Grundlagen der Programmkonstruktion
Freiwillige Übung 2

Wenn Sie in einer freiwilligen Übungsgruppe angemeldet sind versuchen Sie die Aufgaben zu lösen, damit Sie im Rahmen der Übung diese Lösungen bzw. Ansätze diskutieren können. Der Besuch der Übung ist sinnlos, wenn Sie [EDIT: **nicht**] vorher versuchen, die Lösungen zu ermitteln.

1 Grammatik

1.1 Grammatik nachvollziehen

Gegeben ist die folgende Grammatik in EBNF-Notation:

```
Expr = Sum
Sum = [ '-' ] Product { '+' Product }
Product = Value { '*' Value }
Value = '0'..'9'
      | '(' Expr ')'
```

Einzelne Zeichen bzw. Schlüsselwörter sind in einzelne Hochkomma gestellt (auf den Folien sind diese **fett** gedruckt). Der Ausdruck `'0'..'9'` meint hier alle Ziffern von 0 bis 9.

Geben Sie an, welche der folgenden Ausdrücke in der Grammatik enthalten ist und welcher nicht (ignorieren Sie dabei "White-Spaces"). Begründen Sie Ihre Entscheidung.

- `0`
- `10`
- `(0)`
- `((0))`
- `-0`
- `--0`
- `-(-0)`
- `9 + 9`
- `2 + -3`
- `-2 + 3`
- `3 * 3`
- `-3 * 3`
- `3 * -3`
- `-2 * 2 + 3`
- `-2 * 2 + (-3)`
- `2 + 2 * 3 + 3`
- `(2 + 2) * (3 + 3)`

1.2 Grammatik erstellen

Ändern Sie die oben angeführte Grammatik ab, sodass sie auch folgende Anforderungen erfüllt:

- Die Grammatik soll auch Zahlen wie 999 (d.h. Zahlen mit mehr als einer Ziffer) akzeptieren.
- Die Grammatik soll auch Subtraktionen und Divisionen akzeptieren (d.h. Ausdrücke der Form `1-1` oder `2/3`).
- Die Grammatik soll nicht nur Zahlen akzeptieren sondern auch Variablen. Eine Variable besteht dabei aus einem kleinen Buchstaben, optional gefolgt von einer Zahl.

2 Java-Bytecode nachvollziehen

In der Praxisbesprechung wurde bereits gezeigt, wie Java-Bytecode aussieht. In dieser Aufgabe sollen Sie versuchen ein Programm in Java-Bytecode nachzuvollziehen.

Eine kurze (vereinfachende) Erklärung der wichtigsten Befehle:

Befehl	Erklärung
`iconst_`*`n`*	Integer-Wert n
`istore_`*`v`*	Speichere den letzten Integer-Wert in der Variable v.
`iload_`*`v`*	Benutze den Wert in der Variable v.
`ifle` *`Zeile`*	Wenn der aktuelle Wert kleiner gleich 0 ist (if less equal), springe zu Zeile *`Zeile`*.
`iadd`	Addiere die letzten zwei Werte.
`isub`	Berechne letzter Wert minus vorletzter Wert.
`iinc` *`v n`*	Erhöhe den Wert der Variable *`v`* um *`n`*
`goto` *`Zeile`*	Springe zu Zeile

Die Argumente der Befehle sind die zuletzt geladenen Werte, z.B. lädt `iconst_3` den Wert 3 und `iconst_5` den Wert 5. Ein darauf folgender Aufruf von `isub` berechnet daher $5 - 3$.

2.1 Aufgabe

Versuchen Sie nachzuvollziehen, wie die folgenden Befehle die Werte in den Variablen 1 und 2 beeinflussen, und notieren Sie die Zwischenergebnisse:

2.1.1 Beispiel

```
iconst_2
istore_1

iconst_5
istore_2
```

Variable 1	
Variable 2	

```
iload_1
iload_2
iadd
istore_1
```

Variable 1	
Variable 2	

```
iinc 2, -1
```

Variable 1	
Variable 2	

```
iload_1
iload_2
isub
istore_1
```

Variable 1	
Variable 2	

2.2 Programm

Das folgende Programm ist der Inhalt der Main-Methode eines Java-Programms:

```
0: iconst_1
1: istore_1
2: iconst_1
3: istore_2
4: iconst_3
5: istore_3

6: iload_3
7: ifle 24

10: iload_1
11: iload_2
12: iadd

13: istore_1

14: iload_1
15: iload_2
16: isub

17: istore_2

18: iinc 3, -1

21: goto 6

24: getstatic #2; //Field java/lang/System.out:Ljava/io/PrintStream;
27: iload_1
28: invokevirtual #3; //Method java/io/PrintStream.println:(I)V

31: return
```

Die Zeilen 24-28 rufen den Befehl `System.out.println()` auf und geben den Inhalt der Variable 1 aus (Zeile 27 lädt dazu dessen Wert).

Fragen

- Versuchen Sie, die Werte in den Variablen mit Hilfe der Beschreibung der Befehle oben nachzuvollziehen. Welcher Wert wird ausgegeben?
- Ersetzen Sie den Befehl in Zeile 4 durch `iconst_4` (bzw. `iconst_5`). Welcher Wert wird nun ausgegeben?
- Was berechnet dieses Programm?

3 Lambda-Kalkül

3.1 Beta-Reduktionen

- Reduzieren Sie den Term $(\lambda f.f\ 0)(\lambda y.y)$ mit der Beta-Reduktion, bis Sie eine Beta-Normalform erhalten.
- Hat der Term $(\lambda x.e(x\ x))\ (\lambda x.e(x\ x))$ eine Beta-Normalform? Begründen Sie ihre Entscheidung.

3.2 Reduktion 2

Beta-Reduktionen dürfen an beliebigen Stellen in einem Lambda-Term vorgenommen werden. Finden Sie einen Term, bei dem Sie an zwei verschiedenen Positionen gleichzeitig die Beta-Reduktion anwenden können.