

Analisi e Progettazione del Software

Prova scritta del 28 febbraio 2018

Si vuole progettare una classe C++ per la gestione delle consegne giornaliere di un *corriere* espresso. Di un corriere interessano il nome, i suoi automezzi, le due coordinate geografiche del suo deposito e l'orario di apertura. Di ciascun *automezzo* interessa la targa e la capacità (valore reale in kg). Di ciascuna *consegna*, interessano le due coordinate geografiche del luogo di consegna e la quantità di merce che deve essere consegnata (sempre in kg).

Le consegne sono effettuate dal corriere tramite *viaggi*, dove un viaggio rappresenta un giro di consegne a vari clienti effettuato da un automezzo. Ciascun viaggio ha un orario di inizio e una sequenza *ordinata* di consegne. Un automezzo può anche effettuare più viaggi nella stessa giornata.

Si considerino le seguenti operazioni fondamentali per la classe che rappresenta il corriere:

InserisciConsegna(c : Consegna) :

La consegna *c* viene inserita nel corriere (ma non assegnata ad un viaggio). Se *c* è già presente, l'operazione non ha effetto.

InserisciAutomezzo(a : Automezzo) :

L'automezzo *a* viene inserito nel corriere.

Precondizioni: Non esiste già un automezzo con la stessa targa di *a*.

CreaViaggio(a : Automezzo, o : Orario, c : Consegna) :

Viene creato un nuovo viaggio con automezzo *a*, orario di partenza *o* e unica consegna *c*.

Precondizioni: L'automezzo *a* e la consegna *c* sono già presenti nel corriere. L'orario *o* è successivo all'orario di apertura del deposito. L'automezzo *a* non effettua altri viaggi con partenza in orario uguale a *o*. La capacità dell'automezzo *a* è maggiore della quantità di merce della consegna *c*.

InserisciInViaggio(a : Automezzo, o : Orario, c : Consegna, p : intero)

La consegna *c* viene inserita in posizione *p* nel viaggio effettuato da *a* con partenza in orario *o*.

Precondizioni: Esiste l'automezzo *a* ed un suo viaggio che parte in orario *o* e che ha almeno *p* consegne inserite. Esiste già la consegna *c*, ma non è inserita in alcun viaggio. La capacità dell'automezzo è maggiore o uguale alla somma delle quantità di merce delle sue consegne inclusa *c*.

OrarioConsegna(c : Consegna) : Orario

Restituisce l'orario in cui verrà effettuata la consegna, in base ai percorsi che compongono il viaggio. Si assuma che l'automezzo vada dal deposito al luogo della prima consegna, poi a quello della seconda e così via. Si assuma disponibile una funzione esterna che riceve come parametri due coppie di due coordinate geografiche e restituisce il tempo in minuti necessari per andare da un punto all'altro (si ignorino i tempi di scarico della merce).

Esercizio 1 (punti 6) Si disegni il diagramma UML delle classi per l'applicazione, che comprenda anche le responsabilità delle associazioni.

Esercizio 2 (punti 6) Si scrivano le definizioni della classe **Corriere** e delle altre classi che compongono il diagramma UML. Si consideri disponibile la classe **Orario** che memorizza un orario della giornata. Si assuma che la classe **Orario** metta a disposizione tutti i metodi e gli operatori che si ritengono opportuni.

Esercizio 3 (punti 12) Si scrivano le definizioni dei metodi della classe **Corriere** che corrispondono alle operazioni sopra elencate e i selettori che si ritengono opportuni. Si definiscano tutti i metodi (modificatori e selettori) delle altre classi del diagramma UML. Si gestiscano le precondizioni tramite il lancio dell'eccezione `invalid_argument`.

Esercizio 4 (punti 6) Si scriva una funzione esterna che riceva come parametri un corriere, un automezzo ed un orario e restituisca la frazione di consegne effettuate da quell'automezzo che verranno effettuate entro quell'orario, rispetto a tutte quelle che l'automezzo effettua.