Impariamo a programmare in Java™

Lezione 5

Gr.it

Operazioni Aritmetiche

Quando entrambi gli operandi sono numeri interi, la divisione ha una caratteristica particolare, che può essere utile ma che va usata con attenzione

- Calcola il quoziente intero, scartando il resto!

7/4 → 1

Il resto della divisione tra numeri interi può essere calcolato usando l'operatore %, che non esiste in algebra ed il cui simbolo è stato scelto perchè è simile all'operatore di divisione.

7%4 → 3

Esercizio

Proviamo a **creare un programma** che dato un **valore in euro** stampi a video tale valore scomposto tra parte intera e centesimi, ad esempio: “2 euro e 35 centesimi”.

Euro: 2,35

Incremento di una variabile

L'***incremento*** di una variabile è l'operazione che consiste ***nell'aumentarne il valore di uno***

```
int counter = 0;
counter = counter + 1;
```

Questa operazione è talmente comune nella programmazione, che il linguaggio Java fornisce un operatore apposito per l'incremento

```
counter ++;
```

E per il decremento

```
counter --;
```

Ricevere Dati in Ingresso

I dati in ingresso ai programmi

I programmi visti finora non sono molto utili, visto che eseguono ***sempre la stessa elaborazione ad ogni esecuzione.***

Il programma Coins1 rappresenta sempre il medesimo borsellino…

- se si vuole che calcoli il valore contenuto in un diverso borsellino, è necessario modificare il codice sorgente (in particolare, le inizializzazioni delle variabili) e compilarlo di nuovo.

I programmi utili hanno bisogno di ***ricevere dati in ingresso dall'utente***.

L'*input standard* dei programmi

Il modo più semplice e immediato per fornire dati in ingresso ad un programma consiste ***nell'utilizzo della tastiera***

- altri metodi fanno uso del mouse, del microfono

Tutti i programmi Java hanno accesso al proprio ***output standard***, tramite l'oggetto ***System.out*** (di tipo **PrintStream**) definito nella classe **System**

Analogamente, l'interprete java mette a disposizione dei programmi in esecuzione il proprio input standard (***flusso di input***), tramite l'oggetto ***System.in*** di tipo **InputStream**

L'*input standard* dei programmi

Sfortunatamente, la classe InputStream ***non possiede metodi comodi*** per la ricezione di dati numerici e stringhe

- InputStream ha un metodo in grado di leggere un byte alla volta, oppure sequenze di byte, ma non altri dati.
- PrintStream ha invece il ***comodissimo*** metodo print

Per ovviare a questo inconveniente, useremo la classe ***Scanner***, che ***fa parte della libreria standard***

È inclusa a partire dalla versione 5.0

La classe ***Scanner*** appartiene al pacchetto ***java.util***

Lo scopo della classe ***Scanner*** è quello di fornire una comoda interfaccia all'oggetto System.in

Leggere l’input con la classe Scanner

La classe Scanner

Prima di tutto bisogna creare un ***oggetto*** della classe **Scanner** usando l'istruzione:

```
Scanner console = new Scanner(System.in);
```

- È possiblie creare un oggetto di questo tipo a partire da un qualsiasi flusso di input

Dopo aver creato uno scanner ("scansionatore") è possibile usarne i metodi di lettura

Durante l'esecuzione dei metodi di lettura di **Scanner**, ***il programma si ferma ed attende*** l'introduzione dei dati in ingresso da tastiera

- Attesa che termina quando l'utente batte il tasto "*Invio*"

I metodi nextInt e nextDouble

```
Scanner console = new Scanner(System.in);
```

Il metodo ***nextInt*** restituisce un valore numerico di tipo ***int***

È consigliabile inserire un messaggio di richiesta dati prima di invocare un metodo di Scanner

```
System.out.print(“Inserisci il dato: ”);
int number = console.nextInt();
```

Il metodo ***nextDouble*** restituisce un valore numerico in virgola mobile

```
System.out.print(“Inserisci il dato: ”);
double fp = console.nextDouble();
```

Esempio

```
... // qui manca qualcosa, che vedremo…
public class Coins4
{  public static void main(String[] args)
   {  Scanner c = new Scanner(System.in);
    System.out.println("Quante lire?");
    int lit = c.nextInt(); // Lire italiane
    System.out.println("Quanti Euro?");
    double euro = c.nextDouble(); // Euro
    // stampa il valore totale
    System.out.print("Valore totale in Euro ");
    System.out.println(euro + lit/1936.27);
   }
}
```

I metodi nextLine e next

Il metodo ***nextLine*** restituisce un oggetto di tipo ***String*** che contiene l'intera riga introdotta dall'utente (fino alla pressione ***del carattere "Invio"***, carattere di "fine riga" o "andata a capo")

```
String line = console.nextLine();
```

Il metodo ***next*** restituisce un oggetto di tipo ***String*** che contiene la parola successiva, cioè una sequenza di caratteri terminata da uno "spazio", o da un carattere di tabulazione, o da un carattere di "fine riga"

```
String word = console.next();
```

Esempio

```
import java.util.Scanner;
public class MakePassword2
{  public static void main(String[] args)
   {   Scanner c = new Scanner(System.in);
    System.out.println("Inserire il Nome");
    String firstName = c.nextLine();
    System.out.println("Inserire il Cognome");
    String lastName = c.nextLine();
    System.out.println("Inserire l'eta'");
    int age = c.nextInt();
    String initials = firstName.substring(0,1) +
    lastName.substring(0,1);
    String pw = initials.toLowerCase()+age;
    System.out.println("La password e' " + pw);
   }
}
```