

Nachdenkzettel Beziehungen/Vererbung

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

eventuell, falls man es überschreiben will oder wenn die Methode in X abstract ist

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?

Wegen der zweiten Zeile wird sich der Code nicht ausführen lassen. Da zur compilezeit noch nicht bekannt ist, dass x ein B Objekt 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?

es führt die überschriebene Methode aus

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)

public double getArea(), ->override ableiten nicht ideal, da z.B bei getHöhe, getBreite Probleme entstehen
rectangle: a,b square: a=b

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

man müsste das meiste in Rectangle überschreiben oder arbeitet in Square mit a und b. Frage:
Muss man dann erben -> Rectangle = Square?

-> Basisklasse shape

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?

Beim erben soll sich die Unterklasse gleich verhalten wie die Oberklasse. Also erweitern.

"Filename" würde die Klasse String aber einschränken und könnte nicht das gleiche tun wie "String"