

## Nachdenkzettel Beziehungen/Vererbung

1. „Class B extends X“. Jetzt fügen Sie eine neue Methode in X ein. Müssen Sie B anpassen?  
Nein es sei denn man will die neu eingefügte Methode verwenden.

```
2. Class B extends X {  
    public void newMethodinB() { .... }  
}
```

Jetzt fügen Sie eine neue public Methode in ihre abgeleitete Klasse ein. Sie möchten diese neue Methode im Code verwenden. Prüfen Sie die folgenden Codezeilen:

```
X x = new B();  
x.newMethodinB();
```

Was stellen Sie fest?

Erste Zeile funktioniert da X eine spezialisierte Klasse von B ist. Zweite nemme da das Objekt X nicht die Methoden von B enthält.

```
2. Class B extends X {  
    @override  
    public void methodinB() { .... }  
}
```

Jetzt überschreiben Sie eine Methode der Basisklasse in ihrer abgeleitete Klasse. Sie möchten diese neue Methode im Code verwenden. Prüfen Sie die folgenden Codezeilen:

```
X x = new B();  
x.methodinB();
```

Was stellen Sie fest?

Die override Methode wird ausgeführt.

3. Versuchen Sie „Square“ von Rectangle abzuleiten (geben Sie an welche Methoden Sie in die Basisklasse tun und welche Sie in die abgeleitete Klasse tun)  
getFlaecheninhalt() sollte in die Rectangle klasse da diese auch in Square verwendet werden kann.

4. Jetzt machen Sie das Gleiche umgekehrt: Rectangle von Square ableiten und die Methoden verteilen.  
da Square nur den Flächeninhalt nur mit a berechnet müsste die Methode in Rectangle überschrieben werden.

5. Nehmen Sie an, „String“ wäre in Java nicht final. Die Klasse Filename „extends“ die Klasse String.  
Ist das korrekt? Wie heisst das Prinzip dahinter?  
Wäre nicht korrekt da die ganzen parameter und Funktionen nicht benötigt werden dies wäre eine Komposition