

# Übung zur VL „Grundlagen der Programmierung“

## 11. Übung

Dr. Zubow

# Vererbung

- Vollziehen Sie nach, was in folgendem Java-Code passiert und erläutern Sie die Ausgabe.

```
class Base {
    protected String s = "base";
    protected String t = "base2";

    public String s() {
        return s;
    }

    public String t() {
        return t;
    }
}

class Derived extends Base {
    protected String s = "derived";
    protected String t = "derived2";

    public Derived() {
    }

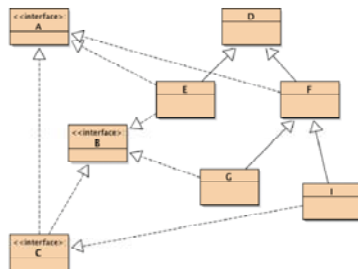
    public String s() {
        return s;
    }
}

public class PolyTest {
    static void out(Object o) {
        System.out.println(o);
    }

    public static void main(String[] args) {
        Derived d = new Derived();
        out(d.s);
        Base b = d;
        out(b.s);
        out(d.s());
        out(b.s());
        out(d.t());
        out(b.t());
    }
}
```

# Vererbung

- Gegeben sei das folgende Klassendiagramm.
- Hinweis:
  - Die Pfeile zeigen von einer Unterklasse auf eine Oberklasse, bzw. von einer Klasse auf ein Interface, dass diese Klasse implementiert.
  - Der Java-Code des Interface C muss also das folgende „Skelett“ besitzen: `interface C extends A, B { ... }`.



- Schreiben Sie den Java-Code der Klasse F („Skelett“).
- Schreiben Sie den Java-Code der Klasse E („Skelett“).
- Welche der folgenden Anweisungen würde ein Java-Compiler akzeptieren? Jede Anweisung sollte unabhängig von den anderen betrachtet werden.
  - `A r = new A();`
  - `A r = new F();`
  - `A r = (A)(new F());`
  - `A r = new G();`
  - `F r = new G();`
  - `G r = new F();`
  - `G r = (G)(new F());`
  - `B r = new I();`

# Code Analyse

- Der folgende Java-Code enthält **Fehler**, die Sie in finden, erklären und berichtigen sollen.
- Bsp.

```

1 //class for Complex numbers
2 class Complex {
3     private float x;
4     private float y;
5     //constructor for class Complex
6     public complex(float r, float i) {
7         //initialize instance variable x
8         float x = r;
9         //initialize instance variable y
10        float y = i;
11    }
12    //return real part of complex number
13    public int Re() {
14        return x;
15    }
16    //return imaginary part of complex number
17    public int Im() {
18        return y;
19    }
20 }

21 //subclass of Complex that includes methods for
22 //writing to real and imaginary parts of complex numbers.
23 class BComplex implements Complex {
24     //method for assigning to real part of complex number
25     public void setRe(float r) {
26         x = r;
27     }
28     //method for assigning to imaginary part of complex number
29     public void setIm(float i) {
30         y = i;
31     }
32     return;
33 }
34 }
35 }

36 //client class: Java program to test the classes Complex and BComplex
37 public class testComplex {
38     public void main(String arg) {
39         Complex c = new BComplex(1.0f, 2.0f);
40         c.setRe(3.0);
41         System.out.println(c.x);
42         System.out.println(c instanceof BComplex);
43     }
44 }
45 }
    
```