



Halle, 13. Juni 2006

## Programmiersprachen (SS 2006)

### Übungsserie 11

#### Aufgabe 1 (Verdeckungsregeln)

Nachfolgend ist ein Quelltext in einer Sprache angegeben, die sowohl geschachtelte Blöcke als auch initialisierte Definitionen *def* und eine Ausgabe von Bezeichnerwerten *print* erlaubt.

Geben Sie an, welche Ausgaben das Programm erzeugt und zwar jeweils für die C- und die Algol-Verdeckungsregel.

```
1 {
2   def a=3
3   def b=5
4   def c=2
5   print c
6   {
7     print b
8     print a
9     def a=4
10    print c
11    {
12      def b=1
13      print b
14      print a
15      print c
16    }
17  def c=6
18  }
19  print c
20 }
```

## Aufgabe 2 (Variablen in Java-Programmen)

Betrachtet werde folgendes Java-Programm „Komische Fakultäet“:

```
class Fakultaet {
    private static int n=0;
    private int i;
    public Fakultaet(int i) {
        this.i=fak(int i);
        n++;
    }
    private int fak(int i) {
        if (i==1) return 1;
        else return i * fak(i-1);
    }
    public int getFak() {
        return i;
    }
    static public int getNum() {
        return n;
    }
}

public class KomischeFakultaet {
    static public void main(String[] args) {
        Fakultaet i = new Fakultaet(3);
        System.out.println(
            "Fakultaet von drei:" + i.getFak() + " \n"
+ "Fakultaet von vier: " + new Fakultaet(4).getFak() + "\n"
+ "Es wurden" + Fakultaet.getNum()
+ "Faklutaeten berechnet.");
    }
}
```

Im Programm treten nur Variablenbezeichner *n* und *i* auf. Sie haben jedoch unterschiedliche Definitions- und Benutzungsstellen sowie unterschiedliche Gültigkeitsbereiche und Lebensdauern.

- Notieren Sie an jeden Auftreten der Bezeichner *n* und *i*, ob es sich um eine Definitions- oder Benutzungsstelle sowie unterschiedliche Gültigkeitsbereiche und Lebensdauern handelt.
- Notieren Sie an jeder Definitionsstelle die Art der definierten Variable:
  - Klassenvariable
  - Objektvariable
  - Parametervariable
  - lokale Variable
- Notieren Sie an jeder Benutzungsstelle die zugehörige Definitionsstelle der Variablen, z.B. durch einen Kommentar der Form „//Def. in Zeile x“.

### Aufgabe 3 (Signaturen und Typinferenz)

- a. Welche Signaturen passen zu den Funktionen? Welche ist jeweils die allgemeinste der passenden?

```
fun function1(a,b)=(b,a);
```

- a) `float*float->float*float`
- b) `'a*'a->'a *'a`
- c) `'a*'b->'c* 'd`
- d) `int*string->string*int`
- e) `'a*'b->'b *'a`

```
fun function2(a,b)=if a then b else b;
```

- a) `bool*bool->bool*bool`
- b) `'a*'a->'a`
- c) `bool*bool->bool`
- d) `bool*'a->'a`

```
fun function3(a,b)=if a then b else function3(not a, "hello");
```

- a) `bool*string->bool`
- b) `bool*string->string`
- c) `bool*string->'a`
- d) `bool*'a->'a`

- b. Geben Sie die allgemeinste (evtl. polymorphe) Signatur der beiden folgenden Funktionen an.

```
fun function4(a,b)=if a=b then false else a;
```

```
fun function5(a,b,c,d,e)=if a<> "weiter" then a else function5(b,c,d,e,e);
```