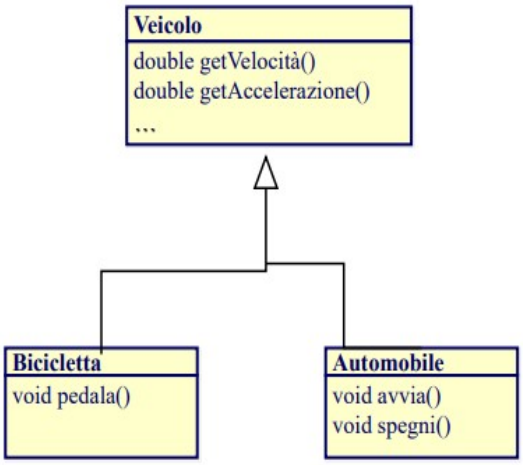
Ereditarietà

L'ereditarietà è un meccanismo per definire una nuova classe (classe derivata) come specializzazione di un'altra (classe base).
La classe base modella un concetto generico mentre la classe derivata modella un concetto più specifico.
La classe derivata dispone di tutte le funzionalità (attributi e metodi) di quella base.
In Java una classe derivata si definisce attraverso la parola chiave "extends" seguita dal nome della classe base.
Gli oggetti della classe derivata sono, a tutti gli effetti, estensioni della classe base, anche nella loro rappresentazione in memoria.
La classe derivata non può operare direttamente sui metodi della classe-base se quelli sono definiti privati.
La super-classe può definire attributi e metodi con visibilità "protected", rendendoli visibili alle sottoclassi.
Nel progetto TestContoBancario si utilizza l'ereditarietà per ereditare nella sotto-classe ContoEsteso i metodi e gli attributi dalla super-classe ContoBancario.



Veicolo.java

```
public class Veicolo {
    private double velocità;
    private double accelerazione;
    public double getVelocità() {…}
    public double getAccelerazione() {…}
}
```

Automobile.java

```
public class Automobile
  extends Veicolo {
    private boolean avviata;
    public void avvia() {…}
}
```

Polimorfismo

Con il polimorfismo si definiscono, nella super-classe, metodi con implementazione generica sostituiti, nelle sottoclassi, da implementazioni specifiche.
Per fare ciò è necessario inserire override (sovrascrivere) prima del metodo nella sotto-classe.
Si utilizzano variabili aventi come tipo quello della super-classe.

This

Il riferimento this in Java viene utilizzato per fare riferimento, all'interno di un metodo o di un costruttore, agli attributi o metodi locali. Questo tipo di riferimento non fa altro che puntare all'oggetto a cui appartiene risolvendo possibili problemi di ambiguità.