

Override e Extends

l'annotazione `@Override` viene utilizzata per indicare che un **metodo nella sottoclasse sta sovrascrivendo un metodo della classe padre**. Quando un metodo viene sovrascritto, la sottoclasse può fornire una propria implementazione specifica, che sostituisce quella della classe padre:

```
class Animale {
    void verso() {
        System.out.println("L'animale fa un verso");
    }
}

class Cane extends Animale {
    @Override
    void verso() {
        System.out.println("Il cane abbaia");
    }
}

public class Main {
    public static void main(String[] args) {
        Animale mioCane = new Cane();
        mioCane.verso(); // Stampa "Il cane abbaia"
    }
}
```

In questo esempio, il metodo `verso()` nella classe `Cane` sovrascrive il metodo `verso()` della classe `Animale`, fornendo un comportamento specifico per i cani.

Inoltre quando una classe `Cane` estende la classe `Animale` usando la parola chiave `extends`, **eredita tutti i metodi e i campi definiti nella classe `Animale`**. Questo significa che `Cane` avrà **accesso a tutti i metodi** di `Animale`, a meno che non vengano sovrascritti.