

Java

```

class TURTLEBREIT extends TURTLE
{
    @Override
    void Gehen(int länge)
    {
        super.Gehen(länge);
        this.Drehen(90);
        super.Gehen(1);
        this.Drehen(90);
        super.Gehen(länge);
        this.Drehen(90);
        super.Gehen(2);
        this.Drehen(90);
        super.Gehen(länge);
    }
}

```

Aufruf der Methode Gehen der Oberklasse

Zeichnet eine 3 Pixel breite Linie

So geht's

1. In der **Unterklasse** den identischen **Methodenkopf** der **zu überschreibenden Methode** implementieren (in Java @Override über dem Methodenkopf ergänzen)
2. **Neue Funktionalität im Methodenrumpf** der **überschreibenden Methode** in der **Unterklasse** implementieren.

Python

```

class TURTLEBREIT(TURTLE):

```

```

    def Gehen(self, länge):

```

Aufruf der Methode Gehen der Oberklasse

Zeichnet eine 3 Pixel breite Linie

```

        super().Gehen(länge)
        self.Drehen(90)
        super().Gehen(1)
        self.Drehen(90)
        super().Gehen(länge)
        self.Drehen(90)
        super().Gehen(2)
        self.Drehen(90)
        super().Gehen(länge)

```

Überschreiben von Methoden

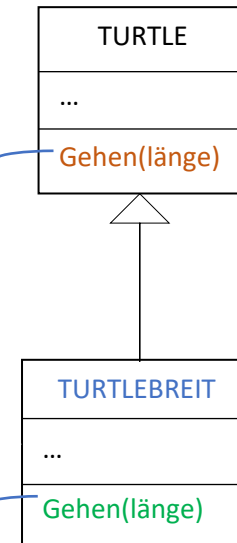
Von Oberklassen geerbte Methoden werden in der **Unterklasse neu festgelegt**. Zugriff auf Methoden der Oberklasse mit super weiter möglich.

- Vererbungsbeziehung vergessen umzusetzen
- Methodenkopf der **überschriebenen Methode** und der **überschreibenden Methode** stimmen nicht überein
- Fälschlicherweise wird nicht die **Methode der Oberklasse** (mithilfe von super./super().), sondern **die der Unterklasse** aufgerufen

Häufige Fehler

Überschriebene Methode

Überschreibende Methode



Klassendiagramm

