

# Esercitazione 5

**Ereditarietà, overriding, overloading**

# Ripasso

- ▶ **OVERRIDING**: si scrive in una sottoclasse un metodo della superclasse con la stessa segnatura
- ▶ **OVERLOADING**: è possibile definire metodi con lo stesso nome ma con signature differenti

Cos'è la SEGNATURA? È la «firma» del metodo costituita dal nome del metodo, dal numero dei suoi parametri e dal loro tipo

Lo stesso nome di un metodo può essere usato per operazioni diverse, con definizioni diverse che richiedono un diverso tipo o numero di parametri.

# Esercizio 1

► Definire una classe Dipendente che ha i seguenti campi:

- Nome → String
- Cognome → String
- oreLavorativeMensili → Integer
- retribuzioneOraria → Integer

Scrivere un metodo che calcola lo stipendio del dipendente nel seguente modo:

$$\text{Stipendio} = \text{oreLavorativeMensili} * \text{retribuzioneOraria}$$

Si definisca anche un particolare tipo (sottoclasse) di dipendente responsabile di progetto (DipendenteResponsabile), al cui stipendio del dipendente (vedi sopra) viene aggiunto un bonus:

$$\text{Stipendio} = \text{oreLavorativeMensili} * \text{retribuzioneOraria} + \text{bonus}$$

Scrivere un *main()* che calcoli lo stipendio per una persona Dipendente e per una persona DipendenteResponsabile.

# Esercizio 2 (1)

► Data la classe Animale:

```
class Animale
{
    private String nome;
    public Animale(String s){
        nome = s;
    }
    public String comeTiChiami()
    {
        return nome;
    }
    public void parla(){}
    public void incontra(Animale a)
    {
        System.out.println(nome + "<:Ciao, " + a.nome + ">");
        parla();
    }
}
```

## Esercizio 2 (2)

- ▶ Si costruiscano due classi *Topo* e *Gatto* che estendono la classe *Animale*. In particolare le due classi devono ridefinire il metodo *parla()*:
  - ▶ Il topo deve ridefinire il metodo in modo che esso stampi sullo schermo «Squit»
  - ▶ Il gatto deve ridefinire il metodo in modo che esso stampi sullo schermo «Miao»
- ▶ Si aggiunga ora nella classe *Topo* una nuova versione del metodo *incontra()* che prende come argomento un oggetto di tipo *Gatto* e stampa:

```
System.out.println(nome + “:<Aiutoooooo!!!>”);
```

- ▶ Analogamente per il *Gatto* si aggiunga una nuova versione del metodo *incontra()* che prende come argomento un oggetto di tipo *Topo* e stampa:

```
System.out.println(nome + “:<Ti prendo!!!>”);
```

Scrivere un *main()* per verificarne il funzionamento

# Esercizio 3 (overloading)

- ▶ Definire una classe Studente che ha i seguenti campi:
  - Nome → String (Ereditato dalla classe Persona)
  - Cognome → String (Ereditato dalla classe Persona)
  - Matricola → Integer
  - ArrayList<Esami> → Esami (nome, voto)
  - Media voti → Integer
  - Tasse → Float

Scrivere un metodo che calcola la media dei voti degli studenti (memorizzarlo nel campo «media voti»)

Scrivere un metodo che, data una tassa massima, calcola la tassa finale applicando uno sconto nel seguente modo:

# Esercizio 3

Se lo studente indica la fascia di appartenenza viene applicato uno sconto in base alla fascia inserita. Successivamente viene applicato anche uno sconto in base alla media dei voti (sulla tassa al netto dello sconto applicato precedentemente)

Se lo studente non indica la fascia di appartenenza viene applicato uno sconto in base alla media dei voti

Fascia	Sconto
1	50%
2	35%
3	25%
4	10%
5	0%

Voto	Sconto
30	10%
29	8%
28	6%
27	4%
altro	0%

Scrivere una classe di prova

NB: Overloading dei metodi