

Diverse Problem

Från föreläsningar

Referenssemantik (1)

```
int[] a1 = {1, 2, 3};  
int[] a2 = {4, 5, 6};  
int[] a3 = doIt(a1, a2);
```

// Before
// Call
// After

```
int[] doIt(int[] x, int[] y) {  
    x[0] = y[1];  
    return x;  
}
```

Visualisera
semantiken genom
att rita bild med
variabler, referenser
och objekt. Före och
efter anrop!

Referenssemantik (2)

```
void program() {  
  H h1 = new H(1);  
  H h2 = new H(2); // Before  
  
  doIt(h1, h2); // Call  
  out.println(h1.i); // After  
}  
  
void doIt(H h1, H h2) {  
  h1.i = 4;  
  H tmp = h1;  
  h1 = h2;  
  h2 = tmp;  
}  
  
class H {  
  int i;  
  H(int i) { this.i = i; }  
}
```

Visualisera
semantiken genom
att rita bild med
variabler, referenser
och objekt. Före och
efter anrop!

Referenssemantik (3)

```
A[] as = new A[3];  
A a1 = new A();  
a1.i = 1;  
a1.s = "aaa";  
  
A a2 = new A();  
a2.i = 2;  
a2.s = "bbb";  
  
as[0] = a1;  
as[1] = a2; // Before  
as[2] = doIt(as); // Call  
// After
```

Visualisera
semantiken genom
att rita bild med
variabler, referenser
och objekt. Före och
efter anrop!

```
class A {  
  int i;  
  String s;  
}
```

```
A doIt( A[] as ){  
  as[0].s = as[1].s;  
  return as[0];  
}
```

Referenssemantik (4)

```
A a = new A();
a.arr = new int[]{1, 2, 3};
a.d = 1.5;
A b = new A();
b.arr = new int[]{4, 5, 6};
b.d = 2.5; // Before
```

```
a = doIt( b); // Call
               // After
```

```
class A {
    int[] arr;
    double d;
}
```

```
A doIt( A a ){
    A tmp = new A();
    tmp.arr = a.arr;
    tmp.d = a.d;
    return tmp;
}
```

Visualisera
semantiken genom
att rita bild med
variabler, referenser
och objekt. Före och
efter anrop!

Typer (1)

Vilka rader kompilerar? Fungerar det att köra (om vi kommenterar bort felaktiga delar)?

```
A a = new A(); // 1
B b = new B(); // 2
IX ix = new IX(); // 3
IY iy = null; // 4

a = b; // 5
b = (A) a; // 6

ix = iy; // 7
iy = ix; // 8

ix = a; // 9
a = ix; // 10
ix = b; // 11
iy = b; // 12

ix = (IX) iy; // 13
```

```
interface IX {
    void doIt();
}
```

```
interface IY {
    void doOther();
}
```

```
class A implements IX {
    public void doIt() {
    }
}
```

```
class B implements IY {
    public void doOther() {
    }
}
```

Typer (2)

Vilka rader kompilerar? Fungerar det att köra (om vi kommenterar bort felaktiga delar)? Om det fungerar vad skrivs ut?

```
A a = new A();  
B b = new B();
```

```
IX ix = a;    // 1  
IY iy = a;    // 2  
a.doIt();     // 3  
ix.doIt();    // 4  
iy.doIt();    // 5  
iy.doOther(); // 6
```

```
iy = b;        // 7  
iy = (IY) b;   // 8  
iy.doOther();  // 9
```

```
iy = new C();  // 10  
iy.doOther();  // 11
```

```
interface IX { void doIt();}  
interface IY { void doOther();}  
  
class A implements IX, IY {  
    public void doIt() {  
        out.println("A doIt()");  
    }  
    public void doOther() {  
        out.println("A doOther()");  
    }  
}  
  
class B {  
    public void doOther() {  
        out.println("B doOther()");  
    }  
}  
  
class C implements IY {  
    public void doOther() {  
        out.println("C doOther()");  
    }  
}
```