Corso di Programmazione
Anno Accademico 2014–2015
Prof. Giovanni Pighizzini

Note ed esercizi aggiuntivi

Gli esercizi proposti sono utili per rivedere gli esempi riportati, che sono stati sviluppati e discussi in dettaglio a lezione.

5. Ancora metodi statici e cicli, ciclo `for`

Esempio. Calcolo del numero di lettere maiuscole in una stringa letta da input.

```
// VERSIONE 1: ciclo while

import prog.io.*;

class ContaMaiuscole {
  public  static void main(String[] args) {
    ConsoleInputManager in = new ConsoleInputManager();
    ConsoleOutputManager out = new ConsoleOutputManager();

    int nMaiuscole = 0;

    //lettura e calcolo
    String s = in.readLine("Inserisci una stringa ");

    int i = 0;
    int lung = s.length();
    while (i < lung) {
      char c = s.charAt(i);
      if (Character.isUpperCase(c))
        nMaiuscole = nMaiuscole + 1;
      i = i + 1;
    }

    //comunicazione risultato
    out.println("La stringa " + s + " contiene " +  nMaiuscole +
                " lettere maiuscole");
  }
}
```

Note

- Osservate che la condizione della selezione è il risultato fornito dal metodo statico `isUpperCase` di `Character`. L'invocazione dei metodi statici deve essere effettuata utilizzando il nome della classe che fornisce il metodo al posto del riferimento all'oggetto, utilizzato invece nel caso di invocazione dei metodi non statici.

- Osservate la struttura del ciclo, basato sulla variabile `i`, inizializzata a `0`, incrementata a ogni iterazione, sino a raggiungere il valore della lunghezza: in una situazione come questa è opportuno utilizzare un ciclo `for`, come mostrato nel codice seguente.

```
// VERSIONE 2: ciclo for
import prog.io.*;

class ContaMaiuscole {
  public  static void main(String[] args) {
    ConsoleInputManager in = new ConsoleInputManager();
    ConsoleOutputManager out = new ConsoleOutputManager();

    int nMaiuscole = 0;

    //lettura e calcolo
    String s = in.readLine("Inserisci una stringa ");

    for (int i = 0; i < s.length(); i = i + 1) {
      char c = s.charAt(i);
      if (Character.isUpperCase(c))
        nMaiuscole = nMaiuscole + 1;
    }

    //comunicazione risultato
    out.println("La stringa " + s + " contiene " +  nMaiuscole +
                " lettere maiuscole");
  }
}
```

Modifichiamo l'esempio in modo che conti il numero di lettere maiuscole in una sequenza di stringhe lette da input, anziché in una stringa sola. L'inserimento della stringa vuota indica la fine della sequenza.

```
import prog.io.*;

class ContaMaiuscole {
  public  static void main(String[] args) {
    ConsoleInputManager in = new ConsoleInputManager();
    ConsoleOutputManager out = new ConsoleOutputManager();

    int nMaiuscole = 0;

    //lettura e calcolo
    String s = in.readLine("Inserisci una stringa (<invio> per terminare) ");
    while (!s.equals("")) {
      //esame della stringa
      for (int i = 0; i < s.length(); i = i + 1) {
        char c = s.charAt(i);
        if (Character.isUpperCase(c))
          nMaiuscole = nMaiuscole + 1;
      }
      s = in.readLine("Inserisci una stringa (<invio> per terminare) ");
    }
```

```
    //comunicazione risultato
    out.println("Le stringhe lette contengono in totale " +  nMaiuscole +
                " lettere maiuscole");
  }
}
```

Note
Si osservi l'uso dei due cicli innestati: quello esterno per la lettura e l'esame delle stringhe, quello interno per l'esame di ciascuna stringa.

Esercizio 5.1
Modificate il codice dell'esempio precedente in modo che comunichi anche:

- il numero totale di stringhe lette,
- il numero medio di lettere maiuscole presenti nelle stringhe,
- la stringa, tra quelle lette, che contiene il maggior numero di lettere maiuscole.

Esercizio 5.2
Scrivete un'applicazione che legga un intero `n`, generi `n` numeri interi a caso, compresi tra `0` e `1000`, visualizzando ciascuno di essi in cifre, in lettere come numero cardinale, in lettere come numero ordinale. Se ad esempio viene letto `4`, una possibile esecuzione produrrà:

```
10 dieci decimo
8 otto ottavo
4 quattro quarto
1000 mille millesimo
```

Utilizzate la classe `Intero` del package `prog.utili` e la classe `Random` del package `java.util`.

Esercizio 5.3
Scrivete un'applicazione che legga un numero intero `n` e comunichi il risultato della somma di frazioni `1/1 + 1/2 + ...  + 1/n`. Nel caso venga inserito un numero negativo, l'applicazione deve chiedere all'utente di ripetere l'inserimento.

Esercizio 5.4
Scrivete un'applicazione che legga un numero `n` e produca un elenco di tutte moltiplicazioni dei numeri da `1` a `n`. Si consiglia di utilizzare due cicli for innestati. Se ad esempio viene fornito il valore `4`, l'output prodotto dovrà essere:

```
1 * 1 = 1
1 * 2 = 2
1 * 3 = 3
1 * 4 = 4
2 * 1 = 2
2 * 2 = 4
2 * 3 = 6
2 * 4 = 8
3 * 1 = 3
3 * 2 = 6
3 * 3 = 9
3 * 4 = 12
4 * 1 = 4
```

```
4 * 2 = 8
4 * 3 = 12
4 * 4 = 16
```

Esercizio 5.5
Ripetete l'esercizio precedente, producendo una tabella delle moltiplicazioni. In una prima versione potete produrre una tabella come la seguente, senza preoccuparvi degli allineamenti:

```
  1 2 3 4
1 1 2 3 4
2 2 4 6 8
3 3 6 9 12
4 4 8 12 16
```

Migliorate poi la visualizzazione, facendo in modo che i numeri di ciascuna colonna risultino allineati a destra, sulla base dello spazio richiesto dal numero più grande (che sarà quello sull'ultima riga):

```
  1 2  3  4
1 1 2  3  4
2 2 4  6  8
3 3 6  9 12
4 4 8 12 16
```

Per ottenere l'allineamento possono essere utili le indicazioni date per l'esercizio 4.11. In alternativa, la classe `ConsoleOutputManager` e la classe `PrintStream` delle librerie standard (a cui appartiene l'oggetto che rappresenta il monitor riferito da `System.out`) offrono un metodo `printf` che permette una visualizzazione "formattata" secondo lo stile della funzione `prinf` del linguaggio C. Consultate la documentazione per conoscere i dettagli.