



# Universität Karlsruhe (TH)

## Lehrstuhl für Programmierparadigmen

Fortgeschr. Objektorientierung SS 2009 <http://pp.info.uni-karlsruhe.de/>  
Dozent: Prof. Dr.-Ing. G. Snelting [snelting@ipd.info.uni-karlsruhe.de](mailto:snelting@ipd.info.uni-karlsruhe.de)  
Übungsleiter: Andreas Lochbihler [lochbihl@ipd.info.uni-karlsruhe.de](mailto:lochbihl@ipd.info.uni-karlsruhe.de)  
Dennis Giffhorn [giffhorn@ipd.info.uni-karlsruhe.de](mailto:giffhorn@ipd.info.uni-karlsruhe.de)

Blatt 1

Ausgabe: 27.04.2009

Besprechung: 29.04.2009

### 1. Dynamische Bindung und Vererbung

Betrachten Sie folgendes Programm: Beschreiben Sie für die einzelnen Anweisungen in der `main`-Methode, welche Ausgaben entstehen und warum sie entstehen.

```
1 class TestSuperSub {
2     public static void main(String[] args) {
3         Super o = new Super(40, 30);
4         Super ou = new Sub(100, 200);
5         Sub u = new Sub(5000, 6000);
6
7         System.out.println(o.y);
8         System.out.println(ou.y);
9         System.out.println(((Super) u).y);
10        System.out.println(u.x);
11        System.out.println(o.getX());
12        System.out.println(ou.getX());
13        System.out.println(u.getX());
14
15        o.dispatch("o.dispatch");
16        ou.dispatch("ou.dispatch");
17        u.dispatch("u.dispatch");
18        u.dispatch2("u.dispatch2");
19    }
20 }
21
22 class Super {
23     private int x; public int y; Super s;
24
25     Super() {
26         System.out.println("Super()");
27         x = 1; y = 2; s = null;
28     }
29
30     Super(int x, int y) {
31         System.out.println("Super(int, int)");
32         this.x = x; this.y = y; s = new Super();
33     }
34
35     public int getX() {
36         System.out.println("Super.getX()");
```

```
37     return x;
38 }
39
40 private void print() {
41     System.out.println("Super.print");
42     System.out.println(x + "=" + getX());
43 }
44
45 public static void foo() {
46     System.out.println("Super.foo ");
47 }
48
49 public void dispatch(String m) {
50     System.out.println(m + "(1)");
51     this.print();
52     System.out.println(m + "(2)");
53     this.foo();
54     System.out.println(m + "(3)");
55     Super.foo();
56 }
57 }
58
59 class Sub extends Super {
60     public int x;    private int y;
61
62     public Sub(int x, int y) {
63         System.out.println("Sub(int, int)");
64         this.x = x; this.y = y;
65     }
66
67     public int getX() {
68         System.out.println("Sub.getX()");
69         return x;
70     }
71
72     public void print() {
73         System.out.println("Sub.print");
74     }
75
76     public static void foo() {
77         System.out.println("Sub.foo");
78     }
79
80     public void dispatch2(String m) {
81         System.out.println(m + "(1)");
82         this.print();
83         System.out.println(m + "(2)");
84         this.foo();
85         System.out.println(m + "(3)");
86         Super.foo();
87     }
88 }
```

---