

Programmazione

II Compitino (Vers. A)

15 gennaio 2015

Cognome Nome
Matricola Anno di corso

- Nei seguenti quesiti, quando vi è richiesto di scrivere un programma, potete limitarvi al *corpo* del metodo main, assumendo se necessario che in sia una variabile di tipo Scanner, già dichiarate e inizializzate.

1. Scrivete un programma che legga un numero n , seguito da una sequenza di esattamente n stringhe; al termine, deve stampare nell'ordine il secondo carattere di ogni stringa, saltando quelle di lunghezza inferiore a due.

Ecco un esempio di esecuzione (le parti in grassetto sono state inserite dall'utente):

```
Quante stringhe: 4
Stringa 1:  mamma
Stringa 2:  p
Stringa 3:  xyp231
Stringa 4:  urka
ayr
```

(Svolgimento sul retro)

2. Scrivete un programma che legga un numero n , seguito da una sequenza di esattamente n interi positivi; al termine, deve stampare gli interi che hanno un numero pari di cifre (nell'ordine in cui sono stati introdotti).

Ecco un esempio di esecuzione (le parti in grassetto sono state inserite dall'utente):

```
Quanti interi:  6
Intero 1:  15
Intero 2:  152
Intero 3:  25
Intero 4:  1
Intero 5:  1500
Intero 6:  3
15
25
1500
```

3. Supponete di avere una classe astratta di nome `Contatore` le cui istanze rappresentano dei contatori. Questa classe (che ha un costruttore senza argomenti) prevede un metodo `valore()` che restituisce il valore attuale del contatore, e un metodo `aggiorna()` che aggiorna il contatore secondo regole che dipendono dalla concreta implementazione. La classe ha due sottoclassi concrete:
- la classe `ContatoreAdditivo`, in cui ogni aggiornamento del contatore aggiunge una quantità costante al contatore stesso; la classe ha un costruttore `ContatoreAdditivo(int valoreIniziale, int incremento)` che specifica il valore iniziale e l'incremento di cui il contatore viene aumentato ogni volta;
 - la classe `ContatoreMoltiplicativo`, in cui ogni aggiornamento del contatore moltiplica il contatore stesso per una quantità costante; la classe ha un costruttore `ContatoreMoltiplicativo(int valoreIniziale, int fattore)` che specifica il valore iniziale e il fattore per cui il contatore viene moltiplicato ogni volta.

Implementate nella classe `Contatore` un metodo `equals` che sovrascriva il metodo di `Object` e che consideri uguali due contatori se e solo se hanno lo stesso valore.

4. Scrivete una classe, di nome `SommaDiContatori` che estende `Contatore` e che funziona come segue:
- il suo costruttore `SommaDiContatori(Contatore[] c)` costruisce un contatore sulla base di un certo array di contatori passato come argomento, e che chiameremo *contatori coinvolti*;
 - il valore del contatore è, in ogni istante, la somma dei contatori coinvolti;
 - ogni volta che il contatore viene aggiornato, questo scatena un aggiornamento di tutti i contatori coinvolti.

5. Considerate la classe Portafogli le cui istanze rappresentano dei *portafogli*; un portafogli contiene, in ogni momento, un certo numero di banconote di ogni taglio, dove i tagli sono da 5, 10, 20, 50, 100, 500 EUR (ad esempio, quindi, un portafogli potrà contenere 2 banconote da 5, 0 banconote da 10, 1 banconota da 20 e 3 da 50). Nella classe c'è un metodo statico

- `public static int indice(int v)`

che dato un valore v uguale a 5, 10, 20, 50, 100, 500, restituisce 0, 1, 2, 3, 4, 5 (rispettivamente), e restituisce -1 su qualsiasi altro valore. C'è inoltre un metodo

- `public static int taglioDiIndice(int i) throws IllegalArgumentException`

che dato un indice i uguale a 0, 1, 2, 3, 4, 5, restituisce 5, 10, 20, 50, 100, 500 (rispettivamente), e solleva un `IllegalArgumentException` altrimenti (si tratta di un'eccezione non controllata).

Il portafogli ha un costruttore

- `public Portafogli()`

che costruisce un portafogli vuoto. La classe possiede dei metodi:

- `public void aggiungiBanconota(int v)`
- `public void togliaBanconota(int v)`

che rispettivamente aggiungono o tolgono una banconota di taglio dato (non fanno niente se v non è uno dei tagli ammessi). Il secondo metodo inoltre non fa niente se non ci sono banconote di quel taglio.

La classe possiede, fra gli altri, i metodi

- `public int quanteDiTaglio(int v)`
- `public int quanteInTutto()`
- `public int valoreTotale()`

che, rispettivamente, forniscono quante banconote sono presenti del taglio indicato (0 se il taglio non è uno di quelli ammissibili), quante banconote sono presenti in tutto e qual è il denaro complessivamente presente nel portafogli.

Scrivete un programma che operi come segue:

- crea un portafogli vuoto;
- chiede ripetutamente all'utente un intero (corrispondente al taglio di una banconota che viene inserita nel portafogli): se l'intero corrisponde a un taglio ammissibile, la banconota viene inserita; altrimenti, viene stampato un messaggio di errore;

- (c) il ciclo di lettura termina non appena l'utente inserisce 0;
- (d) alla fine del ciclo viene stampato il taglio per cui ci sono più banconote (scegliendo arbitrariamente se ci sono più tagli massimi).

Ecco un esempio di esecuzione (le parti in grassetto sono state inserite dall'utente):

```
Banconota:  50
Banconota:  5
Banconota:  15
Taglio inesistente!  -- inserimento ignorato
Banconota:  50
Banconota:  5
Banconota:  50
Banconota:  50
Banconota:  0
Il taglio da 50 è il più frequente
```

6. Considerate la seguente funzione definita ricorsivamente sugli interi:

$$f(x) = \begin{cases} f(x/2) + 1 & \text{se } x \text{ 'e' maggiore di } 1 \\ 1 & \text{altrimenti} \end{cases}$$

Scrivete un metodo statico con intestazione

```
public static int f( int x )
```

per il calcolo di f .

Rispondete inoltre alle seguenti domande:

- quanto vale $f(5)$?
- quanto vale $f(15)$?
- riuscite a dire quanto vale $f(x)$ in generale?

7. Scrivete la classe `Portafogli` descritta sopra. *Potete omettere i metodi `indice` e `taglioDiIndice`*

