



# Universität Karlsruhe (TH)

## Lehrstuhl für Programmierparadigmen

Fortgeschr. Objektorientierung SS 2009

Dozent: Prof. Dr.-Ing. G. Snelting

Übungsleiter: Andreas Lochbihler

Dennis Giffhorn

<http://pp.info.uni-karlsruhe.de/>

[snelting@ipd.info.uni-karlsruhe.de](mailto:snelting@ipd.info.uni-karlsruhe.de)

[lochbihl@ipd.info.uni-karlsruhe.de](mailto:lochbihl@ipd.info.uni-karlsruhe.de)

[giffhorn@ipd.info.uni-karlsruhe.de](mailto:giffhorn@ipd.info.uni-karlsruhe.de)

Blatt 12

Ausgabe: 10.07.2009

Besprechung: 15.07.2009

### 1. Palsberg-Schwartzbach Typinferenz

- (a) Führen Sie für das folgende Programm einen Typcheck mit dem System von Palsberg-Schwartzbach durch.

---

```
1      class A {
2          int n(int i) { return i+1; }
3      }
4      class B extends A {
5          int nn(int j) { return j+2; }
6      }
7      A a=new A();
8      B b=new B();
9      int k=a.n(5);
10     k=b.n(8);
11     k=a.nn(42);
12     k=b.nn(k);
```

---

- (b) Inferieren Sie für das folgende Programm alle Typmengen mit dem System von Palsberg-Schwartzbach.

---

```
1      class A {
2          ? f() { return new C(); }
3      }
4      class B extends A {
5          ? f() { return new D(); }
6      }
7      class C {
8          ? g() { return new B(); }
9      }
10     class D extends C {
11         ? g() { return new A(); }
12     }
13     p=new B();
14     while(true) {
15         q=p.f();
16         p=q.g();
17     }
```

---

## 2. Generizität – Rekursive Typen

Betrachten Sie die Typen  $T_1 = \mu\tau.\{next : \tau, data : object\}$  und  $T_2 = \mu\sigma.\{next : \sigma, next2 : \sigma, data : object\}$ . Stehen  $T_1$  und  $T_2$  in einer Vererbungsbeziehung, und wenn ja, in welcher? Benutzen Sie die Konversionsregeln aus dem Skript zum Beweis.