



Programmazione 3 e Laboratorio di Programmazione 3

Esercizi IS

Angelo Ciaramella

Calciatori e squadre di calcio

- Scrivere un **class diagram UML** che esprime l'associazione tra un **calciatore** e **squadre**. Arricchire, successivamente, il modello per esprimere singolarmente le seguenti proprietà
 - una **squadra** deve avere almeno **1 1 calciatori** e non esistono vincoli al numero di calciatori tesserati e un calciatore può essere associato al più a 1 squadra;
 - l'associazione descrive tutte le **squadre** per cui un calciatore ha giocato. Si vuole anche che venga specificata, per ogni squadra per cui il calciatore ha giocato, l'ultima stagione calcistica (anno) in cui ciò è avvenuto e il numero di goal segnati
 - si introduca anche l'entità **Allenatore** e la sua associazione con **Squadra**, specificando che una squadra deve avere esattamente un allenatore e un allenatore può allenare al più una squadra



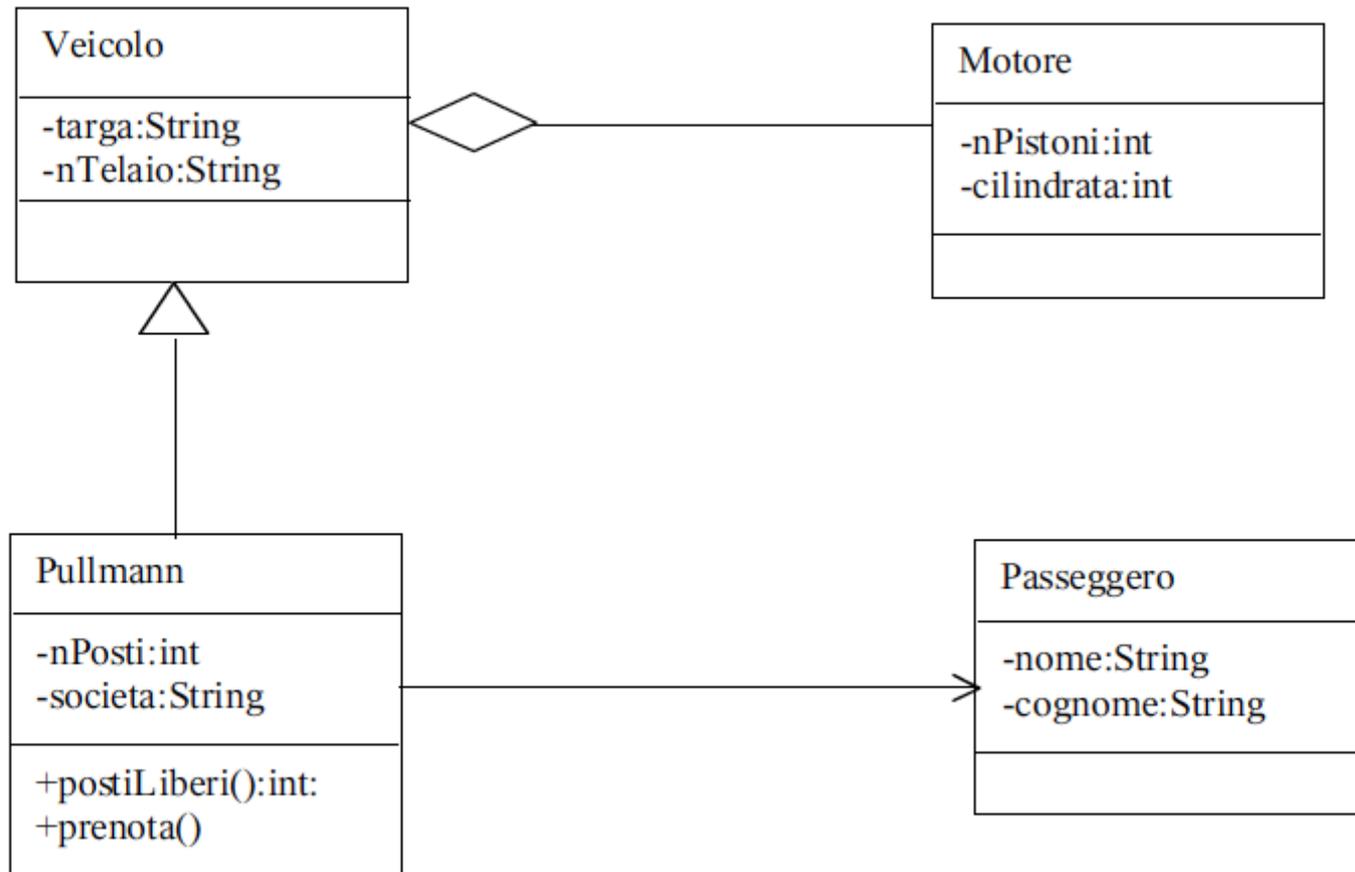
Mappa città

- La **mappa di una città** può essere vista a diversi livelli gerarchici
 - insieme di **zone** connesse da **vie principali**
 - ogni **zona** ha un **nome**, un **peso** e al proprio interno può contenere altre **zone** e/o **isolati**
 - ogni **via** ha un **nome** e una **lunghezza**, connette due o più **zone**, ma può anche collegare (separare) due o più **isolati**
 - ogni **isolato** è identificato da un **nome** e dal **numero di case** (**numeri civici**) in esso contenute
 - una **via** può solo collegare **isolati** della stessa **zona**, oppure zone diverse, ma non isolati di zone diverse
 - Ogni **zona** deve poter conoscere la **somma delle case in essa contenute**, direttamente o transitivamente
 - la **mappa** deve essere in grado di calcolare il **cammino minimo** (rispetto alla somma delle lunghezze delle vie attraversate) tra **due abitazioni di due zone qualsiasi**
- Descrivere con un **diagramma delle classi UML** la specifica sopra riportata



Pullman e passeggeri

- Scrivere il codice Java del seguente diagramma delle classi



Cantanti e case discografiche

- Si vogliono usare i diagrammi UML per esprimere l'associazione tra cantanti e case discografiche
 - una casa discografica può avere un numero arbitrario di cantanti e un cantante può incidere musica solo per una casa discografica
 - oltre ai cantanti singoli esistano i gruppi, che sono fatti da più cantanti
 - introdurre l'entità cd e si esprima in un diagramma UML le seguenti proprietà di cd, case discografiche e cantanti
 - un cd viene pubblicato da una e una sola casa discografica e un cantante pubblica un numero arbitrario di cd
 - si consideri un'implementazione in cui esistono due classi Java Cantante e CasaDiscografica



Macchina-Pilota

- Una **macchina radiocomandata** formata da due parti
 - la **macchina** stessa ed il **radiocomando**
- Ci sono 2 tipi di **comandi**
 - di **direzione**
 - Si può scegliere se girare a **destra** o a **sinistra**
 - di **velocità**
 - **accelerare** (in questo caso, di quanto accelerare) o se **frenare**
- Ogni radiocomando può essere configurato per funzionare su **frequenze diverse**
- La macchina si compone di
 - un **motore**, un **sistema di trasmissione**, di **4 ruote**, di un **ricevitore/controllore**, e di **2 servocomandi**
 - I servocomandi possono essere di 2 tipi, di direzione o di velocità.
 - una macchina ha un servocomando per la direzione (piazato sulle ruote anteriori) e uno per la velocità (installato sul motore)
- Il ricevitore/controllore della macchina può essere configurato per funzionare su **diverse frequenze** (la frequenza del ricevitore deve essere la stessa del radiocomando)
- Si rappresenti mediante un **diagramma delle classi UML** il sistema



Macchina-Pilota

- Scrivere un **Sequence diagram UML** che descrive la seguente situazione
 - il **guidatore accende** la macchina, quindi inizialmente **accelera** in direzione rettilinea, quindi **curva a destra**
 - durante la **curva frena**, quindi rimette le **ruote** in **posizione rettilinea**
 - Dopo avere rimesso le ruote in posizione rettilinea **frena**, quindi **spegne il motore**

