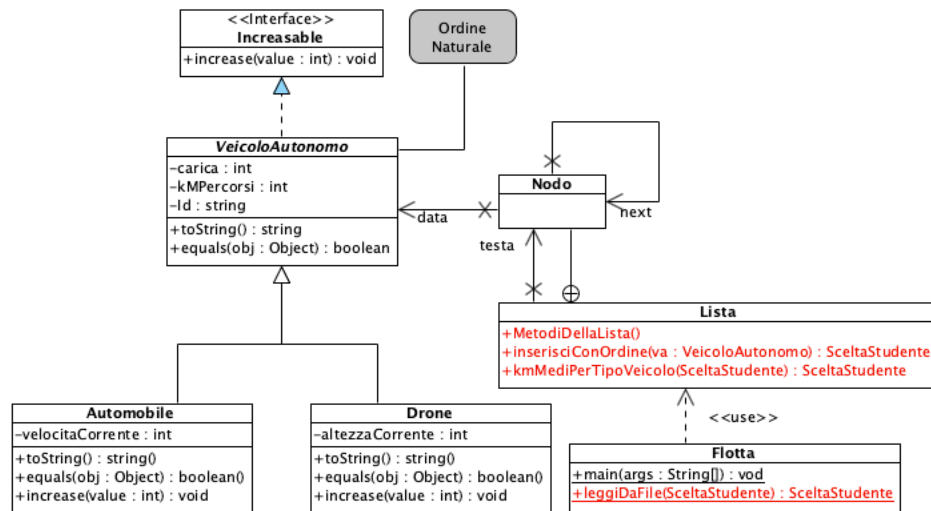


Corso di Programmazione A.A. 2023/2024

Esame del 28 Giugno 2024

Prof. Domenico Amalfitano

Si vuole implementare un'applicazione software per la gestione di una flotta di veicoli a guida autonoma. Il software è modellato dal diagramma UML riportato di seguito.



Requisiti del software.

1. La flotta utilizza una lista di veicoli autonomi che si specializzano in automobile e drone.
2. La lista inserisce un nuovo veicolo sempre in prima posizione.
3. I veicoli sono dotati di un ordinamento naturale che tiene conto della carica di ogni veicolo.
4. Ogni veicolo autonomo è `increasabile` in quanto implementa il metodo `increase` in base alla seguente logica:
 - a. Incrementa di `value` la velocità corrente delle automobili.
 - b. Incrementa di `value` l'altezza corrente dei droni.
5. Due veicoli autonomi sono uguali quando hanno lo stesso identificativo. Due automobili sono uguali se oltre ad avere lo stesso id hanno anche uguale velocità corrente. Analogamente due droni sono uguali se oltre ad avere lo stesso id hanno anche uguale altezza corrente.
6. Il metodo `kmMediPerTipoVeicolo` restituisce la media dei km percorsi dai veicoli della stessa tipologia, calcolata come il rapporto tra la somma dei km percorsi da una tipologia di veicoli e il numero di veicoli di quella tipologia presenti nella flotta. Lo studente gestisca opportunamente le situazioni in cui nella flotta non siano presenti veicoli di una o di entrambe le tipologie.
7. Il metodo `inserisciConOrdine` riceve in ingressi un veicolo autonomo. Il veicolo viene inserito in ultima posizione e vengono spostati, dalla lista, e inseriti dopo di esso tutti i veicoli che sono strettamente minori secondo l'ordine naturale. Nella pagina successiva è riportato un esempio di funzionamento del metodo.
8. Nel diagramma UML mancano i costruttori e i metodi di get e set degli attributi. Lo studente implementi i metodi che ritiene necessari.
9. È cura dello studente, inoltre, dotare la classe **Lista** dei metodi canonici a supporto dell'implementazione dei metodi richiesti. È ammesso l'uso dei container Java a supporto dell'implementazione dei metodi `kmMediPerTipoVeicolo` e `inserisciConOrdine`.
10. Si implementi la classe utente, **Flotta**, per testare le funzionalità del sistema tenendo conto anche di diversi scenari d'uso.
11. La classe **Flotta** carica i veicoli autonomi della lista da un file di testo opportunamente formattato dallo studente.

Esempio di Funzionamento del metodo inserisciConOrdine

Lista iniziale

```
carica=9, kmPercorsi=45, Identificativo=BMW, velocitaCorrente=11  
carica=2, kmPercorsi=13, Identificativo=Audi, velocitaCorrente=12  
carica=3, kmPercorsi=4, Identificativo=Audi, velocitaCorrente=12  
carica=11, kmPercorsi=7, Identificativo=DJI, altezzaCorrente=22  
carica=2, kmPercorsi=7, Identificativo=Drxon, altezzaCorrente=23
```

Inseriamo il veicolo:

```
carica=4, kmPercorsi=9, Identificativo=AutelRobotics, altezzaCorrente=32
```

Lista dopo chiamata del metodo

```
carica=9, kmPercorsi=45, Identificativo=BMW, velocitaCorrente=11  
carica=11, kmPercorsi=7, Identificativo=DJI, altezzaCorrente=22  
carica=4, kmPercorsi=9, Identificativo=AutelRobotics, altezzaCorrente=32  
carica=2, kmPercorsi=13, Identificativo=Audi, velocitaCorrente=12  
carica=3, kmPercorsi=4, Identificativo=Audi, velocitaCorrente=12  
carica=2, kmPercorsi=7, Identificativo=Drxon, altezzaCorrente=23
```