Esame di Programmazione 2

6 luglio 2023

- [redacted]
- Prima di caricare un file su upload, verificare che sia compilabile; cancellare o includere in commenti le parti che provocano errori in compilazione. **I file che danno errori in compilazione non verranno valutati**. Può essere utile usare opzione `-Xlint` di `javac`, che controlla l'uso corretto dei tipi generici.
- **Non** vanno definiti package.

Voli

Lo scopo è definire delle classi per gestire dei voli. Le classi da definire devono essere pubbliche, inoltre:

- i costruttori e i metodi descritti nella specifica devono essere pubblici;
- i campi di una classe devono essere privati;
- le segnature sono a volte scritte in modo incompleto (mancano alcuni tipi);
- è possibile aggiungere a una classe ulteriori metodi (pubblici o privati);
- le classi elencate non devono compiere operazioni di stampa (`System.out.println`, ecc.) o di lettura da standard input; le uniche classi che possono farlo sono le classi che implementano una applicazione (classi con metodo `main`);
- è ammesso che i messaggi stampati dal proprio programma abbiano un formato diverso da quelli riportati nel testo, purché le informazioni siano equivalenti e chiaramente identificabili.

Si racconmanda di dare ai nuovi metodi dei nomi significativi, che esplicitino il loro significato, eventualmente scrivere commenti chiarificatori.

Classe `Data`

Classe che descrive una data. La classe definisce un solo costruttore che costruisce una data di cui si specificano giorno, mese e anno (tre interi).

Classe `Passeggero`

Classe che descrive un passeggero. La classe definisce un solo costruttore che costruisce un passeggero di cui si specificano nome e cognome (due stringhe). Ogni passeggero ha

un punteggio, che inizialemente è zero a che, a ogni volo effettuato, viene incrementato del punteggio associato al volo stesso. La classe definisce il metodo pubblico

- `addPunti(n)`

  Incrementa il punteggio di questo passeggero di `n` punti (valore intero).

Classe `PasseggeroPremium`

Sottoclasse della classe passaggero che descrive un passeggero premium, ossia un passeggero che ha diritto a un bonus quando ha raggiunto i 100 punti. Il bonus consiste in punti da aggiungere al punteggio maturato; per default il bonus è di 10 punti. La classe definisce un solo costruttore che costruisce un passeggero premium di cui si specificano nome e cognome (due stringhe). La classe definisce il metodo

- `setBonus(n)`

  Metodo statico che stabilisce che il valore del bonus è `n` (un intero).

Il bonus va assegnato al passeggero una volta sola, ossia la prima volta che il punteggio del passeggero raggiunge un valore maggiore o uguale a 100.

Classe `Volo`

Classe che descrive un volo. Un volo è caratterizzato da un codice (una stringa), la data in cui è effettuato, il punteggio associato al volo (un intero) e l'elenco dei passeggeri, da rappresentare con un oggetto di tipo `List<Passeggero>`. Il codice definisce un particolare tragitto (es., Milano Linate 10.05, Roma Fiumicino 11.10); è quindi possibile definire voli con stesso codice ma con date diverse, e in tal caso anche i punteggi possono essere diversi. Su uno stesso volo non possono comparire due passeggeri con stesso nome e cognome. La classe definisce i seguenti costruttori:

- un costruttore che costruisce un volo di cui si specificano il codice (una stringa), la data (tipo Data) e il punteggio (un intero).
- un costruttore che costruisce un volo di cui si specificano il codice (una stringa), giorno, mese e anno (tre interi) e il punteggio (un intero).

La classe definisce i seguenti metodi:

- `add(p)`

  Aggiunge a questo volo il passeggero `p` (tipo Passeggero), se questo è possibile (ossia, se questo volo non contiene già un passeggero con stesso nome e cognome di `p`). Se il passeggero è stato aggiunto, il suo punteggio va incrementato nel modo specificato sopra.

Classe `GestoreVoli`

Classe che gestisce più voli; i voli vanno rappresentati mediante un oggetto di tipo `List<Volo>`. La classe definisce un solo costruttore che costruisce un gestore vuoto. La classe definisce i seguenti metodi:

- `add(v)`

  Aggiunge a questo gestore il volo `v` (tipo Volo) se questo è possibile, ossia se non è già stato inserito un volo con stesso codice e stessa data del volo `v` (si ricordi che è invece possibile avere voli con stesso codice e date distinte).
- `add(v,p)`

  Se il volo `v` (tipo Volo) è definito nel gestore e se il passeggero `p` (tipo Passeggero) non è presente nel volo `v`, aggiunge `p` al volo `v` (come specificato sopra).

  Si consiglia di usare i metodi `indexOf` e `get` definiti nella interfaccia List (vedere la documentazione).

Esercizio 1

Compilare il file `Esercizio1.java` (vedi Fig. 1). Il comando `java Esercizio1` deve stampare le linee nelle figure 2, 3 e 4; l'ordine in cui le informazioni sono stampate non è rilevante.

Esercizio 2

Aggiungere alla classe GestoreVoli i seguenti metodi:

- `listaPasseggeri()`

  Restituisce la lista di tutti i passeggeri che sono su almeno uno dei voli gestiti.
- `listaPasseggeriRegolari`

  Restituisce la lista di tutti i passeggeri regolari (ossia, non premium) che sono su almeno uno dei voli gestiti.

Definire la applicazione `Esercizio2` che costruisce un gestore di voli come in `Esercizio1.java`, eliminando le operazioni di stampa, e successivamente stampa:

- l'elenco di tutti i passeggeri in ordine alfabetico;
- l'elenco di tutti i passeggeri in ordine decrescente rispetto al punteggio e, a parità di punteggio, in ordine alfabetico;
- l'elenco di tutti i passeggeri regolari in ordine alfabetico.

La applicazione `Esercizio2` deve stampare:

```
=======   ELENCO PASSEGGERI (ORDINE ALFABETICO) ======
Carlo,Bianchi, PUNTI: 145 *PREMIUM*
Donatella,Bianchi, PUNTI: 212 *PREMIUM*
Alberto,Rossi, PUNTI: 147
Barbara,Verdi, PUNTI: 132
Elisa,Verdi, PUNTI: 142
Fabio,Verdi, PUNTI: 142
=======   ELENCO PASSEGGERI (ORDINE PER PUNTI DECRESCENTE)  ======
Donatella,Bianchi, PUNTI: 212 *PREMIUM*
Alberto,Rossi, PUNTI: 147
Carlo,Bianchi, PUNTI: 145 *PREMIUM*
Elisa,Verdi, PUNTI: 142
Fabio,Verdi, PUNTI: 142
Barbara,Verdi, PUNTI: 132
=======   ELENCO PASSEGGERI REGOLARI (ORDINE ALFABETICO) ======
Alberto,Rossi, PUNTI: 147
Barbara,Verdi, PUNTI: 132
Elisa,Verdi, PUNTI: 142
Fabio,Verdi, PUNTI: 142
```

Esercizio 3

Aggiungere alla classe GestoreVoli il seguente metodo:

- `listaVoli()`

  Restituisce la lista di tutti i voli gestiti.

Definire la applicazione `Esercizio3` che costruisce un gestore di voli come in `Esercizio1.java`, eliminando le operazioni di stampa, e successivamente stampa l'elenco di tutti i voli in ordine cronologico; i voli con la stessa data vanno stampati in ordine alfabetico. Per ogni volo è sufficiente stampare la data e il codice. La applicazione `Esercizio3` deve stampare:

```
=======   ELENCO VOLI (ORDINE CRONOLOGICO)   ======
06/07/2023 -- VOLO V1
06/07/2023 -- VOLO V2
06/07/2023 -- VOLO V3
13/10/2023 -- VOLO V2
13/10/2023 -- VOLO V3
05/12/2023 -- VOLO V4
10/01/2024 -- VOLO V1
```

```
public class  Esercizio1 {

    public static void main(String[] args)  {
       Passeggero pA = new Passeggero( "Alberto", "Rossi");
       Passeggero pAbis = new Passeggero( "Alberto", "Rossi");
       Passeggero pAter = new PasseggeroPremium( "Alberto", "Rossi");
       Passeggero pB = new Passeggero( "Barbara", "Verdi");
       Passeggero pC = new PasseggeroPremium( "Carlo", "Bianchi");
       Passeggero pD = new PasseggeroPremium( "Donatella", "Bianchi");
       Passeggero pE = new Passeggero( "Elisa", "Verdi");
       Passeggero pF = new Passeggero( "Fabio", "Verdi");
       GestoreVoli gv = new GestoreVoli();
       Volo v1_1 = new Volo( "V1", 6, 7, 2023,20);
       Volo v1_2 = new Volo( "V1", 10, 1, 2024,30);
       Volo v2_1 = new Volo( "V2", 6, 7, 2023,25);
       Volo v2_2 = new Volo( "V2", 13, 10, 2023,20);
       Volo v3_1 = new Volo( "V3", 6, 7, 2023,10);
       Volo v3_2 = new Volo( "V3", 13, 10, 2023,40);
       Volo v4 = new Volo( "V4", 5, 12, 2023,102);
       gv.add( v1_1 );
       gv.add( new Volo(  "V1", 6, 7, 2023, 40  ) ); // non aggiunto (duplicato)
       gv.add(v1_2);
       gv.add(v2_1);
       gv.add(v2_2);
       gv.add(v3_1);
       gv.add(v3_2);
       gv.add(v1_1, pA);
       gv.add(v1_1, pAbis);  // non aggiunto (pA e pAbis sono la stessa persona)
       gv.add(v1_1, pAter);  // non aggiunto (pA e pAter sono la stessa persona)
       gv.add(v1_1, pB);
       gv.add(v1_1, pC);
       gv.add(v1_1, pD);
       gv.add(v1_1,pD);
       gv.add(v1_1,pC);
       gv.add(v2_1, pA);
       gv.add(v2_1, pC);
       gv.add(v3_1, pB);
       gv.add(v3_1, pC);
       gv.add(v3_1, pD);
       gv.add(v3_2, pD);
       gv.add(v3_2,pE);
       gv.add(v3_2,pF);
       gv.add(v4,pC);  // non aggiunto (il volo v4 non e' nel gestore)
       gv.add(v4);
       gv.add(v4,pA);
       gv.add(v4,pB);
       gv.add(v4,pE);
       gv.add(v4,pF);
       System.out.println("====================  VOLI (1)  ==================== ");
       System.out.println( gv );
       gv.add(v1_2,pD);
       gv.add(v1_2,pC);
       System.out.println("====================  VOLI (2)  ==================== ");
       System.out.println( gv );
       PasseggeroPremium.setBonus(40);
       gv.add(v2_2,pC);
       gv.add(v4,pD);
       // **** FINE INSERIMENTO VOLI ****
       System.out.println("====================  VOLI (3)  ==================== ");
       System.out.println( gv );
    } // end main

} //end class
```

Figura 1: File `Esercizio1.java`.

```
==================  VOLI (1)  ==================
VOLO V1, data: 06/07/2023  Punti:  20
Alberto,Rossi, PUNTI: 147
Barbara,Verdi, PUNTI: 132
Carlo,Bianchi, PUNTI: 55 *PREMIUM*
Donatella,Bianchi, PUNTI: 70 *PREMIUM*

VOLO V1, data: 10/01/2024  Punti:  30

VOLO V2, data: 06/07/2023  Punti:  25
Alberto,Rossi, PUNTI: 147
Carlo,Bianchi, PUNTI: 55 *PREMIUM*

VOLO V2, data: 13/10/2023  Punti:  20

VOLO V3, data: 06/07/2023  Punti:  10
Barbara,Verdi, PUNTI: 132
Carlo,Bianchi, PUNTI: 55 *PREMIUM*
Donatella,Bianchi, PUNTI: 70 *PREMIUM*

VOLO V3, data: 13/10/2023  Punti:  40
Donatella,Bianchi, PUNTI: 70 *PREMIUM*
Elisa,Verdi, PUNTI: 142
Fabio,Verdi, PUNTI: 142

VOLO V4, data: 05/12/2023  Punti:  102
Alberto,Rossi, PUNTI: 147
Barbara,Verdi, PUNTI: 132
Elisa,Verdi, PUNTI: 142
Fabio,Verdi, PUNTI: 142
```

Figura 2: Linee stampate dalla applicazione `Esercizio1` (parte 1).

```
==================  VOLI (2)  ==================
VOLO V1, data: 06/07/2023  Punti:  20
Alberto,Rossi, PUNTI: 147
Barbara,Verdi, PUNTI: 132
Carlo,Bianchi, PUNTI: 85 *PREMIUM*
Donatella,Bianchi, PUNTI: 110 *PREMIUM*

VOLO V1, data: 10/01/2024  Punti:  30
Donatella,Bianchi, PUNTI: 110 *PREMIUM*
Carlo,Bianchi, PUNTI: 85 *PREMIUM*

VOLO V2, data: 06/07/2023  Punti:  25
Alberto,Rossi, PUNTI: 147
Carlo,Bianchi, PUNTI: 85 *PREMIUM*

VOLO V2, data: 13/10/2023  Punti:  20

VOLO V3, data: 06/07/2023  Punti:  10
Barbara,Verdi, PUNTI: 132
Carlo,Bianchi, PUNTI: 85 *PREMIUM*
Donatella,Bianchi, PUNTI: 110 *PREMIUM*

VOLO V3, data: 13/10/2023  Punti:  40
Donatella,Bianchi, PUNTI: 110 *PREMIUM*
Elisa,Verdi, PUNTI: 142
Fabio,Verdi, PUNTI: 142

VOLO V4, data: 05/12/2023  Punti:  102
Alberto,Rossi, PUNTI: 147
Barbara,Verdi, PUNTI: 132
Elisa,Verdi, PUNTI: 142
Fabio,Verdi, PUNTI: 142
```

Figura 3: Linee stampate dalla applicazione `Esercizio1` (parte 2).

```
==================  VOLI (3)  ==================
VOLO V1, data: 06/07/2023  Punti:  20
Alberto,Rossi, PUNTI: 147
Barbara,Verdi, PUNTI: 132
Carlo,Bianchi, PUNTI: 145 *PREMIUM*
Donatella,Bianchi, PUNTI: 212 *PREMIUM*

VOLO V1, data: 10/01/2024  Punti:  30
Donatella,Bianchi, PUNTI: 212 *PREMIUM*
Carlo,Bianchi, PUNTI: 145 *PREMIUM*

VOLO V2, data: 06/07/2023  Punti:  25
Alberto,Rossi, PUNTI: 147
Carlo,Bianchi, PUNTI: 145 *PREMIUM*

VOLO V2, data: 13/10/2023  Punti:  20
Carlo,Bianchi, PUNTI: 145 *PREMIUM*

VOLO V3, data: 06/07/2023  Punti:  10
Barbara,Verdi, PUNTI: 132
Carlo,Bianchi, PUNTI: 145 *PREMIUM*
Donatella,Bianchi, PUNTI: 212 *PREMIUM*

VOLO V3, data: 13/10/2023  Punti:  40
Donatella,Bianchi, PUNTI: 212 *PREMIUM*
Elisa,Verdi, PUNTI: 142
Fabio,Verdi, PUNTI: 142

VOLO V4, data: 05/12/2023  Punti:  102
Alberto,Rossi, PUNTI: 147
Barbara,Verdi, PUNTI: 132
Elisa,Verdi, PUNTI: 142
Fabio,Verdi, PUNTI: 142
Donatella,Bianchi, PUNTI: 212 *PREMIUM*
```

Figura 4: Linee stampate dalla applicazione `Esercizio1` (parte 3).