

Nachdenkzettel Beziehungen/Vererbung

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

No, you don't have to change B after implementing a new method in X.

```
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?

The method can't be called, because the instantiated object is of type X.

```
3. 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?

The child class's method is called even though the object is of type X.

4. 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 class Rectangle {  
  
    private int xSize;  
    private int ySize;  
  
    public Rectangle(int xSize, int ySize){  
        this.xSize = xSize;  
        this.ySize = ySize;  
    }  
  
    public int getXSize(){  
        return xSize;  
    }  
  
    public int getYSize(){  
        return ySize;  
    }  
  
    public int getArea(){  
        return ySize*xSize;  
    }  
  
    public int getCircumference(){  
        return 2*(xSize+ySize);  
    }  
  
    @Override  
    public String toString(){  
        return "This object is a rectangle with width of " + xSize + " and height of " + ySize;  
    }  
  
}
```

```
package org.example;  
  
public class Square extends Rectangle{  
  
    public Square (int size){  
        super(size, size);  
    }  
  
    @Override  
    public String toString(){  
        return "This object is of the type Square and its borders are exactly " + getXSize() + " long";  
    }  
  
}
```

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

```
class Square {
    int base;
    int sides = 4;
    int alpha = 90;

    public Square(int base) {
        this.base = base;
    }
}

class Rectangle extends Square {
    int height;

    public Rectangle(int base, int height) {
        super(base);
        this.height = height;
    }

    public int surfaceArea() {
        return base * height;
    }

    public int circumvent() {
        return 2 * base + 2 * height;
    }
}
```

6. Nehmen Sie an, „String“ wäre in Java nicht final. Die Klasse Filename „extends“ die Klasse String. Ist das korrekt? Wie heisst das Prinzip dahinter?

Yes, as it is an „is a“-association, you can implement it using inheritance.