

Übungsblatt 5 zu “Programmiersprachen”

Berthold Hoffmann, Studiengang Informatik (hof@tzi.de)

Besprechung am 29. 11. 2004

Aufgabe 1: Funktionen

In imperativen und objektorientierten Sprachen sind die Rümpfe von Funktionsvereinbarungen keine Ausdrücke, sondern Befehle, die im allgemeinen globale Variablen benutzen und sogar überschreiben können.

In Ada wurde versucht, trotzdem zu verhindern, dass sich eine Funktionsprozedur anders als eine mathematische Funktion verhält.

1. Zähle die Bedingungen auf, die an Ada-Funktionsprozeduren in dieser Hinsicht gestellt werden.
2. Reichen die Bedingungen aus, um Funktionsprozeduren referenziell transparent zu machen? (Eine Funktion heißt *referenziell transparent* wenn verschiedene Aufrufe mit den gleichen Argumentwerten stets das gleiche Ergebnis liefern.)
3. Reichen die Bedingungen aus, um zu verhindern, dass Funktionsprozeduren Seiteneffekte ausüben? (Eine Prozedur übt einen Seiteneffekt aus, wenn sie den Wert mindestens einer Variable verändert, die nicht lokal ist.)
4. Überlege ggf. zusätzliche Bedingungen, die Funktionsprozeduren referenziell transparent bzw. seiteneffektfrei machen könnten.
5. Erscheinen Dir diese Bedingungen praktikabel?

Aufgabe 2: Parameterübergabe

Betrachte folgende Prozedur in einer fiktiven Programmiersprache:

```
procedure multiply (in out m,n: Integer) is
begin
  m:= m * n;
  writeln (m, n);
end multiply;
```

Nimm an, die Variable *i* enthielte den Wert 1, und die Variable *j* enthielte den Wert 3. Was wird in den Prozeduraufrufen `multiply(i, j)`; und `multiply(i, i)`; ausgegeben, wenn die Parameter als *Wert-Ergebnisparameter* bzw. als *Variablenparameter* übergeben werden?