

## Note ed esercizi aggiuntivi

### 14. Metodi, sottoprogrammi, passaggio dei parametri

*Esempio.*

```
import prog.io.*;

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

        int num = in.readInt("Numero? ");

        int quad = quadrato(num);
        int quarta = quadrato(quad);

        out.println("Quadrato: " + quad);
        out.println("Quarta potenza: " + quarta);
    }

    public static int quadrato(int x) {
        x = x * x;
        return x;
    }
}
```

*Note*

- Il metodo statico `quadrato` della classe `Test` viene richiamato due volte da `main` con argomenti differenti. In questo caso il metodo funge da *sottoprogramma* permettendo di suddividere il codice di `main` in codice più semplice (vista la brevità dell'esempio la semplificazione qui è del tutto irrilevante) e di sostituire parti di codice ripetute con chiamate al sottoprogramma che, grazie all'uso dei parametri, è in grado di effettuare la stessa operazione su dati differenti.
- Il linguaggio Java fornisce esclusivamente il *passaggio per valore*. Al momento della chiamata di un metodo vengono calcolati i valori degli argomenti e copiati nei rispettivi parametri. I parametri sono variabili locali dei metodi: ogni modifica a essi non ha effetto sugli argomenti forniti nella chiamata (le modifiche a `x` in `quadrato` non hanno effetti sugli argomenti `num` e `quad` utilizzati nelle due chiamate).

*Esempio.*

```
public class Alfa {
    private int x = 1;

    public void aggiungi(int k) {
        x = x + k;
    }

    public String toString() {
        return String.valueOf(x);
    }
}

public class Beta {
    public static void inc(Alfa a) {
        a.aggiungi(1);
    }

    public static void nuovo(Alfa a) {
        a = new Alfa();
    }

    public static void annulla(Alfa a) {
        a = null;
    }
}

class Prova {
    public static void main(String[] a) {
        Alfa r = new Alfa();
        System.out.println(r);
        Beta.inc(r);
        System.out.println(r);
        Beta.nuovo(r);
        System.out.println(r);
        Beta.annulla(r);
        System.out.println(r);
    }
}
```

*Note*

- La classe `Beta` fornisce alcuni metodi statici che ricevono tramite il parametro il riferimento a un oggetto di tipo `Alfa`.
- Poiché il passaggio dei parametri è per valore, al momento della chiamata di ognuno di questi metodi l'argomento fornito (un riferimento di tipo `Alfa`) è copiato nel corrispondente parametro. Ad esempio, quando nella classe `Prova` si chiama

```
Beta.inc(r)
```

il riferimento contenuto nella variabile `r` viene copiato nel parametro `a` del metodo. In tal modo, `a` si riferisce allo *stesso* oggetto cui si riferisce `r`. Pertanto:

- l'effetto di una modifica all'oggetto riferito da **a** è visibile all'esterno del metodo, in quanto l'oggetto permane al termine dell'esecuzione del metodo,
- l'effetto di una modifica alla variabile **a** *non produce alcun effetto* visibile all'esterno del metodo, in quanto il parametro **a** è una variabile locale del metodo, che viene distrutta al termine dell'esecuzione del metodo stesso.

**Esercizio 14.1**

Cosa stampa il metodo `main` della classe `Prova`? Per rispondere disegnate i dati presenti nella memoria della macchina e la loro evoluzione durante l'esecuzione. Dopo avere risposto fate eseguire l'applicazione sul vostro computer e confrontate il risultato ottenuto con le risposte che avete dato. Accertatevi di avere capito bene il meccanismo del passaggio dei parametri e la distinzione, fondamentale, tra oggetti e riferimenti a oggetti.