

Simulazione Laboratorio di Programmazione - Edizione 2

24 gennaio 2014

NOTA Non è possibile usare le classi del package `prog.io` del libro di testo

L'obiettivo è scrivere un'applicazione per la gestione della prenotazione di voli. Ci sono due tipi di passeggeri: i passeggeri normali e i passeggeri *star* in possesso di una tessera fedeltà. Questi ultimi accumulano dei punti ad ogni prenotazione. Ovvero, quando un passeggero star effettua la prenotazione di un volo, il suo punteggio viene incrementato del bonus associato al volo.

Le classi da implementare sono quelle descritte sotto. Si noti che:

- Le classi, oltre ai metodi specificati, possono definire altri metodi che si ritengono utili.
- Le classi e i metodi definiti devono essere pubblici, i campi delle classi devono essere privati.
- Le operazioni di stampa vanno effettuate esclusivamente nelle classi che definiscono il metodo `main`.

Classe `Passeggero`

La classe `Passeggero` definisce un passeggero. Possiede un solo costruttore che costruisce un passeggero di cui viene specificato il nome.

Classe `PasseggeroStar`

Sottoclasse di `Passeggero` che rappresenta un passeggero star. Possiede un solo costruttore che costruisce un passeggero star di cui viene specificato il nome. Il punteggio iniziale del passeggero è zero; a ogni prenotazione il punteggio viene incrementato del valore del bonus associato al volo.

Classe `Volo`

La classe `Volo` rappresenta un volo. Un volo è descritto dal codice (stringa) e dal bonus (intero non negativo). Non vanno posti limiti al numero delle prenotazioni che possono essere effettuate.

La classe `Volo` possiede due costruttori:

- un costruttore che costruisce un volo di cui si specificano codice e bonus;
- un costruttore che costruisce un volo di cui si specifica solo il codice, in questo caso il bonus è zero.

La classe `Volo` definisce i metodi:

- `addPrenotazione(p)` Effettua su questo volo la prenotazione per il passeggero `p`; se `p` ha già una prenotazione non viene compiuta alcuna operazione (un passeggero non può avere più di una prenotazione per volo).
- `cerca(s)` Se un passeggero di nome `s` è prenotato sul volo, ritorna un riferimento ad esso, altrimenti ritorna `null`.

Esercizio 1

Scrivere un programma `Esercizio1` che costruisce un volo con codice `az123` e bonus 10. Il programma legge da **standard input** una sequenza di linee ognuna delle quali descrive una operazione da compiere sul volo `az123`. Ogni linea in input può avere uno dei seguenti formati:

```
P, nome  
S, nome  
Q, nome
```

Il primo carattere della linea definisce l'operazione da compiere.

Le linee che iniziano con `P` richiedono la prenotazione per il passeggero avente nome specificato.

Le linee che iniziano con `S` richiedono la prenotazione per il passeggero star avente nome specificato.

In entrambi i casi, la richiesta di prenotazione ha successo solamente se il passeggero non ha già prenotato il volo, altrimenti non viene effettuata alcuna operazione.

La linee che iniziano con `Q` verificano se il passeggero con nome specificato è prenotato sul volo corrente. Se si tratta di un passeggero star, ne stampano il punteggio corrente.

Si assume che l'input sia inserito correttamente. Non vanno fatte assunzioni sul numero di linee in input. Si noti che l'input va letto da *standard input*, e non da un file.

Esempio

Supponiamo che le linee sullo standard input siano

```
S, aldo  
P, barbara  
S, aldo  
Q, aldo  
Q, barbara  
Q, giorgio
```

Il programma `Esercizio1` deve stampare

```
Il passeggero aldo (10 punti) è prenotato sul volo (az123, BONUS 10)  
Il passeggero barbara è prenotato sul volo (az123, BONUS 10)  
Il passeggero giorgio non è prenotato sul volo (az123, BONUS 10)
```

Esercizio 2

Scrivere un programma `Esercizio2` che permette di gestire più di un volo. Più precisamente, oltre ai comandi descritti nell'Esercizio 1, il programma accetta anche linee della forma

`V, codice, bonus`

Il comando `V` definisce un volo avente codice e bonus specificati. Le operazioni di prenotazione `P` e `S` descritte nell'Esercizio 1 vanno effettuate sul volo corrente, ovvero l'ultimo volo definito dal comando `V`.

Si assume che la prima operazione richiesta è la definizione di un volo e che i voli definiti sono tutti distinti tra loro (ossia, hanno codici diversi). Non vanno posti limiti sul numero di voli che possono essere definiti.

Nota. Per poter stampare le prenotazioni su tutti i voli, occorre memorizzare i voli in un'opportuna struttura. Prestare attenzione alla gestione dei passeggeri star. Conviene definire una struttura che contiene tutti i passeggeri star che hanno richiesto una prenotazione in modo da memorizzare correttamente i loro punteggi.

Esempio

Supponiamo che le linee sullo standard input siano

```
V, az123, 10
P, Mario
S, Giuseppe
S, Alberto
Q, Alberto
V, xyz99, 25
P, Mario
S, Giuseppe
S, Davide
Q, Giuseppe
Q, Mario
```

Il programma `Esercizio3` deve stampare

```
Il passeggero Alberto (10 punti) è prenotato sul volo (az123, BONUS 10)
Il passeggero Giuseppe (35 punti) è prenotato sul volo (az123, BONUS 10)
Il passeggero Giuseppe (35 punti) è prenotato sul volo (xyz99, BONUS 25)
Il passeggero Mario è prenotato sul volo (az123, BONUS 10)
Il passeggero Mario è prenotato sul volo (xyz99, BONUS 25)
```