

Programmazione Orientata agli Oggetti

1 luglio2021

Cosa fare

Si progetti una o più gerarchie di tipo in Java in modo da supportare le operazioni indicate nel tema specificato sotto. Definire **classi e interfacce**, le signature dei **metodi**, e gli **stati astratti e concreti**, specificando il **ruolo e protocollo** delle classi/interfacce e i **contratti** dei metodi più importanti; **implementare lo stato concreto e i metodi**. Valuterò l'elaborato in base alla qualità del progetto e della sua implementazione (responsabilità, tipologie, contratti, ADT, parametrizzazione, qualità del body, pattern di design, astrazioni, incapsulamento). Non leggerò spiegazioni complesse, schemi e diagrammi.

Non consegnare il compito se non si sono implementati almeno i metodi per le operazioni principali e un Main che funga da client per questa API con esempi di chiamate di ciascuna delle operazioni principali.

La soluzione va scritta su dei file di testo – in particolare non binari - (ad es. usando IntelliJ, Eclipse o altro a piacere) che devono essere caricati sul sistema elearning.uniud.it entro il termine di scadenza (non ci sono deroghe se non per casi particolari concordati a priori e giustificati adeguatamente).

Scrivere i propri nome e matricola su ciascuno dei file con la soluzione.

Trasporti passeggeri su treno

L'azienda "Freccie" dispone di vari materiali rotabili (motrici, carrozze passeggeri, carrozze letto, carrozze ristorante) che vengono utilizzati in 0 o più treni (un treno deve avere 1 motrice, $N \leq 8$ carrozze, di cui 1 ristorante) in numero finito (K1 motrici, K2 carrozze, K3 ristoranti, K4 carrozze letto). Ciascun treno ha una stazione di partenza (es NA) e una di arrivo (ad es UD), una serie di stazioni intermedie (es Roma, FI, BO, FE, PD, TV, PN), un orario di partenza, un orario stimato di arrivo in ogni stazione. Se l'orario di un treno va oltre le 24 nel treno ci deve essere almeno 1 carrozza letto. Le carrozze hanno una capacità massima di passeggeri. I vari treni possono percorrere solo dei percorsi predefiniti (sia come partenza, come arrivo, che stazioni intermedie e tempi di percorrenza). Ogni treno deve avere una capacità minima di passeggeri. In ogni istante di tempo, il materiale rotabile si trova in 3 possibili stati: o in un treno in corsa, o in una stazione (in attesa di far parte di un altro treno) o in officina in stazione per la manutenzione. Ovviamente, quando si crea un treno che deve partire dalla stazione S, in S ci deve essere tutto il materiale necessario che non deve essere in manutenzione.

Si necessita di un'API che consenta di implementare queste operazioni principali (segnature solo indicative):

- poter definire un treno, associarlo a un percorso soddisfacendo alla capacità minima:
 - Treno: **creaTreno**(...) data una motrice, e una lista di carrozze, e una capacità minima
 - void: **associaPercorso**(treno, percorso, data e ora di partenza)
- poter cambiare stato al materiale rotabile e poter mettere in manutenzione (o riattivare) del materiale guasto o che necessita di lavoro:
 - void: **partito**(treno, stazione)
 - void: **arrivato**(treno, stazione)
 - void: **inManutenzione**(materiale, stazione)
 - void: **riabilita**(materiale, stazione)
- poter estrarre un elenco dei treni in corsa e del loro materiale rotabile in un certo momento:
 - **treniInCorsa**(...)
- in base ai treni definiti, poter estrarre l'elenco del materiale rotabile previsto essere in una stazione a una certa data-ora:
 - **materialeInStazione**(stazione, data-ora)

Scrivere un main che contenga chiamate a ciascuno di questi metodi.