Corso di Laurea in Ingegneria Informatica e Automatica
A. A. 2016/2017

Programmazione orientata agli oggetti

Luca Iocchi, Massimo Mecella

L2.1
Metodi statici, passaggio di parametri, ricorsione

Sommario

- Metodi statici
  - invocazione
  - definizione
- Passaggio di parametri
- Metodi ricorsivi

Invocazione di metodi statici

Sintassi

nomeClasse.nomeMetodo (*parametri attuali*)

Esempio

```
double x = Math.cos(alpha);
```

Funzioni matematiche

- La classe Math del pacchetto java.lang contiene molti metodi statici che eseguono varie funzioni matematiche.

Esempio

```
double t = Math.cos(alpha) + Math.sqrt(delta);
```

Numeri pseudo-casuali

- Numeri pseudocasuali sono generabili con il metodo statico random() della classe Math.
- random() restituisce un valore di tipo double nell'intervallo [0,1)
- Per calcolare un valore intero bisogna effettuare un cast esplicito.

Esempio

```
int x =(int) (Math.random () * 6 + 1); // numero intero tra 1 e 6
```

Definizione di metodi statici

Sintassi

```
intestazione  blocco_istruzioni

public static tipoRisultato nomeMetodo ( parametri formali )
{
    …
}
```

Esempio

```
public static int somma (int a, int b)
{
    …
}
```

tipoRisultato può essere un tipo primitivo, una classe, oppure void

Istruzione return

Sintassi

```
return espressione;
```

Esempio

```
public static int somma (int a, int b)
{
    return a+b;
}
```

Overloading di metodi

- L'***overloading di metodi*** è presente quando si usano gli stessi nomi per indicare metodi diversi
- La ***segnatura*** di ciascun metodo coinvolto dal processo di ***overloading*** deve essere **unica**
- La ***segnatura*** include il ***numero***, ***tipo*** e l'***ordine dei parametri***
- Il tipo del valore di ritorno di un metodo **non** fa parte della ***segnatura*** ma del ***prototipo***
- Il compilatore determina la versione del metodo da invocare analizzando i parametri

Esempi di metodi statici

```
public static String generaSaluto()
{
    return "Buon giorno";
}

public static void stampaSaluto()
{
    String s = generaSaluto();
    System.out.println(s);
}

public static void stampaSaluto(String nome)
{
    System.out.println(generaSaluto()+" "+nome);
}
```

Esempi di metodi statici

```
public class ProvaMetodi {

    public static String generaSaluto()
    …
    public static void stampaSaluto()
    …
    public static void stampaSaluto(String nome)
    …

    public static void main (String args[])
    {
       stampaSaluto();
       stampaSaluto("Mario");
    }
}
```

Variabili locali dei metodi

Le variabili locali definite all'interno di un metodo sono visibili solo nel metodo stesso.

Esempio

```
void f(int x) {
   int y = …;
   ….
   System.out.println(y);
}

void g() {
   // y non è visibile
}
```

Passaggio dei parametri

- Tipi di dato primitivi

  passaggio per valore

- Riferimenti ad oggetti

  passaggio per riferimento

Passaggio dei parametri

```
public class PassaggioParametri {

    public static void stampaInfoPersona (String nome, int eta)
    {
          System.out.println(nome+" "+eta);
    }

    public static void main (String args[])
    {
       String s = "Mario Rossi"; int eta = 32;
       stampaInfoPersona (s,eta);
    }
}
```

Metodi ricorsivi

Java usa un meccanismo standard di attivazione dei metodi (statici) basato sulla pila dei record di attivazione.

Si possono quindi definire metodi ricorsivi nella maniera standard già vista per altri linguaggi di programmazione.

Esempio

```
public static int fattoriale(int n) {
   if (n==0)
      return 1;
   else
      return n * fattoriale(n-1);
}
```

Variabili statiche con valori costanti

Valori costanti possono essere definiti all'interno di una classe (fuori dalle definizioni dei metodi) con la definizione di una variabile che assume un valore con non può essere modificato (final)

```
public static final tipo nomeVariabile=valore;
```

Esempio

```
public class Utilita {

   public static final int ZERO=0;
   ....
```

Variabili statiche con valori costanti

Per accedere alle variabili statiche (costanti) di una classe si usa la forma

nomeClasse.nomeVariabile

Usando il modificatore di accesso public la variabile è accessibile da qualsiasi altra classe.

Esempi

Math.PI denota la costante π definita nella classe Math

Utilita.ZERO (vedi slide precedente) contiene il valore 0

Variabili statiche di una classe

E' anche possibile definire variabili statiche di una classe (variabili di classe) il cui contenuto può variare a run-time.

Si definiscono all'interno di una classe con la sintassi

public static tipo nomeVariabile;

Una variabile di classe pubblica può essere letta e modificata (nomeClasse.nomeVariabile) da qualsiasi metodo di qualsiasi classe (variabile globale).

Organizzazione dei programmi Java

Un programma Java è normalmente organizzato in più classi.
Ogni classe pubblica deve essere memorizzata in un file.
Il nome del file deve coincidere con il nome della classe (con estensione .java).

Esempio

```
// file Utilita.java
public class Utilita { … }

// file Programma.java
public class Programma { … }
…
  Utilità.metodo(); // usa metodi della classe Utilità
```

Esercizio 2.1.1

Realizzare un programma equivalente all'esercizio 1.2.2 implementando un metodo statico che trasforma una stringa in modo tale che il primo carattere sia maiuscolo e gli altri siano minuscoli. Applicare quindi il metodo per la correzione sia del nome che del cognome, eliminando quindi duplicazioni di codice.

Nota:
Questa soluzione è più **modulare** rispetto alla versione precedente.

Esercizio 2.1.2

Fornire una versione più **efficiente** del programma per il calcolo del massimo comun divisore usando il metodo di Euclide e implementandolo in modo ricorsivo.

Metodo di Euclide:

Dati due valori interi positivi x e y (x>=y) e indicato con r il resto della divisione tra interi x / y:

- mcd(x,y) = y, se r = 0 (ovvero, x è multiplo di y)
- mcd(x,y) = mcd(y,r), se r != 0