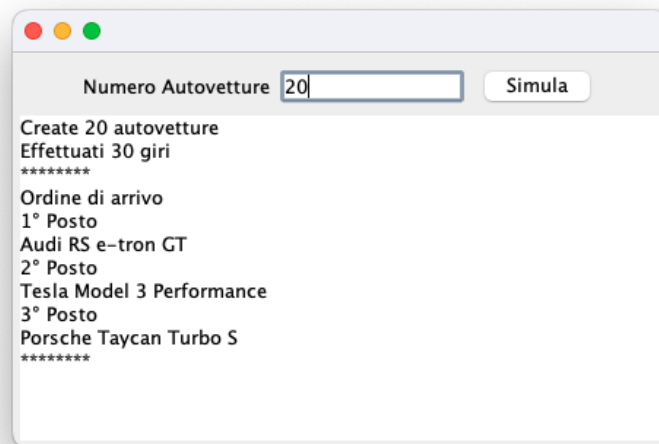
Se l'autovettura è spenta viene lanciata l'eccezione controllata `AutoSpentaException`.

Ad esempio l'autovettura elettrica "Tesla Model Y" con 347 Cv per percorrere un giro di 6 km con 12 curve e 5 rettilinei impiegherà $6 * (12 * 15 + 5 * 45) / 347 = 6,13$ minuti, mentre la vettura ibrida "Toyota RAV4" con 218 Cv impiegherà $6 * (12 * 20 + 5 * 40) / 218 = 12,13$ minuti.

I metodi `stop()` e `start()` sono uguali per tutte le autovetture. Il metodo `start()` cambia solo lo stato della macchina in "accesa", mentre `stop()` lo cambia in "spenta" ed azzerà il tempo impiegato.

- **[8 punti]** Scrivere la classe `GranPrix` che modella un insieme di autovetture che devono partecipare ad una gara. La classe è caratterizzata dal numero di giri, dal numero di km di un giro e dal numero di curve e di rettilinei. La classe è corredata dai seguenti metodi:
 - `void aggiungiVettura(Autovettura a)` che inserisce l'autovettura `a` nella lista in modo ordinato rispetto al numero di cavalli. Nel caso in cui nella lista è già presente una autovettura con lo stesso nome di `a` viene lanciata l'eccezione non controllata `InsertionException`.
 - `void avviaMotori()` che accende i motori di tutte le autovetture.
 - `void pitStop(int i)` che incrementa il tempo dell'autovettura in posizione `i` di 2 minuti.

- `void effettuaGiro()` che simula l'effettuazione di un giro di tutte le autovetture invocando il metodo `getTempoGiro`.
- **[10 punti]** Relativamente alle classi dei punti precedenti, scrivere una o più classi Java che realizzino un'interfaccia grafica che consenta la creazione di un numero (specificato in una casella di testo) di autovetture a caso (come mostrato in figura) tra le tre tipologie definite nell'esercizio 1. Verrà quindi creato un gran prix di 30 giri, dove ogni giro è lungo 7km, con 8 curve e 5 rettilinei, a cui saranno aggiunte le autovetture create. L'interfaccia grafica contiene anche un bottone per simulare la competizione. In particolare, ogni autovettura effettuerà tutti i giri della gara, e ad ogni giro verrà invocato il metodo `pitStop()` su un autovettura scelta a caso. Quando tutti i giri sono stati effettuati stampare a video le autovetture nei primi tre posti.



NOTE:

Ogni violazione delle regole enunciate ai punti sottoelencati comporta l'annullamento della prova (l'elaborato viene valutato 0).

1. Durante la prova d'esame è vietato usare:
 - a. libri e appunti sia in forma cartacea che in forma digitale
 - b. supporti di memoria esterni
 - c. un font di dimensione maggiore di 10 punti.
2. Il nome del progetto consegnato deve cominciare con COGNOME seguito dal carattere underscore e quindi dal NOME (tutto in maiuscole). Ad esempio, il nome del progetto di Marco Rossi può essere ROSSI_MARCO, ROSSI_MARCO_P2, ROSSI_MARCO_ESERCIZIO, ROSSI_MARCO_549449384, etc.
3. Il file da consegnare deve essere creato da eclipse seguendo i passi:
 - a. Seleziona "export..." nel menu file
 - b. Seleziona "Archive File" in "General"
 - c. Pressa "Next"
 - d. Seleziona progetto da esportare
 - e. Controllare il percorso del file (nell'area di testo con etichetta "To archive file:")
 - f. Assicurarsi che i pulsanti radio nel pannello Options siano selezionati su "Save in zip format" e "Create directory structure for files"
 - g. Pressa "Finish"

Assicurarsi che i progetti consegnati possono essere importati in eclipse come:

General → Existing Projects into Workspace