

Prova di Laboratorio di Programmazione — Edizione 2

31 gennaio 2014

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

L'obiettivo è implementare un sistema per la prenotazione di aule. Si noti che:

- Oltre ai metodi richiesti in ciascuna classe, è opportuno implementare altri metodi che si ritengono utili per svolgere l'esercizio nel modo più corretto ed elegante.
- I campi delle classi devono essere privati mentre le classi e i metodi definiti devono essere pubblici.
- Le operazioni di stampa vanno effettuate esclusivamente nelle classi che definiscono il metodo `main`.

Esercizio 1

Si implementi la classe seguente

Classe `Evento`.

Istanze di questa classe sono eventi caratterizzati da: giorno del mese, ora d'inizio e durata (numero intero di ore). Assumiamo che un evento inizi e finisca nello stesso giorno e duri sempre almeno un'ora.

Scrivere un programma `Esercizio1` che legge da **standard input** una sequenza di linee con i seguenti formati:

```
P,giorno,oraInizio,durata
S
```

Le linee che iniziano con `P` richiedono la creazione e memorizzazione di un evento con le caratteristiche specificate. Le linee che iniziano con `S` richiedono la stampa **in ordine cronologico** degli eventi memorizzati.

Un evento viene memorizzato se e solo se non è uguale a nessun evento memorizzato precedentemente, dove l'uguaglianza fra due eventi è definita nel modo seguente: due eventi sono uguali se e solo se uno di essi inizia quando l'altro non è ancora finito (per esempio, un evento che finisce alle dieci è uguale a un evento che inizia alle nove dello stesso giorno, ma non è uguale a un evento che inizia alle dieci di un giorno qualsiasi).

Si assuma che l'input sia inserito correttamente. Non vanno fatte assunzioni sul numero di linee in input. L'input va letto da **standard input**, possibilmente utilizzando la redirectione fornita da shell (si veda l'esempio sotto).

Esempio

Se il file `in1.txt` contiene le linee seguenti

```
P,3,15,2
P,1,17,2
P,1,13,4
P,1,11,3
P,1,9,1
S
```

allora il comando `java Esercizio1 < in1.txt` deve stampare

```
Giorno 1, dalle 9 alle 10
Giorno 1, dalle 13 alle 17
Giorno 1, dalle 17 alle 19
Giorno 3, dalle 15 alle 17
```

Esercizio 2

Si implementi la classe seguente:

Classe Aula.

Istanze di questa classe rappresentano aule che possono ospitare eventi. Un'aula è caratterizzata da un nome (p.es., aula Alfa) e da un elenco (di lunghezza non specificata) di eventi che hanno luogo in quell'aula.

La classe Aula definisce i metodi:

- prenota(e)
Prenota l'aula per evento fornito in argomento. Se l'evento è uguale a un altro già prenotato per quell'aula (dove l'uguaglianza è definita come nell'Esercizio 1) non viene effettuata alcuna operazione.
- isFree(giorno, oraInizio, durata)
Ritorna true se è possibile prenotare nell'aula un evento con le caratteristiche fornite in argomento, altrimenti ritorna false.
- occupazione(giorno)
Per il giorno fornito in argomento, ritorna il numero totale di ore durante le quali l'aula è occupata.

Scrivere un programma Esercizio2 che legge da **standard input** una sequenza di linee con i seguenti formati:

```
P,nomeAula,giorno,oraInizio,durata
F,giorno,oraInizio,durata
```

Le linee che iniziano con P richiedono prenotazione nell'aula specificata di un evento con le caratteristiche specificate (se l'evento è uguale ad un altro già prenotato per quell'aula non viene effettuata alcuna operazione). Le linee che iniziano con F richiedono la stampa di tutte le aule che possono prenotare un evento con le caratteristiche specificate. Vicino ad ogni aula disponibile, stampare il numero di ore di quel giorno durante le quali l'aula è già impegnata.

Nota: La classe Aula dovrà avere un campo adatto a contenere l'elenco degli eventi prenotati. Inoltre, dato che è necessario gestire le prenotazioni su più aule, anche le diverse aule utilizzate dovranno essere progressivamente memorizzate in una struttura adatta.

Si assuma che l'input sia inserito correttamente. Non vanno fatte assunzioni sul numero di linee in input. L'input va letto da **standard input**, possibilmente utilizzando la redirectione fornita da shell (si veda l'esempio sotto).

Esempio

Se il file in2.txt contiene le linee seguenti

```
P,Alfa,3,15,2
P,Beta,3,10,2
P,Beta,1,11,3
P,Alfa,3,9,2
P,Beta,1,9,1
P,Alfa,1,17,2
F,2,10,2
F,3,11,2
```

allora il comando `java Esercizio2 < in2.txt` deve stampare

```
Ricerca per il giorno 2 dalle 10 alle 12
Disponibile Alfa, occupata per 0 ore
Disponibile Beta, occupata per 0 ore
Ricerca per il giorno 3 dalle 11 alle 13
Disponibile Alfa, occupata per 4 ore
```

Istruzioni per la consegna

Consegnare soltanto i file seguenti (**non** i file `.class`):

Evento.java
Aula.java

Esercizio1.java
Esercizio2.java

come un unico file compresso creato col comando

```
> zip provalab Evento.java Aula.java Esercizio1.java Esercizio2.java
```

Eeguire l'upload del file all'indirizzo `upload.di.unimi.it`.

Non consegnare file che non compilano.