

Programmazione

II Compitino (Vers. A)

17 dicembre 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 l'ultimo carattere di ciascuna stringa, dall'ultima alla prima, saltando le stringhe di lunghezza inferiore a 3.

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

```
Quante stringhe: 6
Stringa 1: mammone
Stringa 2: p
Stringa 3: xyp231
Stringa 4: urka
Stringa 5: ur
Stringa 6: nannanni
iale
```

(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 la somma degli interi ignorando le ripetizioni (cioè, anche se un intero compare molte volte deve essere sommato una volta sola).

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: 15
Intero 4: 1
Intero 5: 1500
Intero 6: 15
1668
```

3. Supponete di avere una classe astratta di nome Cane le cui istanze rappresentano dei cani. Nella classe Cane sono presenti i seguenti metodi:

- un metodo `abbaia()` che restituisce (sottoforma di stringa) il rumore che il cane fa quando abbaia; questo metodo è astratto perché il rumore dipende dalla razza;
- un metodo *statico* `contaMaiuscole(String x)` che restituisce il numero di lettere maiuscole¹ in una data stringa;
- un metodo `abbaiaPiuForteDi(Cane x)` che restituisce `true` se il cane su cui è invocato abbaia più forte del cane passato come argomento; si dice che un cane abbaia più forte di un altro se il rumore che fa abbaiano contiene più lettere maiuscole.

Implementate la classe Cane.

¹La classe `Character` contiene un metodo statico `isUpperCase(char c)` che restituisce `true` se il carattere passato come argomento è una lettera maiuscola, `false` altrimenti.

4. Scrivete una classe, di nome Chihuahua, che estenda la classe Cane con i seguenti costruttori e metodi
- un costruttore che riceve il peso del cane in grammi (un valore double);
 - il metodo abbaia (il rumore che fa un chihuahua è *caiCAIcai*);
 - un metodo peso() che restituisce il peso;
 - il metodo toString() che restituisce la stringa *chihuahua (xxx g)* dove *xxx* sia il peso;
 - il metodo equals (facendo overriding del metodo della classe Object) che stabilisce che un chihuahua è uguale a un altro chihuahua purché abbiano lo stesso peso.

5. Scrivete una classe, di nome Molosso, che estenda la classe Cane con i seguenti costruttori e metodi
- un costruttore che riceve l'altezza del cane in centimetri (un valore double) e una stringa che rappresenta il colore del molosso;
 - il metodo abbaia (il rumore che fa un molosso è *BAUBAUBAU*);
 - un metodo altezza() che restituisce l'altezza;
 - un metodo colore() che restituisce il colore;
 - il metodo toString() che restituisce la stringa *molosso xxx (yyy cm)* dove *xxx* sia il colore e *yyy* l'altezza;
 - il metodo equals (facendo overriding del metodo della classe Object) che stabilisce che un molosso è uguale a un altro se il loro colore è stesso, *indipendentemente dall'altezza*.

6. Considerate la classe `Rubrica` le cui istanze rappresentano delle rubriche di contatti. La classe possiede, fra gli altri, i seguenti metodi di istanza:

- `int size()`: restituisce il numero di contatti presenti nella rubrica;
- `Contatto get(int i)`: restituisce l'*i*-esimo contatto (i contatti sono numerati da 0 a $n - 1$, dove n sia il numero di contatti esistenti).

La classe `Contatto` ha, fra gli altri, i metodi:

- `String numero()`: restituisce il numero di telefono del contatto;
- `String nominativo()`: restituisce il nominativo del contatto.

Scrivete una classe `UtilitaRubrica` contenente due metodi statici:

- `conta`: data una `Rubrica` e una stringa rappresentante un numero di telefono, restituisce quanti contatti in rubrica hanno quel numero;
- `stampaDuplicati`: data una `Rubrica` stampa i nominativi di tutti i contatti che hanno numeri di telefono duplicati (cioè, il cui numero compare più di una volta nella rubrica). Per scrivere questo metodo, utilizzate il precedente.

7. Considerate la seguente funzione definita ricorsivamente sugli interi:

$$f(x) = \begin{cases} f(x/10) + x \bmod 10 & \text{se } x \geq 10 \\ x & \text{altrimenti,} \end{cases}$$

dove $x/10$ indica la divisione intera per 10 e $x \bmod 10$ è il resto della divisione intera per 10. Scrivete un metodo statico con intestazione

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

per il calcolo di f .

Rispondete inoltre alle seguenti domande:

- quanto vale $f(5000)$?
- quanto vale $f(15)$?
- riuscite a dire quanto vale $f(x)$ in generale?

8. Scrivete un metodo statico che, preso come argomento un array di Cane, restituisca un array di due elementi: il primo dei due è il cane che abbaia più forte, il secondo è quello che abbaia più piano.

