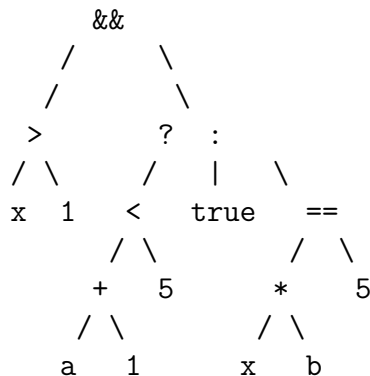


4. Übungsblatt - Informatik 1 - Lösungsbeispiele

Aufgabe 1 (Syntaxbaum)

Gegeben sei folgender Syntaxbaum:



Geben Sie den zugehörigen Java-Ausdruck an.

Lösungsvorschlag:

Der Ausdruck ist vollständig geklammert.

`(x > 1) && ((a + 1) < 5 ? true : (x * b == 5))`

Aufgabe 2 (Schrittweise Verfeinerung)

Entwickeln und beschreiben Sie ein Verfahren, um für ein gegebenes Datum im Gregorianischen Kalender (Tag, Monat, Jahr) und Uhrzeit (z.B. 12 Uhr 44 Minuten 33 Sekunden gegeben als Stunden, Minuten, Sekunden) die seit Jahresbeginn vergangenen Sekunden zu berechnen. Die Sommerzeit in Deutschland braucht nicht berücksichtigt werden.

Beschreiben Sie das zu lösende Problem zuerst etwas präziser mit dem Gegeben-Gesucht-Schema.

Geben Sie kein Java-Programm an, auch nicht in Teilen. Das Verfahren soll möglichst genau umgangssprachlich beschrieben werden analog der Aufgabe in den Rechnerübung zur Berechnung des Wochentags. Formelhafte Berechnungen können mathematisch angegeben werden.

Um ein Verfahren zu entwickeln, können Sie die Technik der Schrittweisen Verfeinerung verwenden.

Lösungsvorschlag:

Gegeben: Ein Datum mit tag, monat, jahr (z.B. 1.1.2013 für 1. Januar 2013) und die Zeit stunden, minuten, sekunden.

Gesucht: Die seit Jahresbeginn vergangenen Sekunden.

Die 1. Verfeinerung könnte die Sekunden aus dem tag, monat und dem jahr berechnen:

Die seit Jahresbeginn vergangenen Sekunden ist die Summe aus

1. sekunden, $60 * \text{minuten}$, $3600 * \text{stunden}$
2. und die vergangenen Sekunden aus dem tag, monat und jahr.

2. Verfeinerung (Einführen einer Variablen)

```
vergangeneSekunden = sekunden + 60 * minuten + 3600 * stunden
```

Addiere die vergangenen Sekunden aus tag, monat und jahr zu vergangeneSekunden.

3. Verfeinerung (der zweiten Zeile)

```
vergangeneSekunden = sekunden + 60 * minuten + 3600 * stunden
```

Es sei sekundenProTag die Anzahl Sekunden eines Tages ($60 * 60 * 24$)

addiere sekundenProTag * (tag - 1) auf die vergangenenSekunden

addieren die vergangenen Sekunden der Monate noch zu vergangeneSekunden.

4. Verfeinerung (mit größerem letzten Verfeinerungsschritt)

```
vergangeneSekunden = sekunden + 60 * minuten + 3600 * stunden
```

```
sekundenProTag = 60 * 60 * 24
```

```
vergangeneSekunden = vergangeneSekunden + sekundenProTag * (tag - 1)
```

Für jeden vorherigen Monat $m = 1, 2, \dots, \text{monat} - 1$ addieren wie folgt die vergangenen Sekunden pro Monat zu vergangeneSekunden:

a) bei $m = 1, 3, 5, 7, 8, 10$ addiere $60 * 60 * 24 * 31$

b) bei allen anderen ausser $m = 2$ und 12 addiere $60 * 60 * 24 * 30$

c) bei $m = 2$ addiere $60 * 60 * 24 * 28$ bzw $60 * 60 * 24 * 29$

falls jahr ein Schaltjahr ist

Aufgabe 3 (Stack, Laufzeitkeller)

Betrachten Sie folgendes Programm:

```
public class Stack {  
  
    public static double foo(double a, double b) {  
        double x = a;  
        b = x + 1.0;  
        return b * 2.0;  
    }  
  
    public static double bar(double x, double y) {  
        double b = y;  
  
        return x + b;  
    }  
}
```

```

    public static void main(String [] s) {
        double b = 1.0;
        double x = -1.0;

        x = foo(1.0, x);
        b = bar( -1.0, foo(1.0, x) );
    }
}

```

Skizzieren Sie den Zustand der Variablen im Laufzeitkellers(Stack) vor dem Ende jeden einzelnen Methodenaufrufs inklusive der main-Methode ohne das Programm auszuführen.

Lösungsvorschlag:

		+-----+	
		x = -1.0	
		y = 4.0	
		b = 4.0	
+-----+	+-----+	+-----+	
a = 1.0	a = 1.0	a = 1.0	
b = 2.0	b = 2.0	b = 4.0	
x = 1.0	x = 1.0	x = 1.0	
+-----+	+-----+	+-----+	+-----+
b = 1.0	b = 1.0	b = 1.0	b = 3.0
x = -1.0	x = 4.0	x = 4.0	x = 4.0
s = ..	s = ..	s = ..	s = ..
+-----+	+-----+	+-----+	+-----+