

# Indice modulo 2

```
public class Main {

    int x = 3; // dichiarazione
    x = 5; // assegnamento
    // Assegnamento == modifica, non inizializzazione
    // di una variabile

    public Main(){
        System.out.println("ciao");
    }

    public static int f(int n){
        return n + 1;
    } // tecnicamente è un metodo, non una funzione

    public int g(int n){
        return n + 1;
    }

    public static void main(String[] args) {
        int x = f(10);
        // garbage collector cancella poi l'oggetto
        int y = new Main().g(13);
        // senza cancellazione
        Main o = new Main();
        int y = o.g(13);
        // esplode
        // il nome di una variabile dice dove, non cosa
        Main o;
        int y = o.g(7);
    }
}
```

```
package it.unive.dais.po2.misc;

public class Main {
    class Zoo {
        public static class Animal {
            protected int weight;

            public Animal(int w) { this.weight = w; }

            public void eat(Animal a) {
```

```

        this.weight += a.weight;
    }
}

static class Dog extends Animal { // sottoclasse
    private boolean pedigree;

    public Dog(boolean pedigree, int w){
        this.pedigree = pedigree;
        super(w);
    }

    public void bark() {
        System.out.println("bau!");
    }
    @Override
    //Subsumption
    // viene chiamata questa funzione da pluto,
    // anche se è un animale, non un cane
    public void eat(Animal a){
        this.weight += a.weight * 2;
    }
}

public static void main(String[] args){
    Dog fido = new Dog(false, 30);
    Dog gigio = fido;
    Animal pluto = new Dog(true, 40); // polimorfismo
    gigio.bark();
    // non funziona
    // pluto è animal, anche se possiede il metodo
    pluto.bark();
    // Dynamic dispatching, chiama override
    // non hai perdita di informazioni a runtime
    pluto.eat(fido);
    gigio.eat(gigio);
}
}

```