

I Compitino - 12 novembre 2008

Cognome

Nome

Matricola

1 (10 punti)

Definite una classe `Treni`, i cui oggetti modellano i treni delle linee ferroviarie. La classe deve avere:

- un costruttore pubblico

`public Treni (int cod, String part, String arr, int posti)` che costruisce un oggetto con un codice identificativo intero `cod`, una città di provenienza `part`, una città di arrivo `arr` e un numero di posti a sedere `posti`.

- un costruttore pubblico

`public Treni (int cod, String arr, int posti)` che costruisce un oggetto con un codice identificativo intero, una città di arrivo e un numero di posti a sedere. La città di provenienza è di default *Milano*.

- i metodi pubblici dal significato ovvio

**`String getStPartenza(), String getStArrivo(),
int getPosti(), String toString()`**

- i metodi

`String getStPartenza(String st_arrivo)`

che riporta la città di partenza del treno se la città d'arrivo è la stessa del parametro e `null` altrimenti e

`String getStArrivo(String st_partenza)`

che riporta la città d'arrivo del treno se la città di partenza è la stessa del parametro e `null` altrimenti

2 (3 punti)

Definite nella classe **Treni** il metodo pubblico **boolean maggiore (Treni t)** che riporta **true** se e solo se il treno corrente ha un numero di posti maggiore del treno parametro esplicito e identica provenienza e arrivo.

```
public boolean maggiore (Treni t) {
```

3 (3 punti)

Immaginate di aver definito due variabili **Treni treno_1, treno_2;**

Inizializzatele con valori esemplificativi e scrivete un'istruzione che faccia uso dei metodi che avete definito nell'esercizio 1. per controllare se c'è coincidenza tra i due treni, dove per *coincidenza* intendiamo soltanto che un treno arrivi in una città da cui ne parte un altro, creando in tal modo un legame diretto tra la città di arrivo del primo treno e la città di partenza del secondo. L'inizializzazione delle due variabili è necessaria per non causare errori, ma l'istruzione deve essere parametrica e generale.

4 (2 punti)

Esprimere in linguaggio Java la seguente condizione, usando gli operatori di relazione e quelli logici:
il numero **n** deve essere **maggiore di 3 ma non di 8**

Esprimere in linguaggio Java la negazione della condizione precedente senza introdurre l'operatore di negazione (applicare la legge di De Morgan).

5 (2 punti)

Esprimete in Java l'affermazione "se ho i soldi e non piove oppure mi invitano allora vado alla partita", introducendo tre variabili booleane *a*, *b*, *c* che esprimano rispettivamente le condizioni "avere i soldi", "piovere" e "essere invitati", mentre l'evento "andare alla partita", sarà rappresentato da una istruzione di stampa di un messaggio opportuno.

Dire per quali valori di *a*, *b*, *c* si verifica l'evento (tabella di verità intera)

6 (3 punti)

Date la stringa: `String riga = new String("Ehi fu, siccome immobile");`

calcolare il valore delle espressioni:

`riga.length()`

`riga.substring(4, 13).length() + riga.indexOf('m');`

`riga.substring(0,4)+`

`riga.substring(4,13).toUpperCase().replace('S','R')`

riga vale:

7 (.5 punto)

Dire qual'è la caratteristica di Java che consente di definire due costruttori diversi, come ad esempio nella classe **Treni**:

Dire come si distinguono i due costruttori:

8 (.5 punto)

Dire qual'è la caratteristica di Java che consente di specializzare un metodo, come il metodo `toString()`, per una data classe:

9 (4 punti)

Date le variabili: `int a = 2, b = 4`; calcolare i valori delle due variabili, dopo aver eseguito separatamente i due blocchi di istruzioni:

<code>a = a++ + --b;</code>	<code>a:</code>
<code>b += a - --b;</code>	<code>b:</code>

E ancora dopo l'inizializzazione `a = 2, b = 4` :

<code>a = ++a + b--;</code>	<code>a:</code>
<code>b += a - b--;</code>	<code>b:</code>

10 (5 punti)

Usando la classe **Dado** introdotta a lezione si scriva il frammento di programma che utilizza due dadi, che vanno dichiarati e istanziati, e un ciclo, in cui si lanciano i dadi fino all'ottenimento di un 12. Una variabile intera conta il numero di iterazioni.