

Informatica Generale

28 aprile 2005

Cognome Nome
Matricola

- 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 iniziate.

1. Per ciascuna delle seguenti espressioni booleane, dite su quante righe della tabella di verità il valore dell'espressione è *vero* ($\wedge = \text{AND}$, $\vee = \text{OR}$, $\oplus = \text{XOR} = \text{OR ESCLUSIVO}$, $\neg = \text{NOT}$):

- (a) $(x \vee \neg x) \wedge (x \vee y)$:
- (b) $\neg x \vee (y \wedge \neg(x \vee \neg y))$:
- (c) $(x \vee y) \oplus (z \oplus \neg x)$:

(Usate il retro per le tabelle di verità)

2. Scrivete un (frammento di) programma Java che operi come segue:

- legga un intero n da tastiera;
- legga n stringhe, memorizzandole in un array a ;
- stampi le stringhe immesse in ordine inverso, limitandosi a quelle che iniziano con il carattere 'a'.

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

```
Quante stringhe: 8
Stringa 0: aiuto
Stringa 1: aiutatemi
Stringa 2: vi
Stringa 3: prego
Stringa 4: ad
Stringa 5: uscire
Stringa 6: di
Stringa 7: qui
```

```
ad
aiutatemi
aiuto
```

(Svolgimento sul retro)

3. Scrivete un (frammento di) programma Java che operi come segue:

- legga degli interi non negativi; ogni volta che l'utente inserisce un intero negativo, il programma deve segnalare un messaggio di errore, e chiedere all'utente di ripetere l'inserimento (finché il numero inserito non sia ≥ 0);
- termini alla lettura di 0;
- al termine, stampi il numero dei valori inseriti che si discostano dalla media per più di 5.

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

```
Valore 0:  5
Valore 1: -4
Errore di inserimento; ripeti, prego
Valore 1: -3
Errore di inserimento:  ripeti, prego
Valore 1: 15
Valore 2:  5
Valore 3:  1
Valore 4:  9
Valore 5:  0
```

Risultato: 2

In questo esempio la media è $(5 + 15 + 5 + 1 + 9)/5 = 35/5 = 7$ e quindi i valori che si discostano dalla media per più di 5 sono 15 ($|15 - 7| = 8 > 5$) e 1 ($|1 - 7| = 6 > 5$).

Suggerimento: La classe `Math` contiene un metodo statico di nome `abs` che, preso come argomento un intero, ne restituisce il valore assoluto. La sua segnatura è:

```
public static int abs(int x)
```

(Svolgimento sul retro)

4. Considerate la classe `Merendina` i cui oggetti corrispondono a tipi di merendine; questa classe ha un costruttore pubblico

```
public Merendina(String nome, int calorie)
```

che costruisce una merendina con dato nome e numero di calorie, e i metodi

```
public boolean ePiuCaloricoDi(Merendina m)
public String getNome()
```

che dicono, rispettivamente, se una merendina è più calorica di una merendina `data`, e restituiscono il nome della merendina.

Scrivete un programma che operi come segue:

- (a) chieda all'utente di inserire un numero intero, diciamo n ;
- (b) dichiari un array di n merendine e lo riempia con merendine inserite dall'utente;
- (c) stampi il nome della merendina più calorica.

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

Inserisci un intero: **3**

Nome: **Ballotona Cremosì**

Cfu: **700**

Nome: **Fiesta Gorgonzola**

Cfu: **500**

Nome: **Oveto Tartufato con Sorpresa**

Cfu: **350**

La merendina più calorica è **Ballotona Cremosì**

(Svolgimento sul retro)

5. Assumete che x e y siano variabili `int`, e che il loro valore *prima di ciascuno dei seguenti assegnamenti* sia rispettivamente 1 e 2. Dite quale sarà il loro valore *dopo* gli assegnamenti indicati:

Assegnamento	x	y
<code>x+=y++</code>		
<code>x=y*x</code>		
<code>x=(y=x+1)+1</code>		
<code>x=(x<3)? (x++==++x? x+1:y+1) : --x</code>		
<code>x=(y*y)/++x</code>		
<code>x=(y==x)</code>		

6. Scrivete il codice di una classe di nome `AtletaOlimpico` le cui istanze rappresentano atleti che partecipano alle olimpiadi. Ogni atleta è caratterizzato da una nazione di appartenenza e da una specialità: questi due dati sono espressi mediante due istanze di due opportune classi, che dovete supporre pre-esistenti, di nome `Nazione` e `Specialita`.

Dovete fornire i seguenti costruttori e metodi pubblici:

- `AtletaOlimpico(String nome, Nazione naz, Specialita spec)`: crea un nuovo atleta con i dati specificati;
- `void vinceOro()`: indica che l'atleta ha vinto una nuova medaglia d'oro (all'atto della costruzione, l'atleta non aveva alcuna medaglia);
- `void vinceArgento()`: indica che l'atleta ha vinto una nuova medaglia d'argento;
- `void vinceBronzo()`: indica che l'atleta ha vinto una nuova medaglia di bronzo;
- `String medaglie()`: restituisce una stringa che descrive quante medaglie ha vinto l'atleta;
- `boolean eMeglioDi(AtletaOlimpico x)`: confronta l'atleta con un altro atleta `x` sulla base del numero di medaglie vinte; più precisamente, si assume che ogni medaglia d'oro valga 3 punti, ogni medaglia d'argento 2 e ogni medaglia di bronzo 1 punto, e si sommano i punti totali per confrontare i due atleti;
- `boolean stessaSpecialita(AtletaOlimpico x)`: restituisce `true` se i due atleti gareggiano nella stessa specialità (si assume che la classe `Specialita` abbia un metodo `equals` con l'usuale segnatura).

(Svolgimento sul retro)