**Nachdenkzettel Beziehungen/Vererbung**

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

Nein, die Subklasse B kann direkt auf die Methoden der Basisklasse X zugreifen und muss nicht verändert werden.

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?

Die Methode kann nicht aufgerufen werden, weil das Objekt als Instanz der Basisklasse erstellt wurde, die neue Methode aber nur in der Subklasse implementiert wurde.

2. Class B extends X {

@override

public void methodinB() { …. }

}

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

X x = new B();

x.methodinB();

Was stellen Sie fest?

Dieses Mal funktioniert der Methodenaufruf problemlos, weil die neue Methode in der Basisklasse implementiert wurde.

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)

Rectangle:

Rectangle(int a, int b){}

getArea(int a, int b){}

getPerimeter(int a, int b){}

Square:

Problem beim Konstruktor, weil a=b sein muss und dass nicht mehr geändert werden kann (@override funktioniert nicht, da der Konstruktor Rectangle(int a) eine andere Signatur hat)

Alle anderen Funktionen funktionieren

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

Square:

Square(int a){}

getArea(int a){}

getPerimeter(int a){}

Rectangle:

Problem bei allen Methoden, weil alle Methoden nur einen Parameter erwarten, Rectangle aber zwei unterschiedliche Seitenlängen haben kann (@override funktioniert nicht, weil die Methoden mit zwei Parametern eine andere Signatur haben)

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?

Nein, diese Implementierung ist nicht korrekt, weil man dann in Filename versuchen würde Regeln für die Benennung der Files zu definieren und z.B. bei Verstößen eine Exception zu werfen und damit die Vererbung bricht. Damit ist die Vererbung von Anfang an falsch gewählt.

Besser ist stattdessen Filename als einfache Klasse zu deklarieren, die einen String als Attribut benutzt, aber nicht von String erbt (weil sich Filename eben nicht wie ein String verhält).