



Halle, 02. Mai 2006

Programmiersprachen (SS 2006)

Übungsserie 5

Aufgabe 1 (Variable)

Variablen werden Werte auf unterschiedliche Weise zugeordnet.

- In imperativen Sprachen durch ...
- In funktionalen Sprachen durch ...
- In logischen Sprachen durch ... (Suchen Sie vielleicht auch im Netz!)

Aufgabe 2 (ungetypter λ -Kalkül)

Eine Programmiersprache wird durch ihre Syntax und ihre Semantik definiert.

- Nennen Sie die Regeln, die die Syntax des ungetypten λ -Kalkül definieren.
- Nennen Sie die Regeln, die die Semantik des ungetypten λ -Kalkül definieren.

Aufgabe 3 (Spracheigenschaften)

Ordnen Sie folgende Aussagen aus dem Java Reference Manual den Spracheigenschaften *Lexik (Grundsymbole)*, *Syntax*, *statische und dynamische Semantik* zu.

1. A block is a sequence of statements and local variable declaration statements within braces.
2. A return statement with an expression must be contained in a method declaration that is declared to return a value or a compile-time error occurs.
3. If execution of the try block completes normally, then no further action is taken and the try statement completes normally.
4. The boolean type has two values, represented by the literals true and false, formed from ASCII letters.
5. A widening conversion of a signed integer value to an integral type T simply sign-extends the two's-complement representation of the integer value to fill the wider format.

6. It is a compile-time error for a line terminator to appear after the opening “and before the closing matching “.
7. The type of the operand expression of the unary-operator must be a primitive numeric type.
8. An array type is written as the name of an element type followed by some number of empty pairs of square brackets [].

Aufgabe 4 (Spracheigenschaften)

Herr Unbekannt hat gerade sein erstes Java-Programm geschrieben. Leider beherrscht er die Sprache noch nicht richtig und ist außerdem etwas schusselig. So haben sich Fehler eingeschlichen. Eigentlich sollte das Programm

```

1. class MyClass
2. {
3.   void output(int number)
4.   {
5.     java.io.PrintStream out;
6.     if(0<number && number<5) out=System.out; else out=null;
7.     out.println("Dies ist Ausgabe Nummer'+(number+1);
8.   }
9.
10. public static void main(String argv [])
11. {
12.   for(int i="0";i<5:i++)
13.     output(i);
14. }
15. }
```

folgende Ausgabe liefern:

```

Dies ist Ausgabe Nummer 1
Dies ist Ausgabe Nummer 2
Dies ist Ausgabe Nummer 3
Dies ist Ausgabe Nummer 4
Dies ist Ausgabe Nummer 5
```

Finden Sie alle 5 Fehler im Programm. Geben Sie genau an, an welcher Stelle der Übersetzer bzw. die Laufzeitumgebung den Fehler melden würde. Begründen Sie in einem kurzen Satz, zu welcher Kategorie der Fehler gehört.

Aufgabe 5 („normalisierte Gleitpunktzahl“)

Praktische Rechnungen werden auf Computern fast ausnahmslos in sog. Gleitpunktarithmetik über der Menge der Gleitpunktzahlen ausgeführt. Der klassische Standard für Gleitpunktzahlen definiert eine „normalisierte Gleitpunktzahl“.

Die Menge $\mathcal{R} = \mathcal{R}(B, t, E_1, E_2)$ der Gleitpunktzahlen zur Basis B mit t -stelliger Mantisse und dem Exponentenbereich $[E_1, E_2]$ besteht aus allen Zahlen x der Form

$$x = (-1)^s \times m \times B^e$$

mit dem Vorzeichen $(-1)^s$, $s \in \{0, 1\}$, der Mantisse

$$m = 0.m_1m_2\dots m_t = \sum_{i=1}^t m_i B^{-i}, \quad m_i \in 0, 1, \dots, B-1$$

und dem ganzzahligen Exponenten $e \in [E_1, E_2]$, $E_1 < 0 < E_2$. Dabei wird die Normalisierungsbedingung

$$m_1 \geq 1$$

oder $m_1 = m_2 = \dots = m_t = 0$ und $e = E_1 \iff x = 0$ gefordert.
Bestimmen Sie die Kardinalität der Menge $\mathcal{R} = \mathcal{R}(B, t, E_1, E_2)$.