Fondamenti di Architetture e Programmazione
III Compitino (Vers. B)

27 gennaio 2006

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 una sequenza di esattamente 10 numeri interi e stampi solo quelli la cui rappresentazione esadecimale inizia con la cifra `A`; ricordiamo che il metodo statico toHexString(**int** x) della classe Integer restituisce la rappresentazione esadecimale dell'intero che gli viene passato come argomento.

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

   ```
   Intero 1:  2563
   Intero 2:  181
   Intero 3:  170
   ...

   2563
   170
   ...
   ```

(Svolgimento sul retro)

2. Supponete che esista una classe Paziente i cui oggetti corrispondono a dei pazienti di uno studio di dietologia; questa classe ha un costruttore pubblico

   ```
   public Paziente( String n, double p, double h )
   ```

   che costruisce un paziente con dato nominativo n, peso p (in chilogrammi) e altezza h (in centimetri), e un metodo pubblico

   ```
   public boolean isPiuPesanteDi( double x )
   ```

   che restituisce true se e solo se il peso la persona supera `x`.

   Scrivete un programma che operi come segue:

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

   (b) chieda all'utente di inserire un valore double (la soglia), diciamo r;

   (c) dichiari un array di N pazienti e lo riempia con dati inseriti dall'utente (che, per ciascun paziente, deve inserire il nominativo, il peso e l'altezza);

   (d) stampi i soli pazienti *il cui peso è maggiore di r* (potete assumere che la classe Paziente disponga di un opportuno metodo toString()), ciascuno accompagnato dal suo peso.

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

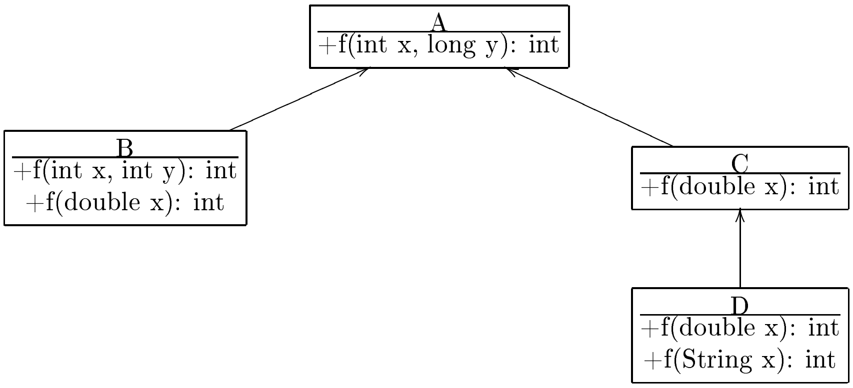
   ```
   Inserisci un intero: 3
   Inserisci una soglia: 75

   Nominativo: Mario Rossi
   Peso kg.: 75.4
   Altezza cm.: 172
   Nominativo: Giovanni Bianchi
   Peso kg.: 77.2
   Altezza cm.: 161
   Nominativo: Maria Marroni
   Peso kg.: 66
   Altezza cm.: 141

   Mario Rossi 75.4
   Giovanni Bianchi 77.2
   ```

(Svolgimento sul retro)

3. 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):

- `d=c1`: ....
- `a1=c1`: ....
- `c1=a3`: ....
- `b=c1`: ....
- `b=a3`: ....
- `c2=d`: ....
- `a1=b`: ....

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

5. La classe Math contiene un metodo statico

```
public static double pow(double a, double b)
```

che dati due double `a` e `b` restituisce a^b. Avete una variabile intera `i` e volete determinare se elevandola alla 7a si ottiene un valore maggiore di 10, mettendo il risultato (true o false) nella variabile booleana `z`; scrivete l'istruzione di assegnamento a `z`:

6. Considerate la seguente funzione definita ricorsivamente sugli interi:

$$f(x) = \begin{cases} x & \text{se } x \text{ è multiplo di 4} \\ x + f(x-1) & \text{altrimenti} \end{cases}$$

Scrivete un metodo statico con intestazione

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

per il calcolo di f.

7. Definite una classe di nome Paziente i cui oggetti corrispondono a dei pazienti di uno studio di dietologia; questa classe deve avere:

   - un costruttore pubblico

     ```
     public Paziente( String n, double p, double h )
     ```

     che specifica il nominativo, il peso (in kg.) e l'altezza (in cm.);
   - un costruttore pubblico

     ```
     public Paziente( String n, double p )
     ```

     che assume che l'altezza sia pari a 1.72;
   - un metodo

     ```
     public boolean isPiuPesanteDi( double x )
     ```

     che determina se la persona su cui è invocato ha un peso maggiore di x;
   - un metodo

     ```
     public boolean stessoPeso( Paziente p )
     ```

     che restituisce true se e solo se il paziente su cui è invocato e p hanno lo stesso peso.

(Svolgimento sulla facciata seguente)

8. Definite una classe di nome PazienteAmmalato che estende Paziente e i cui oggetti corrispondono a dei pazienti con una malattia. Questa classe deve avere:

   - un costruttore pubblico

     ```
     public PazienteAmmalato( String n, String m, double p, double h )
     ```

     che specifica il nominativo, la malattia, il peso (in kg.) e l'altezza (in cm.);

   - un metodo

     ```
     public boolean stessaMalattia( PazienteAmmalato p )
     ```

     che restituisce true se e solo se il paziente su cui è invocato e p hanno la stessa malattia.