

Übungsblatt 8 zu “Programmiersprachen”

Berthold Hoffmann, Studiengang Informatik (hof@informatik.uni-bremen.de)
Selbstständige Bearbeitung am 10. Mai 2010, kurze Besprechung am 11. Mai 2010

Reichweiten in Ada

Welche Bindungen existieren an jedem der nummerierten Punkte in folgendem Ada-Programm:

```
package p;  
  x : constant Integer := 999;           -- (1)  
  subtype Nat is Integer range 0 .. x; -- (2)  
  var m, n: Nat;                         -- (3)  
function f (n: Nat) return Nat is  
  begin  
    ...                                   -- (4)  
  end f;  
  
  procedure w (i: Nat) is                -- (5)  
    n: constant Integer := 6;           -- (6)  
  begin  
    ...                                   -- (7)  
  end w;  
  
begin  
  ...                                     -- (8)  
end p.
```

Reichweiten in ML

Was ist die Umgebung an jedem nummerierten Punkt in folgendem ML-Programm?

```
let  
  val pi = 3.14159  
  and sin = fn (x: real) =>  
    ...                                   -- (1)  
  and cos = fn (x: real) =>  
    ... ;                                 -- (2)  
  val tan = fn (x: real) =>               -- (3)  
    sin (x) / cos (x);                   -- (4)  
  val rec f = fn (n: int) =>              -- (5)  
    ...                                   -- (6)  
in  
  ...                                     -- (7)  
end
```

(Beachten Sie, daß in einer Funktionsabstraktion “**fn** ($I : T$) $\Rightarrow E$ ” die Reichweite des formalen Parameters I der Funktionsrumpf E ist.)

Rekursive Definitionen in ML

Die rekursive Wertedefinition in ML “**val rec** $x = \dots x \dots$ ” ist nur erlaubt, wenn der definierte Wert von x eine Funktion ist.

Weshalb ist das wohl so?

(Hinweis: Betrachten Sie “**val rec** $x = x + 1$ ”.)

Abstraktion in Java

Untersuchen Sie Methodendefinitionen in Java.

1. Welche Größen können als Argumente an eine Methode übergeben werden?
Fehlen in Ihrer Aufstellung Größen, die man sinnvoll als Parameter übergeben könnte?
2. Definieren Sie – möglichst einfach – wie die aktuellen Parameter an die formalen Parameter einer Methode übergeben werden.
3. In welcher Reihenfolge werden die Argumente einer Methode übergeben?
4. In älteren Sprachen wie Algol-68 und Ada war die Reihenfolge, in der Argumente an eine Prozedur übergeben werden, *undefiniert*, d.h. dem Übersetzer überlassen.
Welche Vor- und Nachteile hat diese Art der Definition gegenüber der von Java?

Hinweis: Diese Fragen dienen der Vorbereitung des nächsten Kapitels über *Abstraktion*. Sie werden erst *am Ende* der nächsten Vorlesung am 11. Mai 2010 besprochen.