Corso di Programmazione
Anno Accademico 2013–2014
Prof. Giovanni Pighizzini

Note ed esercizi aggiuntivi

17. Struttura delle classi, ereditarietà e costruttori

Esempio.

```
public class K {
  private int x;
  private int y = 2;
  private static int z = 5;

  public void inc() {
    x++;
    y++;
    z++;
  }

  public String toString() {
    return "x == " + x + ", y == " + y;
  }

  public static int getZ() {
    return z;
  }

}
```

Note

- Ogni oggetto della classe `K` possiede due campi `x` e `y`, entrambi di tipo `int`. Il campo `z`, essendo statico, appartiene invece alla classe.

- Poiché non è indicato alcun costruttore, il compilatore aggiunge un costruttore privo di argomenti che inizializza i campi ai valori di default o ai valori indicati insieme alla dichiarazione. In questo caso il campo `x` viene inizializzato a `0`, mentre il campo `y` a `5`. Il costruttore privo di argomenti aggiunto dal compilatore, come prima operazione, richiama sempre il costruttore privo di argomenti della superclasse, in questo caso `Object`, per predisporre la parte ereditata (nel caso di `Object` la parte ereditata è sostanzialmente una scatola vuota, pertanto in questo caso la chiamata al costruttore della superclasse appare irrilevante).

```
public class H extends K {
  private int w = 4;

  public void inc() {
    super.inc();
    w++;
  }

  public String toString() {
    return super.toString() + ", w == " + w;
  }

}
```

Note

- Ogni oggetto della classe `H` possiede un campo `w`, oltre alla parte ereditata dalla superclasse `K`.
- Poiché non è indicato alcun costruttore, il compilatore aggiunge un costruttore privo di argomenti. Il costruttore chiama prima di tutto quello della superclasse `K` che predispone la parte di oggetto ereditata come indicato sopra. Successivamente assegna a `w` il valore `4`.

Esercizio 17.1
Considerate la seguente applicazione che utilizza le classi dell'esempio precedente.

```
class ProvaH {
  public static void main(String[] a) {
    H r = new H();
    System.out.println(r.toString());
    r.inc();
    System.out.println(r.toString());
    H t = new H();
    System.out.println(r.toString());
    t.inc();
    System.out.println(t.toString());
    System.out.println(K.getZ());
  }
}
```

Simulate l'esecuzione del metodo `main` disegnando l'evoluzione dei vari dati utilizzati (oggetti, campi statici, variabili locali, parametri). Quali risultati vengono visualizzati? Dopo avere risposto, fate eseguire l'applicazione sul vostro computer e confrontate i risultati ottenuti con le risposte che avete fornito.