Programmazione
Preparazione al II Compitino

16 gennaio 2009

Cognome ..... Nome .....
Matricola .....

- Usate il retro dei fogli per scrivere lo svolgimento degli esercizi di programmazione.
- Nei seguenti quesiti, quando vi è richiesto di scrivere un programma, potete limitarvi al *corpo* del metodo main, assumendo se necessario che in e out siano due variabili di classe ConsoleInputManager e ConsoleOutputManager (rispettivamente), già dichiarate e inizializzate.

1. Scrivete un programma che legga un numero n, seguito da una sequenza di esattamente n stringhe; al termine, deve ristampare la prima metà di ciascuna stringa (per le stringhe di lunghezza dispari, si arrotondi la metà all'intero più piccolo).

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

   ```
   Quante stringhe:  4
   Stringa 1:  mamma
   Stringa 2:  pippo
   Stringa 3:  xyp231
   Stringa 4:  urka
   ma
   pi
   xyp
   ur
   ```

(Svolgimento sul retro)

2. Scrivete un programma che legga un numero n, seguito da una sequenza di esattamente n stringhe; al termine, deve stampare le stringhe non ripetute.

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

```
Quante stringhe:  6
Stringa 1:  genuflesso
Stringa 2:  mamma
Stringa 3:  pippo
Stringa 4:  mamma
Stringa 5:  ramoso
Stringa 6:  genuflesso
pippo
ramoso
```

(Svolgimento sul retro)

3. Considerate la classe Mail i cui oggetti corrispondono a dei *messaggi di posta elettronica*; questa classe ha un costruttore pubblico

   - public Mail(String mittente, String[] destinatari, String titolo, String testo)

   che costruisce un messaggio inviato dall'indirizzo mittente a una certa lista di indirizzi destinatari, e con un certo titolo e testo. La classe possiede, fra gli altri, i metodi

   - public String getMittente()
   - public boolean eFraIDestinatari(String dest)
   - public int getNumeroDestinatari()

   che, rispettivamente, forniscono il mittente, dicono se un dato indirizzo compare fra i destinatari e forniscono il numero di destinatari.

   Scrivete un programma che operi come segue:

   (a) chieda all'utente di inserire un numero intero, diciamo n;

   (b) dichiari un array di n mail e lo riempia con mail inserite dall'utente;

   (c) stampi i soli messaggi che contengano `rossi@yahoo.com` fra i destinatari e siano inviati a non più di 2 persone (la classe Mail dispone di un opportuno metodo toString()).

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

   Inserisci un intero: **4**

   Mittente: **xxx@valium.it**
   Quanti destinatari: **2**
   Destinatario: **aaa@vermeschifoso.com**
   Destinatario: **rossi@yahoo.com**
   Titolo: **Appuntamento**
   Testo: **Ci vediamo domani alle otto**

   Mittente: **bbb@valium.it**
   Quanti destinatari: **3**
   Destinatario: **ccc@magna.magna.it**
   Destinatario: **rossi@yahoo.com**
   Destinatario: **bbb@magna.magna.it**
   Titolo: **Riunione**
   Testo: **La riunione sarà nella sala rossa**

   Mittente: **bbb@valium.it**
   Quanti destinatari: **1**

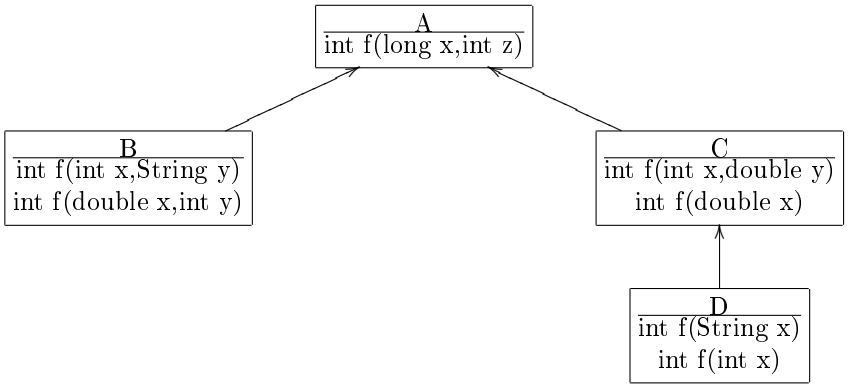
Destinatario: **rossi@yahoo.com**
Titolo: **Riunione 2**
Testo: **La riunione sarà nella sala grigia (non rossa!)**

Mittente: **bbb@valium.it**
Quanti destinatari: **1**
Destinatario: **aaa@vermeschifoso.com**
Titolo: **Baci**
Testo: **Vi voglio molto bene**

"Appuntamento" (da xxx@valium.com)
"Riunione 2" (da bbb@valium.com)

(Svolgimento in questa pagina)

4. Considerate la seguente gerarchia dei tipi (rappresentata mediante un diagramma UML):



e ipotizzate che tutte le classi abbiano un costruttore pubblico senza argomenti. Assumete le seguenti definizioni e inizializzazioni:

```
A a1, a2, a3, a4;
B b;
C c1, c2;
D d;
```

```
a1 = new A(); a2 = new B(); a3 = new C(); a4 = new D();
b = new B();
c1 = new C(); c2 = new D();
D d = new D();
```

Per ciascuno dei seguenti assegnamenti, dite se l'assegnamento è consentito, oppure se richiede (per poter essere compilato) un operatore di cast nel secondo membro (e in tal caso indicate l'assegnamento corretto), oppure se l'assegnamento non è consentito nemmeno con un cast (nel senso che anche con un cast produrrebbe un errore di compilazione o un'eccezione in esecuzione):

- b=a4: .....
- c=a2: .....
- a1=b: .....
- a2=c1: .....
- a1=b: .....
- c2=b: .....
- c1=(C)c2: .....
- c1=a4: .....

- c2=(A)a1: .....

5. Facendo riferimento al precedente diagramma UML e alle variabili definite e inizializzate come sopra, considerate le seguenti invocazioni di metodo. Per ciascuna di esse dovete dire se è consentita e in caso affermativo dovete indicare il nome della classe che contiene il metodo che verrà effettivamente eseguito:

   - d.f(3,`"pippo"`): .....
   - d.f(3,4.2): .....
   - a1.f(3,4): .....
   - a3.f(5,3): .....
   - a4.f(`"pippo"`): .....
   - b.f(3): .....
   - d.f(3.4): .....
   - a2.f(3,4): .....

6. Considerate la seguente funzione definita ricorsivamente sugli interi:

$$f(x) = \begin{cases} f(x+1) - f(x-1) & \text{se } x \text{ è pari} \\ x & \text{altrimenti} \end{cases}$$

Scrivete un metodo statico con intestazione

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

per il calcolo di f.

7. Scrivete la classe Mail con i seguenti metodi e costruttori:

   - **public** Mail(String mitt, String[] dest, String titolo, String testo): costruisce una mail con dato mittente, destinatari, titolo e testo;
   - **public** String getMittente(): restituisce il mittente;
   - **public boolean** eFraIDestinatari(String dest): restituisce **true** se e solo se dest compare fra i destinatari;
   - **public int** getNumeroDestinatari(): restituisce il numero dei destinatari;
   - **public** String getDestinatario(**int** i) **throws** IllegalArgumentException: restituisce l'i-esimo destinatario (con i compreso fra 0 e $n-1$, dove n è il numero di destinatari); se il valore i è negativo o maggiore di $n-1$, il metodo deve sollevare una IllegalArgumentException.

(Svolgimento sulla facciata seguente)

8. Definite una classe di nome MailCC che estende Mail e i cui oggetti corrispondono a delle e-mail che hanno anche dei destinatari in CC (cioè, che ricevono la e-mail solo per conoscenza). Questa classe deve avere:

   - un costruttore pubblico

     ```
     public MailCC(String mitt, String[] dest, String[] cc, String tit, String txt)
     ```

     che specifica oltre ai dati di una normale e-mail anche l'elenco dei destinatari in CC;
   - un metodo

     ```
     public int numeroCC()
     ```

     che restituisce il numero di destinatari in CC;
   - il metodo ereditato eFraIDestinatari(String dest) deve restituire **true** anche se dest compare fra i destinatari in CC.