

# Estensione delle classi

## 1 Estensione di una classe

Una classe B può dichiarare di estendere un'altra classe A nella sua intestazione:

```
class B extends A
```

Ciò significa che:

- la classe B è ottenuta estendendo lo stato e il comportamento di A
- B è *sottoclasse* di A
- A è *superclasse* di B

In generale, è possibile estendere una classe conoscendone anche solo il contratto: non è necessaria alcuna informazione relativa alla sua implementazione.

## 2 Ereditarietà dei membri

La sottoclasse eredita tutti i campi e i metodi della superclasse, ma ciò non implica che essi siano *accessibili* nel codice della sottoclasse: ciò dipende dalla loro visibilità.

I costruttori, invece, *non* vengono ereditati.

## 3 Metodi eseguibili da un oggetto

1. Metodi propri della classe dell'oggetto, cioè definiti per la prima volta in tale classe.
2. Metodi ridefiniti, che hanno la segnatura uguale a quella di un metodo della superclasse, ma ne ridefiniscono il comportamento.
3. Metodi ereditati, cioè definiti in una delle superclassi e non ridefiniti.

## 4 Costruttori e gerarchia

Per costruire un oggetto di una sottoclasse:

1. viene costruito un oggetto della superclasse
2. l'oggetto viene adattato alle esigenze della sottoclasse

Nel corpo di un costruttore, è possibile inserire come prima istruzione

`super(lista_argumenti)`

per invocare un costruttore della superclasse diretta.

- Generalmente, un costruttore si avvale dell'ausilio di
  - un altro costruttore della stessa classe (`this`)
  - un costruttore della superclasse (`super`)

L'unica eccezione è il costruttore della classe `Object`, che non ha superclassi.

- Se non si definiscono costruttori, ne viene automaticamente aggiunto uno senza argomenti e senza codice.
- Se un costruttore non ne invoca esplicitamente un altro, viene automaticamente invocato il costruttore privo di argomenti della superclasse.

Per questi motivi, se la superclasse non ha un costruttore senza argomenti

- è obbligatorio scrivere almeno un costruttore nella sottoclasse che invochi uno dei costruttori con argomenti della superclasse
- ogni costruttore della sottoclasse deve invocare esplicitamente un altro costruttore (`this` o `super`)

altrimenti si verifica un errore in fase di compilazione.

Anche le classi astratte, nonostante non possano essere istanziate, hanno dei costruttori, che sono utilizzati esclusivamente per costruire istanze delle sottoclassi.

## 5 La pseudo-variabile `super`

Oltre all'invocazione di costruttori della superclasse, la parola riservata `super` si può utilizzare come pseudo-variabile: essa consente di accedere a metodi e campi della superclasse.

Questa funzione di `super` è utile, ad esempio, quando si vuole invocare un metodo della superclasse di cui esiste anche una versione ridefinita nella sottoclasse.