

Parte teorica (15 punti in totale, 30 minuti)

Domanda 1 (5 punti): A quali componenti di un programma ad oggetti puo' essere applicato il modificatore final? Cosa comporta cio' nei vari casi?

Domanda 2 (5 punti): E' possibile implementare un metodo all'interno di un'interfaccia? Se si', come? Se no, perche'? Ed e' possibile definire un campo? Che restrizioni ci sono rispetto a quello che si puo' implementare e definire all'interno di una classe?

Domanda 3 (5 punti): Quali sono le differenze tra una variabile di tipo int e una di tipi Integer? E' possibile assegnare una variabile di tipo int a una variabile di tipo Integer? Ed una variabile di tipo Integer a una variabile di tipo int? Perche'?

Parte pratica (15 punti in totale, 1h00')

Domanda 1 (7 punti):

Si implementi in Java un programma orientato ad oggetti di gestione dei treni. In particolare, si implementi una classe Treno, in cui ciascun treno e' identificato da un numero, e contiene una serie di stazioni in cui si ferma (ad esempio, parte da Venezia Mestre, ferma a Padova, Bologna, Firenze, Roma e arriva a Napoli). Ciascuna stazione e' rappresentata da una stringa (ad esempio, "Venezia Mestre"). Per ciascuna stazione, il treno fornisce un metodo che ritorna l'orario di arrivo (o partenza nel caso in cui sia la stazione di partenza) del treno come istanza della classe java.util.Date. Nel momento in cui viene costruito un treno, e' necessario inserire queste informazioni, e non deve essere possibile modificarle successivamente. Un treno generico deve definire poi un metodo che ritorna una collezione di classi (rappresentate come stringhe, ad esempio "Standard", "Premium", "Business" ed "Executive") che pero' non puo' essere implementato genericamente.

Domanda 2 (4 punti):

Implementare poi due tipi di treno specifici, ovvero le classi TrenoRegionale e FrecciaRossa. Le classi di viaggio sono uguali per tutti i treni della stessa tipologia, ovvero "Prima classe" e "Seconda classe" per i treni regionali, e "Standard", "Premium", "Business" ed "Executive" per le frecce rosse).

Domanda 3 (4 punti):

Si implementi infine la classe Biglietto. Tale classe deve contenere una classe di viaggio, e opzionalmente l'identificativo di un treno. Deve inoltre essere possibile validare il biglietto. Si implementi quindi un metodo che validi il biglietto. Tale metodo riceve un treno, e una classe, e valida il biglietto se questo e' valido, ovvero: (i) se il biglietto e' stato gia' validato lanciata un'eccezione controllata, (ii) se il treno e' regionale controlla solo la classe (in quanto tali biglietti non sono specifici per un treno), (iii) se il treno e' una freccia rossa, controlla sia classe che identificativo treno. Il metodo ritorna true se il biglietto viene validato correttamente, false altrimenti.

Tutte le implementazioni devono massimizzare l'incapsulamento (encapsulation) attraverso tecniche di information hiding.