

I Metodi

Fondamenti di Informatica A-K

Esercitazione 3

Introduzione al calcolatore e Java

Linguaggio Java, basi e controllo del flusso

I metodi: concetti di base

Stringhe ed array

Classi e oggetti, costruttori, metodi statici, visibilità

Eclipse, ereditarietà e polimorfismo

Collezioni Java

Esercizi d'esame

I metodi

- Alcune sequenze di istruzioni vanno ripetute più volte all'interno di un programma: pertanto, è utile poter scrivere tali sequenze una volta sola e farvi riferimento ogni volta che serve. I metodi assolvono a questa funzione.
- Un metodo raggruppa una sequenza di istruzioni che realizzano una funzionalità del programma e assegna loro un nome.
- Quando si usa un metodo, si dice che si invoca (o si chiama) il metodo stesso.

Esempio

```
public class Prova {  
    public static void scrivi() {  
        System.out.println("prova");  
    }  
  
    public static void main(String[] args) {  
        scrivi();  
    }  
}
```

Metodi void

- L'intestazione dei metodi che non restituiscono alcun valore, come visto nell'esempio **precedente**, vanno preceduti dalla clausola **void**.

```
public static void scrivi() {  
    System.out.println("prova");  
}
```

Esercizio CalcolaArea

1. Creare una classe con **CalcolaArea** in cui sia presente:
2. Un metodo che calcola l'Area di un triangolo con base 5.34 e altezza 3.5 e stampi il risultato a video
3. Un metodo che calcola l'area di un rettangolo con base 4.3 e altezza 2.4 e stampi il risultato a video
4. Un metodo che calcola l'area di cerchio con raggio 2 e stampi il risultato a video
5. Una funzione **main** che invochi i tre metodi

Soluzione CalcolaArea

```
public class CalcolaArea {  
    public static void calcolaAreaTriangolo() {  
        System.out.println("Area Triangolo: " +  
((5.34*3.5)/2));  
    }  
    public static void calcolaAreaRettangolo() {  
        System.out.println("Area Rettangolo: " + (4.3*2.4));  
    }  
    public static void calcolaAreaCerchio() {  
        System.out.println("Area Cerchio: " + (3.14159*2*2));  
    }  
    public static void main(String[] args) {  
        calcolaAreaTriangolo();  
        calcolaAreaRettangolo();  
        calcolaAreaCerchio();  
    }  
}
```

Metodi che restituiscono un valore

- Anche la definizione di un metodo che restituisce un valore prevede due parti: intestazione e corpo.
- L'intestazione di un metodo che restituisce un valore è simile a quella di un metodo **void**, ma al posto della parola chiave **void** indica il nome del tipo di ritorno.
- Il corpo della definizione di un metodo che restituisce un valore è simile a quello di un metodo **void**, ma al suo interno deve contenere almeno un'istruzione **return**.

Esempio

```
public class Prova {  
    public static double piGrecoPer2() {  
        return 3.14159*2;  
    }  
    public static void main(String[] args) {  
        double val = piGrecoPer2();  
    }  
}
```

Esercizio CalcolaAreaReturn

- Modificare il programma precedente in modo tale che sia il main a stampare a video i risultati ma siano sempre i metodi ad effettuare i calcoli.

Soluzione CalcolaAreaReturn 1/2

```
public class CalcolaAreaReturn {  
    public static double calcolaAreaTriangolo() {  
        return (5.34*3.5)/2;  
    }  
    public static double calcolaAreaRettangolo() {  
        return 4.3*2.4;  
    }  
    public static double calcolaAreaCerchio() {  
        return 3.14159*2*2;  
    }  
}
```

Soluzione CalcolaAreaReturn 2/2

```
public static void main(String[] args) {  
    double areaT = calcolaAreaTriangolo();  
    System.out.println("Area Triangolo: " + areaT);  
    double areaR = calcolaAreaRettangolo();  
    System.out.println("Area Rettangolo: " + areaR);  
    double areaC = calcolaAreaCerchio();  
    System.out.println("Area Cerchio: " + areaC);  
}
```

Variabili locali ai metodi

Ricordare sempre:

1. Una variabile dichiarata all'interno di un metodo o più in generale in un blocco (parentesi graffe) è detta "locale" di tale metodo o blocco.
2. Le variabili locali possono essere usate esclusivamente all'interno del metodo o blocco in cui sono state definite.
3. Se due metodi o blocchi hanno variabili locali con lo stesso nome, si tratta comunque di due variabili distinte.

Esempio

```
public class Prova {  
    public static double piGrecoPerN( int n ) {  
        double var = 3.14159*n;  
        return var;  
    }  
    public static void main(String[] args) {  
        double perDue = piGrecoPerN(2);  
        double perTre = piGrecoPerN(3);  
    }  
}
```

Esercizio CalcolaAreaVar

- Modificare l'esercizio precedente assegnando il risultato del calcolo dell'area a delle variabili locali a ciascun metodo chiamate tutte area. Successivamente ritornare con la clausola return la variabile stessa.

Soluzione CalcolaAreaVar 1/2

```
public class CalcolaAreaVar {  
    public static double calcolaAreaTriangolo() {  
        double area = (5.34*3.5)/2;  
        return area;  
    }  
    public static double calcolaAreaRettangolo() {  
        double area = 5.34*3.5;  
        return area;  
    }  
    public static double calcolaAreaCerchio() {  
        double area = 3.14159*2*2;  
        return area;  
    }  
}
```


Soluzione CalcolaAreaVar 2/2

```
public static void main(String[] args) {  
    double areaT = calcolaAreaTriangolo();  
    System.out.println("Area Triangolo: " + areaT);  
    double areaR = calcolaAreaRettangolo();  
    System.out.println("Area Rettangolo: " + areaR);  
    double areaC = calcolaAreaCerchio();  
    System.out.println("Area Cerchio: " + areaC);  
}
```

Metodi con **parametri**

- Passare dei parametri ad un metodo ci consente di utilizzare lo stesso come una funzione matematica.
- In Java si può scegliere il nome dei parametri formali di un metodo indipendentemente dal fatto che il nome sia già stato usato in qualche altro metodo.
- Nella definizione di un metodo, un parametro formale viene descritto nell'intestazione, fra parentesi e dopo il nome del metodo.
- Quando viene chiamato un metodo, ogni suo parametro viene inizializzato con il valore dell'argomento corrispondente nell'invocazione del metodo (cosiddetta chiamata per valore).

Esempio

```
public class Prova {  
    public static double piGrecoPerN(int n) {  
        return 3.14159*n;  
    }  
    public static void main(String[] args) {  
        double perDue = piGrecoPerN(2);  
        double perTre = piGrecoPerN(3);  
    }  
}
```

Esercizio CalcolaAreaParam

- Modificare l'esercizio precedente affinché il calcolo delle aree possa essere effettuato su triangoli, rettangoli e cerchi con parametri qualsiasi.

Soluzione CalcolaAreaParam 1/2

```
public class CalcolaAreaParam {  
    public static double calcolaAreaTriangolo(double base, double altezza){  
        double area = (base*altezza)/2;  
        return area;  
    }  
    public static double calcolaAreaRettangolo(double base, double altezza){  
        double area = base*altezza;  
        return area;  
    }  
    public static double calcolaAreaCerchio(double raggio) {  
        double area = 3.14159*raggio*raggio;  
        return area;  
    }  
}
```

Soluzione CalcolaAreaParam 2/2

```
public static void main(String[] args) {  
    double areaT = calcolaAreaTriangolo(1,2);  
    System.out.println("Area Triangolo: " + areaT);  
    double areaR = calcolaAreaRettangolo(2);  
    System.out.println("Area Rettangolo: " + areaR);  
    double areaC = calcolaAreaCerchio(2);  
    System.out.println("Area Cerchio: " + areaC);  
}
```

Documentazione

- Un buon programma deve essere ben documentato, ma spesso ciò non viene fatto.
- Java fornisce uno strumento automatico per produrla a partire dai commenti scritti dal programmatore nel programma stesso.

```
/**  
 * Applicazione Java da linea di comando  
 * Stampa la classica frase di benvenuto  
 * @author SPZ  
 * @version 1.0, 04/05/2017  
 */  
public class Esempio {  
    public static void main(String args[]) {  
        System.out.println("Ciao Mondo!");  
    }  
}
```

Documentazione

- Per produrre la relativa documentazione, si invoca il comando fornendo in ingresso i file sorgenti (.java):

```
javadoc Esempio.java
```