

Übungsblatt 3: Software-Entwicklung 1 (WS 2016/17)

Ausgabe: 07.11.15

Abgabe: 14.11.15



Auf diesem Blatt werden Sie Ihre ersten Programmieraufgaben für SE1 lösen. Falls Sie bisher noch keine Erfahrung im Programmieren haben, ist es besonders wichtig, die Aufgaben selbst zu bearbeiten um die nötige Übung zu bekommen. Die folgenden Blätter bauen auf den Grundlagen auf, die auf diesem Blatt geübt werden. Beachten Sie auch, dass es auf Programme, die nicht kompilieren, höchstens 50% der Punkte gibt.

Wir bieten am Freitag (11.11.) von 15:30 bis 16:30 eine **zusätzliche Fragestunde** für Programmieranfänger im Terminalraum 32-411 an. Dort helfen wir Ihnen gerne, wenn Sie Probleme beim Lösen der Aufgaben oder beim Verwenden des Java-Compilers haben.

Aufgabe 1 Sprachen und Grammatiken (12 Punkte)

In dieser Aufgabe verwenden wir eine Mengennotation für Sprachen. Dabei steht a^n für das Wort in dem a n mal wiederholt wird. Wörter lassen sich auch hintereinander schreiben, so steht $a^n b^m$ für das Wort in dem zuerst n mal a und dann m mal b steht. Außerdem werden Klammern benutzt um ganze Gruppen zu wiederholen, zum Beispiel steht $(a^2 b)^3$ für das Wort $aabaabaab$. Mit diesen Abkürzungen für Wörter können wir leicht Mengen von Wörtern definieren. Zum Beispiel steht die Menge $\{a^n b^{2m} a^n \mid n \geq 0 \text{ und } m \geq 1\}$ für die Sprache $\{bb, abba, aabbaa, \dots, bbbb, abbbba, \dots\}$. Für alle Wörter der Sprache gilt also, dass vorne genau so viele a stehen wie hinten, und dass in der Mitte eine gerade Anzahl von b steht. Außerdem muss es mindestens zwei b in der Mitte geben, weil $m \geq 1$ gilt.

a) Geben Sie jeweils eine Grammatik für die folgenden Sprachen an:

$$\begin{aligned} L_0 &= \{a^n b^m \mid n \geq 1 \text{ und } m \geq 2\} \\ L_1 &= \{a^n b^m \mid n \geq 1 \text{ und } m \geq n + 2\} \\ L_2 &= \{(ab)^n \mid n \geq 0\} \\ L_3 &= \{a^n b^{2m} a^{3n} \mid n \geq 0 \text{ und } m \geq 1\} \end{aligned}$$

b) Geben Sie die Sprachen für folgende Grammatiken in Mengen-Schreibweise an:

$$\begin{aligned} \Gamma_3 &= (N, T, \Pi, S) \quad \text{wobei: } N = \{S\}, T = \{a\}, \Pi = \{S \rightarrow aS \mid a\} \\ \Gamma_4 &= (N, T, \Pi, S) \quad \text{wobei: } N = \{S\}, T = \{a, b\}, \Pi = \{S \rightarrow aSa \mid b\} \\ \Gamma_5 &= (N, T, \Pi, S) \quad \text{wobei: } N = \{S, X\}, T = \{a, b, c\}, \Pi = \left\{ \begin{array}{l} S \rightarrow aXa \\ X \rightarrow bSb \mid c \end{array} \right\} \end{aligned}$$

c) Geben Sie zur folgenden Beschreibung eine Grammatik und eine Definition in Mengennotation an:

Jedes Wort der Sprache ist eine mindestens einelementige Folge von a gefolgt von einer möglicherweise leeren Folge von x , die zur linken und rechten durch ein einzelnes l bzw. r begrenzt ist.

Beispiel: Die Worte $alr, aalxr, alxxxxr$ sind in der Sprache und die Worte $lar, arl, xrl, lxxr, allrr$ sind **nicht** in der Sprache enthalten.

Aufgabe 2 Typen (Sinnvoll bearbeiten)

Geben Sie jeweils einen möglichen Typ für die Variablen `x1` bis `x10` an, so dass gültige Variablendeklarationen entstehen. Werten Sie außerdem die jeweiligen Ausdrücke aus.

```
x1 = 40 + 2;           x6 = 1 + 2 == 3;
x2 = 3.1415;           x7 = 1 + "2";
x3 = "hello" + " " + "world";
x4 = 100 + 1.;          x8 = 1 / 3;
x5 = 1 + 2 < 3;         x9 = 3 < 4 && 6 < 5 || 7 < 8;
                        x10 = Integer.parseInt("4" + 4);
```

Aufgabe 3 Berechnungen mit Programm-Parametern (8 Punkte)

Hinweis: Für diese und andere Aufgabe benötigen Sie einige mathematische Funktionen. Informieren Sie sich deshalb über die Math Bibliotheksfunktionen von Java. Die Dokumentation finden Sie unter <https://docs.oracle.com/javase/8/docs/api/java/lang/Math.html>.

- a) Schreiben Sie ein Java-Programm “IsStronglyMonotonicIncreasing3”, welches drei `double`-Werte x, y , und z nimmt und “true” ausgibt, wenn die drei Zahlen von links nach rechts immer größer werden (also $x < y < z$ gilt).

Beispiel:

```
> java IsStronglyMonotonicIncreasing3 1 2 3
true
> java IsStronglyMonotonicIncreasing3 4.1 4.2 4.2
false
```

*Zur Erinnerung: Die Funktion `Double.parseDouble(String s)` wandelt einen String in einen **double** Wert um.*

- b) Schreiben Sie ein Programm “Windchill”, welches eine Temperatur T (in Grad Celsius) und eine Windgeschwindigkeit v (in Kilometern pro Stunde) nimmt und die sogenannte “Windchill-Temperatur” (WCT) berechnet und ausgibt. Dieser Wert lässt sich mit der folgenden Formel berechnen:

$$WCT = 13,12 + 0,6215 \cdot T - 11,37 \cdot v^{0,16} + 0,3965 \cdot T \cdot v^{0,16}$$

Hinweis: Verwenden Sie die Funktion `Math.pow`.

Beispiel:

```
> java Windchill 5 15
1.7489672954177302
```

Aufgabe 4 Logarithmus berechnen (5 Punkte)

Schreiben Sie ein Java-Programm, welches zwei `int` Zahlen x und b als Programm-Parameter nimmt und die größte `int` Zahl n berechnet, so dass $b^n \leq x$ gilt. Geben Sie n auf der Konsole aus. Sie können davon ausgehen, dass $b > 1$ und $x \geq 1$ gilt. Verwenden Sie für diese Aufgabe keine mathematischen Funktionen aus der Java Standard-Bibliothek.

Aufgabe 5 Programm-Parameter und Terminierung (Sinnvoll bearbeiten)

Für diese Aufgabe ist nur Teil e) abzugeben. Die anderen Teilaufgaben sollten Sie in der Abnahme erklären können.

Gegeben ist das folgende Java-Programm:

```
public class Range {
    public static void main(String[] args) {
        int start = Integer.parseInt(args[0]);
        int end   = Integer.parseInt(args[1]);
        int step  = Integer.parseInt(args[2]);

        int i = start;
        while (i != end) {
            System.out.println(i);
            i = i + step;
        }
    }
}
```

- a) Speichern Sie das obige Programm in einer Datei namens “Range.java” und übersetzen Sie es mit dem Java-Übersetzer.
- b) Führen Sie das Programm mit dem Befehl `java Range 0 100 5` aus. Was ist die Ausgabe?
- c) Was passiert, wenn Sie andere Argumente an das Programm übergeben? Zum Beispiel nur eine Zahl, vier Zahlen, oder Buchstaben? Wenn Fehler auftreten, wo steht die Zeilen-Nummer, in der der Fehler auftrat?
- d) Für welche Eingaben terminiert das Programm, wenn man annimmt, dass es keine Überläufe gibt? Begründen Sie jeweils, warum das Programm terminiert. Wie sieht es mit Überläufen aus?

Hinweis: Was ist die Ausgabe von `java Range 0 25 4`?

Wenn Sie ein Programm von der Shell aus starten, dann können Sie es mit der Tastenkombination `Strg+C` abbrechen.

- e) Verbessern Sie das Programm so, dass es immer terminiert und es für alle bereits terminierenden Eingaben im Verhalten unverändert bleibt.

Aufgabe 6 Simulation von Mensa-Schlangen (Sinnvoll bearbeitet)

In dieser Aufgabe sollen Sie die Länge von Mensa-Schlangen simulieren, indem Sie das Ankommen von Studenten durch Zufallszahlen simulieren.

Wir machen dazu die folgenden vereinfachten Annahmen über die Mensa und das Ankommen von Studenten:

- Pro Minute werden 5 Essen ausgegeben. Diese Geschwindigkeit ist immer gleich.
 - Jede Minute kommen 0 bis 10 Studenten an und stellen sich in die Schlange. Die Anzahl ist dabei gleich verteilt und unabhängig von Einflüssen wie der Zeit oder der Schlängellänge.
 - Die Studenten kommen immer zur vollen Minute, nicht kontinuierlich dazwischen.
 - Am Anfang steht niemand in der Schlange.
- a) Schreiben Sie ein Programm, welches die Länge der Mensa-Schlange Minute für Minute simuliert und für jede Minute die Länge der Schlange auf der Konsole ausgibt. Es sollen insgesamt 120 Minuten simuliert werden.
- b) Verwenden Sie Programm-Parameter, um die Zahl der Studenten pro Minute (Minimum und Maximum) und die Geschwindigkeit der Essensausgabe festlegen zu können.
- c) (*Freiwillige Zusatzaufgabe*) Wenn Sie jede Zahl in einer eigenen Zeile ohne zusätzlichen Text ausgeben, können Sie die ausgegeben Daten mit dem “Pipe”-Operator (“|”) an andere Programme weiterleiten. Für diese Aufgabe kann zum Beispiel das Programm `gnuplot`¹ verwendet werden, um aus den Daten einen Graphen zu zeichnen. Führen Sie dazu den folgenden Befehl aus, wobei Sie “`java MensaSchlange 5 0 10`” durch den Aufruf Ihres Programms ersetzen:

```
java MensaSchlange 5 0 10 | gnuplot -p -e "plot '<cat' with lines"
```

¹<http://gnuplot.sourceforge.net/> gnuplot ist auf den Rechnern des SCI bereits für Sie installiert.