

## Nachdenkzettel Beziehungen/Vererbung

1. „Class B extends X“. Jetzt fügen Sie eine neue Methode in X ein. Müssen Sie B anpassen?

Nein, B muss nicht angepasst werden, da B von X erbt.

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?

Dieser Code führt zu einem Compile Error, da die Methode nur in B enthalten ist.

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?

Der Code wird ausgeführt (--> Polymorphie).

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)

Nur der Konstruktor muss in der Klasse Square angepasst werden.

4. Jetzt machen Sie das Gleiche umgekehrt: Rectangle von Square ableiten und die Methoden verteilen.

Da Square seine Methoden mit quadratischen Eigenschaften implementiert, müssten in der Rectangle-Klasse alle geerbten Methoden angepasst werden (inklusive Konstruktor).

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?

Das ist korrekt. Das Prinzip heisst Vererbung.