

Aufgabenblatt 1

Teil 0: Organisatorisches

Registrieren Sie sich auf der Moodle-Seite der Vorlesung

<https://moodle.uni-kassel.de/moodle/course/view.php?id=8875>

sowie beim Abgabesystem (Link siehe Moodle-Seite). Sollten Probleme auftreten, klären Sie diese bitte in den Übungen im CIP-Pool. Wenn dies nicht möglich ist, schicken Sie bitte unverzüglich eine Problembeschreibung an support-plm@uni-kassel.de. Aus organisatorischen Gründen kann sich die Beantwortung der Emails verzögern. Falls Sie keine rechtzeitige Antwort bekommen (aber nur dann!), dürfen Sie die Lösungen der Blätter 1 (und ggf. auch 2) ausnahmsweise per Email abgeben (siehe dazu Teil 3 des Aufgabenblattes).

Denken Sie außerdem daran, sich das Lehrbuch zu besorgen, wie in der Vorlesung erklärt.

Teil 1: Schaffen einer Programmierumgebung und Test der Abgabe

Schritt 1: Installation einer Programmierumgebung. Falls Sie keinen eigenen Rechner besitzen, wenden Sie sich bitte an das ITS. Es verfügt über Rechnerpools und verleiht auch Notebooks.

Während des Semesters benötigen Sie zum Editieren, Compilieren und Ausführen Ihrer Java-Programme eine Programmierumgebung.

Sie müssen zunächst das Java SE Development Kit (JDK) sowie einen Editor zur Eingabe der Programmtexte installieren. Beides ist in der Installationsanleitung beschrieben, welche Sie auf der Moodle-Seite der Vorlesung finden.

Schritt 2: Test der Programmierumgebung. Machen Sie sich zunächst mit dem gewählten Editor vertraut. Tippen Sie dann das folgende Java-Programm im Editor ein (die Bedeutung wird später erklärt):

```
class Hallo {  
    public static void main(String[] args) {  
        System.out.println("Gruesse vom ersten Java-Programm.");  
    }  
}
```

Speichern Sie das Programm unter `Hallo.java` ab und übersetzen Sie es. Sofern es nicht bereits durch den Editor unterstützt wird, erfolgt das Übersetzen (oder: Kompilieren) von Programmen durch Eingabe von

`javac Programmname.java`

im Kommandofenster. Achten Sie darauf, dass das Programm unter exakt diesem Namen (ohne Endung `.txt`) gespeichert wurde und sich im aktuellen Verzeichnis befindet. In Windows entspricht das Kommandofenster der Windows- Eingabeaufforderung, die Sie mit `cmd` erreichen (siehe Installationsanleitung).

Falls beim Übersetzen Fehlermeldungen erscheinen, sollten Sie überprüfen, ob Sie sich verschrieben haben (zum Beispiel Semikolon, Klammern oder Groß-/Kleinschreibung).

Führen Sie dann das Programm aus. Tippen Sie dazu im Kommandofenster

`java Programmname`

(ohne Endung) ein. Wenn die Ausschrift erscheint, haben Sie die Aufgabe erfolgreich gelöst.

Schritt 3: Abgabe der Lösung. Erstellen Sie ein jar-Archiv mit Ihrer Lösung, wie in Kapitel 5 der Installationsanleitung erklärt. Das Archiv zu Aufgabenblatt 1 sollte zwei Dateien enthalten: `Hallo.java` sowie Ihre Lösung zu Teil 2 dieses Aufgabenblattes. Wenn Sie Teil 2 noch nicht bearbeitet haben, packen Sie zunächst nur `Hallo.java` in das jar-Archiv und übertragen es ins Abgabesystem. Rechtzeitig vor dem Abgabetermin laden Sie dann Ihre vollständige Lösung hoch. Diese ersetzt automatisch die Testversion.

Bitte klären Sie eventuelle Probleme schnellstmöglich (siehe Abschnitt 0).

Teil 2: Grundlagen der Programmierung

Arbeiten Sie Kapitel 1 des Lehrbuches¹ mit Ausnahme von Abschnitt 1.6 durch und prägen Sie sich die Bedeutungen der Grundbegriffe ein: Daten, Befehl, Byte, Algorithmus, Programm, Anweisung, Compiler, Assertion. Bei Ablaufdiagrammen genügt es, sie lesen zu können, die Symbole müssen Sie sich nicht merken. Lösen Sie dann schriftlich die folgenden Aufgaben.

- a) Gibt es bei einer typischen Programmausführung mehr Variablen, mehr Werte oder gleich viele Variablen wie Werte? Begründen Sie.

(2 Punkte)

- b) Welche der folgenden Wertzuweisungen bzw. Folgen sind lt. Lehrbuchkapitel zulässig? Geben Sie für die unzulässige(n) Wertzuweisung(en) an, warum sie nicht zulässig ist/sind.

1. `null ← 1`

3. `1 ← 1`

2. `null + eins ← 1`

4. `i ← 'A'`
`i ← 7`

(4 Punkte)

¹H. Mössenböck: Sprechen Sie Java? dpunkt-Verlag, 2014, Link zu Kapitel 1 auf Moodle-Seite

c) Wenn man bei der Sequenz

$$x \leftarrow 3$$
$$y \leftarrow 5$$

die beiden Anweisungen vertauscht zu

$$y \leftarrow 5$$
$$x \leftarrow 3$$

bleibt die Funktionalität die gleiche. D.h., in beiden Fällen enthält am Ende die Variable x eine 3 und die Variable y eine 5. Ist das bei jeder Sequenz so? Wenn ja, begründen Sie. Wenn nein, geben Sie ein konkretes Gegenbeispiel an.

(3 Punkte)

d) Lässt sich das Ablaufdiagramm aus Lehrbuchabschnitt 1.4.3 so umschreiben, dass **keine** Verzweigung verwendet wird? Alle übrigen Anweisungsarten sind erlaubt. Die Funktionalität muss die gleiche bleiben. Wenn ja, geben Sie ein alternatives Ablaufdiagramm an. Wenn nein, begründen Sie.

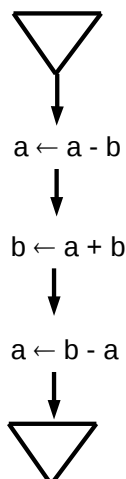
(6 Punkte)

e) Simulieren Sie in einem Schreibtischtest die Funktionsweise des Algorithmus zur Quadratwurzelberechnung für $x = 24$. Ändern Sie dabei die Schleifenbedingung ab zu $|a - \text{root}| > 0.01$. Geben Sie die Tabelle des Schreibtischtests sowie das Ergebnis an.

(5 Punkte)

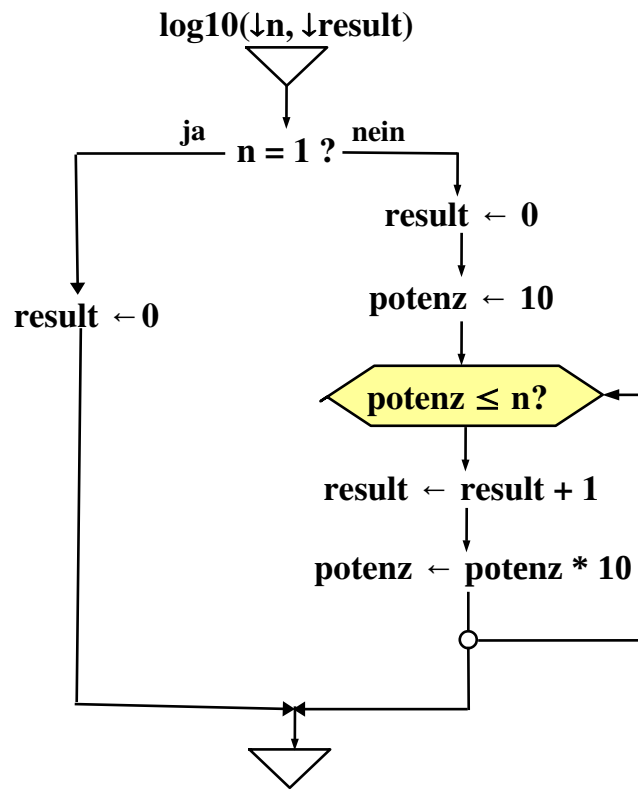
f) Beschreiben Sie in wenigen Worten, was der folgende Algorithmus tut, also welches Problem er löst. Dies können Sie beispielsweise durch einen Schreibtischtest herausfinden.

algorithm($\uparrow a$, $\uparrow b$)



(3 Punkte)

- g) Der folgende Algorithmus berechnet zu einer gegebenen natürlichen Zahl $n \geq 1$ den Logarithmus zur Basis 10, abgerundet auf die nächstkleinere ganze Zahl. Geben Sie je ein Beispiel dafür an, wo im Algorithmus eine Wertzuweisung, Folge, Verzweigung und Schleife eingesetzt wird.



(4 Punkte)

- h) Vereinfachen Sie den Algorithmus aus Aufgabe g), so dass mindestens eine Anweisung eingespart wird. Unzulässig ist die Verwendung des Logarithmus als Rechenoperation. Es genügt, wenn Sie die Änderung beschreiben, Sie müssen kein weiteres Diagramm zeichnen.

(3 Punkte)

- i) Wir betrachten eine Zahlenfolge $(a_n)_{n \in \mathbb{N}}$ mit

$$\begin{aligned} a_0 &= 2 \\ a_1 &= 2 \text{ und} \\ a_{i+1} &= 5 * a_{i-1} - 4 * a_i + 7 \end{aligned}$$

Zeichnen Sie ein Ablaufdiagramm, das zu gegebenem $n \geq 0$ den Wert von a_n berechnet. Sie dürfen höchstens fünf Variablen verwenden, in denen jeweils eine Zahl gespeichert wird.

(10 Punkte)

Teil 3: Abgabe

Schreiben Sie die Lösung am Rechner und speichern Sie sie im PDF-Format (oder als ASCII-Text) ab. Gängige Textverarbeitungsprogramme wie Word oder OpenOffice unterstützen das Exportieren in PDF-Format. Nutzen Sie diese Funktionalität. Achten Sie bitte darauf, dass keine persönlichen Daten (Name, Matrikelnummer, etc.) in Ihrer Abgabe stehen. Erzeugen Sie dann ein Archiv und übertragen Sie es, wie in der Installationsanleitung beschrieben, ins Abgabesystem. Falls Sie vor dem Abgabetermin etwas korrigieren möchten, übertragen Sie einfach das geänderte Archiv. Jede neue Abgabe überschreibt die vorherige. Sie können Ihre Lösung also *nicht* in mehreren Teilen übertragen.

Sollten sich Probleme mit dem Abgabesystem nicht rechtzeitig lösen lassen (siehe Abschnitt 0), ist ausnahmsweise eine Abgabe per Email möglich. Senden Sie dazu ihre Lösung **vor** Abgabeschluss als jar-Archiv (oder pdf, txt) an `support-plm@uni-kassel.de` mit dem Betreff „Java Abgabe“. Auch bei einer Abgabe per Email sollten Sie sich vorab im Abgabesystem registrieren, damit Ihre Lösung erfasst werden kann.

Abgabetermin: Die Lösungen sind bis spätestens Donnerstag, den 26.10.2017 um 8:00 Uhr (strikt!) über das elektronische Abgabesystem einzureichen. Nachträglich eingereichte Lösungen zählen als nicht abgegeben.