Prof. Dr. Carsten Vogt

Praktikum „Praktische Informatik 1“

24.10.2022

Technology
Arts Sciences
TH Köln

Runde 2 – Hausaufgabe

Ende der Bearbeitungsfrist: Montag, 28.11.2022, 12:00 Uhr

Dieses Aufgabenblatt wird zwar schon am 24.10. ausgegeben; der Stoff, den Sie zu seiner Bearbeitung brauchen, wird aber erst in den Vorlesungen am 27.10. und 3.11. behandelt. Dementsprechend endet die Abgabefrist erst am letzten November-Montag. Beginnen Sie aber **möglichst früh** mit Ihrer Arbeit, da dieses Blatt **deutlich anspruchsvoller und umfangreicher als Blatt 1** ist.

Termin- und Themenhinweise:

- Beim Live-Termin zu Runde 2 werden Sie mit Kontrollstrukturen (Verzweigungen und Schleifen) programmieren. Hierzu benötigen Sie unbedingt das entsprechende grundlegende Wissen aus Vorlesungskapitel 6. Ohne dieses Wissen werden Sie die Aufgaben nicht lösen können. Daher wird dies zu Beginn des Termins in einem kurzen Ilias-Test geprüft. Er entscheidet darüber, ob Sie teilnehmen können.
- Vergessen Sie nicht, Ihren **Live-Termin für Runde 3 zu buchen – Frist: 5.12.22, 12:00h**. Nach Ablauf der Frist dürfen Sie nur noch mit Genehmigung durch Prof. Vogt umbuchen. Dies gilt auch für die Termine der Runden 1 und 2, die Sie bereits gebucht haben.

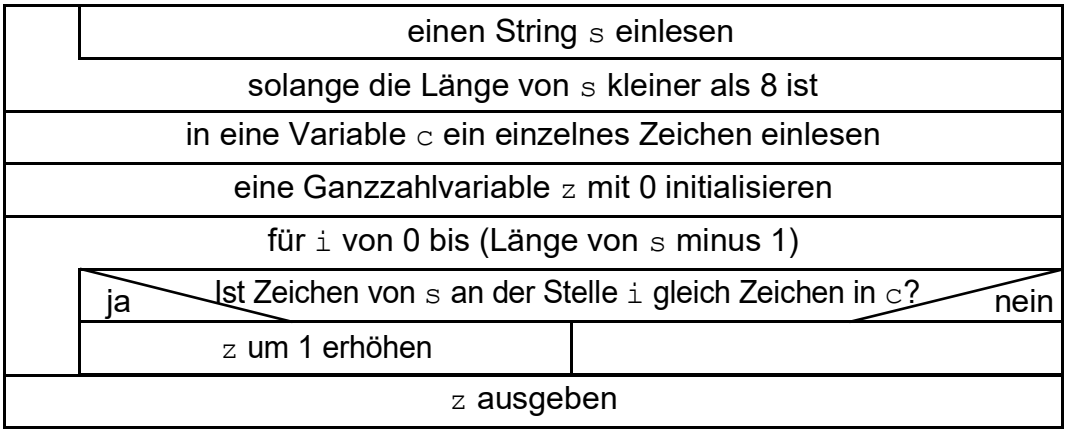
Befolgen Sie bei der Bearbeitung des Aufgabenblatts <u>unbedingt</u> diese Punkte – bitte <u>gründlich</u> lesen, Nichtbeachtung gefährdet Ihren Praktikumserfolg:

- Lesen Sie die einzelnen Aufgaben zunächst gründlich und vollständig durch, bevor Sie mit der Lösung beginnen.
- Erstellen Sie in Ihrem Praktikums-Gruppenverzeichnis im Ilias-PI1-Kurs (Praktikum > Gruppen > Gruppe *XX*) einen Ordner/Unterverzeichnis mit dem Namen *Hausaufgabe02* und legen Sie Ihre Lösungen in diesem Ordner ab. Speichern Sie dort die einzelnen Dateien, also kein ZIP-Archiv oder ähnliches.
- Benennen Sie die Dateien exakt so, wie es in den Aufgaben gefordert ist.
- Erstellen Sie pro Programmieraufgabe *eine* Datei mit dem geforderten Java-Quellcode. Diese Datei muss mit DrJava oder dem `javac`-Befehl zu übersetzen sein.
  Reichen Sie also *keine Eclipse-Projekte* oder ähnliches ein!
- Beachten Sie die Abgabefrist. Wir werden alle Dateien, die nach Ablauf der Frist erstellt oder bearbeitet wurden, bei der Bewertung Ihrer Lösungen nicht berücksichtigen!
- Erarbeiten Sie in Ihrer Gruppe *eine* gemeinsame Lösung, also keine gesonderten Lösungen für die einzelnen Gruppenmitglieder.

Fortsetzung nächste Seite

Aufgabe 2A_1:

Gegeben ist das folgende Struktogramm:



Schreiben Sie ein entsprechendes Java-Programm in einer Datei *Aufgabe_2A_1.java* und führen Sie es aus. Wichtig: Das **Programm soll dem Struktogramm genau entsprechen**; abweichende Lösungen werden nicht anerkannt!

Was macht dieser Algorithmus also? Schreiben Sie die **Antwort** auf die Frage **als Kommentar** oben in die Datei *Aufgabe_2A_1.java*. Die Antwort soll möglichst knapp und treffend sein – beschreiben Sie also nicht die Vorgehensweise, sondern sagen Sie nur, was der Endwert von z besagt!

Zusatzaufgaben für etwas Anspruchsvollere:

- Erweitern Sie Struktogramm und Programm so, dass der Algorithmus mehrfach hintereinander ausgeführt wird, solange der/die Benutzer(in) das wünscht.
- Erweitern Sie das Programm so, dass ein zweiter String erzeugt wird, der aus `s` entsteht, in dem Sie alle Vorkommen von `c` streichen.

Fortsetzung nächste Seite

Aufgabe 2A_2:

Erstellen Sie **zuerst ein Struktogramm** für den folgenden Algorithmus:

- Einlesen einer ganzen Zahl `n` von der Tastatur; dabei Wiederholung der Eingabe, solange `n` kleiner als 1 ist.
- Feststellen, wie viele Ziffern die Binärdarstellung von `n` hat.
  - Beispiele: Für `n`=27 hat die Binärdarstellung 11011 fünf Ziffern, für `n`=92 hat die Binärdarstellung 1011100 sieben Ziffern.
  - Vorgehensweise: Sie müssen die Binärdarstellung von `n` nicht explizit ermitteln. Programmieren Sie stattdessen eine Schleife, in der `n` wiederholt ganzzahlig durch 2 geteilt wird, bis es zu 0 wird. Zählen Sie dabei in einer Zählvariablen, wie oft die Schleife durchlaufen wird.
- Das Resultat auf den Bildschirm ausgeben.

Das Struktogramm muss so detailliert sein, dass in ihm **alle Schleifen und Verzweigungen** des späteren Programms zu erkennen sind. Bitte beachten Sie dazu auch das Video zur systematischen Erstellung eines Struktogramms, das Sie in Ilias finden.

Sie können das Struktogramm entweder mit einem Software-Werkzeug erstellen, oder Sie können es per Hand auf ein Blatt Papier zeichnen, das Sie dann einscannen oder fotografieren. In jedem Fall müssen Sie eine PDF-, JPG- oder PNG-Datei erzeugen; andere Formate werden nicht akzeptiert. Benennen Sie die Datei mit *Aufgabe_2A_2.xxx*, wobei *xxx* für das Dateiformat (pdf, jpg, png) steht.

Schreiben Sie **danach ein Java-Programm** in einer Datei *Aufgabe_2A_2.java*, das **Ihrem Struktogramm genau entspricht**.

Fortsetzung nächste Seite

Aufgabe 2A_3: Kontrollstrukturen

Das Programm in dieser Aufgabe soll ein Dreieck aus Sternen mit `n` Zeilen ausgeben, wobei in der `i`-ten Zeile `i` Sterne stehen. `n` soll beliebig gewählt werden können:

```
*
**
***
****
```
usw., bis n Zeilen erreicht sind

Stellen Sie **zuerst ein Struktogramm** für einen entsprechenden Algorithmus auf:

- Zunächst soll eine ganze Zahl `n` eingelesen werden. Der Einlesevorgang soll wiederholt werden, bis `n` größer als 3 ist.
- Anschließend sollen zwei geschachtelte Schleifen ausgeführt werden. Der Laufindex `i` für die äußere Schleife entspricht den Zeilen des Dreiecks und läuft daher von 1 bis `n`; der Laufindex für die innere Schleife entspricht den Spalten in der jeweiligen Zeile und läuft daher jeweils von 1 bis `i`. Im Körper der inneren Schleife wird jeweils ein einzelner Stern ausgegeben.

Das Struktogramm muss so detailliert sein, dass in ihm **alle Schleifen** des späteren Programms zu erkennen sind. Außerdem wichtig: Es muss deutlich werden, wo ein "**Zeilenvorschub**" stattfindet, d.h. wo zum Beginn einer neuen Zeile umgeschaltet wird.

Erstellen Sie eine Datei *Aufgabe_2A_3.xxx* wie in der vorherigen Aufgabe, auch hier wieder als PDF-, JPG- oder PNG-Datei.

Setzen Sie **danach das Struktogramm in ein Java-Programm** *Aufgabe_2A_3.java* um. Auch hier soll das **Programm dem Struktogramm genau entsprechen**.