

Grundlagen der Programmierung

(Vorlesung von Prof. Bothe)

WS 15/16

Übungsblatt 3: Algorithmen in Java & Grammatiken

Abgabe: bis **9:00 Uhr** am **30.11.2015** über **Goya**

Die Lösung dieses Übungsblatts soll in Gruppen von je 2 Personen erfolgen. Die Abgabe der Lösungen erfolgt durch Hochladen **einer angegebenen Datei für jede Aufgabe** im Goya-System.

Hinweis: Die Lösungen zu den Aufgaben 1 und 2 sollen als **Java-Quellcode** erfolgen, aus diesen Quellen soll auf dem Rechner `gruenau5.informatik.hu-berlin.de` durch den Aufruf von `javac` möglich sein, das jeweilige Java-Programm zu erzeugen. Das Java-Programm wird dann wie in der jeweiligen Aufgabe angegeben ausgeführt. Bitte testen Sie ihre Lösung.

Aufgabe 1 (5 Punkte) → `Square.java`

Schreiben Sie ein Java-Programm mit dem Namen `Square`, das ein nichtausgefülltes Quadrat mit dessen Diagonalen ausgibt. Die Begrenzung des Quadrats soll aus dem Plus-Zeichen (+) bestehen, die Diagonalen aus dem Stern-Zeichen (*). Die Seitenlänge (Anzahl der Plus-Zeichen auf einer Seite des Quadrats) soll dabei per Konsolenargument einstellbar sein. Verwenden Sie das Rumpfprogramm am Ende der Aufgabestellung und beachten Sie die Hinweise zu Konsolenargumenten.

Nachfolgend ein paar Beispielaufrufe mit den entsprechenden Quadraten:

```
java Square 7
```

```
+++++++
+*      *+
+ * * +
+  *  +
+ * * +
+*      *+
+++++++
```

```
java Square 4
```

```
++++
+***+
+***+
++++
```

```
java Square 2
```

```
++
++
```

```
java Square 0
```

Ihr Programm könnte folgendermaßen beginnen:

```
public class Square {
    /**
     *
     * @param args die Variable args enthält alle (durch Leerzeichen
     * getrennte) Konsolenargumente, die beim Aufruf des Programms in der
     * Konsole übergeben wurden.
     */
    public static void main(String[] args) {
        // mit dem Index 0 wird auf das erste Konsolenargument zugegriffen
        String arg1 = args[0]; // arg1 ist eine Zeichenkette vom Typ String
        // mit Hilfe der Funktion Integer.parseInt wird die Zeichenkette
        // in eine Ganzzahl von Typ int umgewandelt.
        int length = Integer.parseInt(arg1);
    }
}
```

Aufgabe 2 (8 Punkte) → IDCheck.java

Ein Verfahren zur Erkennung von Eingabefehlern ist die Verwendung von Prüfziffern. Diese werden aus einer Eingabezahl berechnet und dann zu dieser hinzugefügt. Ein Beispiel für eine Prüfziffer ist die letzte Stelle der 10-stelligen Nummer eines deutschen Personalausweises.

Neuer Personalausweis DE



Alter Personalausweis DE



Ausweisnummer

Die Prüfziffer (10. Stelle) wird aus den ersten 9 Stellen der Ausweisnummer nach folgendem Schema berechnet:

- von links nach rechts werden die einzelnen Ziffern sich wiederholend mit 7, 3, 1 gewichtet, wobei Buchstaben die folgenden Werte haben: A=10, B=11, C=12, ...,
- die jeweiligen Produkte aus dem Wert der jeweiligen Ziffer und ihrem Gewicht werden errechnet,
- die Endziffer der Summe der einzelnen Produkte ist die Prüfziffer.

Beispiel für **T220001293**:

Zeichen	Wert	Gewicht	Produkt
T	29	7	203
2	2	3	6
2	2	1	2
0	0	7	0
0	0	3	0
0	0	1	0
1	1	7	7
2	2	3	6
9	9	1	9
			233

Schreiben Sie ein Java-Programm, das anhand einer 10-stelligen Ausweisnummer, die mit Aufruf des Programms übergeben wird, entscheidet, ob die Ausweisnummer gültig ist (d.h., ob die Prüfziffer stimmt). Sie können mit dem folgenden Java-Programm beginnen:

```
public class IDCheck {
    public static void main(String[] args) {
        String id = args[0]; // 1. Argument in der Konsole
    }
}
```

Damit lässt sich das Java-Programm dann z.B. folgendermaßen aufrufen:

```
java IDCheck T220001293
```

Bei diesem Aufruf soll dann folgende Ausgabe erscheinen (und nichts anderes):

```
Ausweisnummer ist gültig
```

Bei folgendem Aufruf:

```
java IDCheck 1220001295
```

Soll dann folgende Ausgabe erscheinen (und nichts anderes):

```
Ausweisnummer ist nicht gültig
```

Hinweise:

Um auf ein bestimmtes Zeichen in einer Zeichenkette zuzugreifen, können Sie folgende Methode aus der Klasse String verwenden:

```
public char charAt(int index)
```

Beispiele:

```
String str = "Hallo Welt";  
// Index 0 greift auf das erste Zeichen zu  
char c = str.charAt(0); // c enthält den Buchstaben 'H'  
// Index 6 greift auf das 7. Zeichen zu  
c = str.charAt(6); // c enthält den Buchstaben 'W'
```

Um den Wert eines Zeichens zu erhalten, können Sie folgende Funktion aus der Klasse Character nutzen:

```
public static int getNumericValue(char c)
```

Beispiele:

```
char c = 'A';  
int value = Character.getNumericValue(c); // value hat den Wert 10  
c = 'T';  
value = Character.getNumericValue(c); // value hat den Wert 29
```

Aufgabe 3 (4 Punkte) → blatt3_aufgabe3.txt

Bitte beachten Sie für diese Aufgabe die Vorrangregeln (Vorlesung II.8) und die (Links-) Assoziativität der meisten Operatoren.

- 1) $a = x == y != z \mid p \geq q/3$
- 2) $b = 4*x \wedge 3-y \gg 5$
- 3) $c \mid= x \mid y \& y \wedge z \mid z \& x$
- 4) $d \ ? \ ++i \ \ll \ 5\%3 \ : \ \sim 1 \ \mid \ -2$

- a) Ergänzen Sie in den obigen Java-Ausdrücken genau alle sinnerhaltenden Klammern ohne Mehrfachklammerung eines Teilausdrucks, indem Sie jeden Teilausdruck explizit klammern. Bei einer Mehrfachklammerung ist ein bereits geklammerter Ausdruck wieder direkt geklammerter, z. B. ist $(1+2)$ einfach geklammerter und $((1+2))$ oder $((((1+2)))$ sind mehrfachgeklammert. Bei Mehrfachklammerung könnte der Ausdruck beliebig lang werden. Zur Vereinfachung sollen außerdem die den gesamten Ausdruck umschließenden Klammern sowie die Klammern um einzelne Zahlen und Variablen weggelassen werden.

Beispiel: Der sinnerhaltend geklammerte Ausdruck zum Ausdruck $1+2*3+4$ ist $(1 + (2 * 3)) + 4$.

- b) Die auftretenden Variablen sind vom Typ `int` oder vom Typ `boolean`, die sich jeweils aus dem Kontext der verwendeten Operatoren ergeben. Welchen Typ hat der jeweilige Ausdruck aus obigen Beispielen 1) ... 4)?
(Möglich sind: `int` | `boolean` | `int oder boolean`)

Beispielformat der Abgabedatei:

1a) $a=(1+(2*3))+4$
1b) `int`

2a) $b=q$
2b) `int oder boolean`

3a) $c=(1>1)\mid\mid(2>2)$
3b) `boolean`

4a) $d=77$
4b) `int`

Aufgabe 4 (3 Punkte) → blatt3_aufgabe4.txt

Gegeben sind Grammatiken G_1 , G_2 und G_3 . Entscheiden Sie ob die jeweils angegebenen Wörter durch die entsprechende Grammatik erzeugt werden oder nicht. Bitte beachten Sie: Alle Antworten in einer Teilaufgabe müssen korrekt sein, damit sie einen Punkt bekommen.

1. $G_1 = \{ \{a, b\}, \{S, N\}, S, R_1 \}$
 $R_1 = \{ (S, aNb), (S, aNbaNb), (N, aNb), (N, \epsilon) \}$
 - 1.1. ab
 - 1.2. abb
 - 1.3. aababb
 - 1.4. abaaabbb

2. $G_2 = \{ \{c, d, e, f\}, \{S, M, N\}, S, R_2 \}$
 $R_2 = \{ (S, cMd), (M, c), (M, cMd), (M, eNdc), (M, edc), (N, eNf), (N, \epsilon) \}$
 - 2.1. cd
 - 2.2. ef
 - 2.3. ccd
 - 2.4. cefd

3. $G_3 = \{ \{g, h\}, \{S, M, N\}, S, R_3 \}$
 $R_3 = \{ (S, gMgg), (M, gMg), (M, \epsilon), (M, hN), (N, hN), (N, \epsilon) \}$
 - 3.1. ghgg
 - 3.2. ghggg
 - 3.3. ghhgg
 - 3.4. ggghhhhhgggg

Beispielformat der Abgabedatei:

- 1.1. ja
- 1.2. nein
- 1.3. ja
- 1.4. nein
- 2.1. ja
- 2.2. nein
- 2.3. ja
- 2.4. nein
- 3.1. ja
- 3.2. nein
- 3.3. ja
- 3.4. nein

Auch in diesem Jahr steht Ihnen wieder der FIT Java Tutor – ein intelligentes Lernsystem für die Java Programmierung – zur Verfügung:
<https://its.fit.informatik.hu-berlin.de/>