Norbert Müller
Universität Trier

Wintersemester 24/25

Miniklausur 2 zur Vorlesung

Grundlagen der Programmierung

Aufgabenblatt

- Sie dürfen NICHT in Gruppen arbeiten. Mit der Teilnahme versichern Sie, dass Sie die Aufgaben alleine bearbeiten. Bearbeitungzeit ist jeweils von **09:00** bis **10:00**. Sie benötigen am Semesterende 50% der Gesamtpunktzahl aller 6 Miniklausuren.
- Die Angaben zur Evaluation bedeuten:
  - Bei "*Evaluation: Zahlen*" wird nur auf Übereinstimmung der Zahlen mit der Vorgabe getestet, sonstige Formatierungen und ausgegebener Text werden ignoriert. Sie brauchen also nicht auf Zeilenenden o.Ä. zu achten.
  - Bei "*Evaluation: Zahlen/Text*" müssen alle ausgegebenen Zeichen mit der Vorgabe übereinstimmen. Die exakte Formatierung (Zeilenumbrüche, etc.) ist jedoch nicht relevant.
  - Die von der Evalierungsroutine ausgewerteten Teile des Ein/Ausgabeprotokolls werden in `blau` dargestellt, die Eingaben in `rot`.
- Es wird bei allen Aufgaben verlangt, Eingaben über den Scanner vorzunehmen. Wenn Sie das Einlesen anders implementieren, werden die Evaluierungen NICHT funktionieren und Sie erhalten KEINE Punkte, selbst wenn Ihre Lösung ansonsten in Ordnung sein sollte.

Aufgabe 1 (Evaluation: Zahlen/Text) (10 Punkte)

Schreiben Sie ein Programm, das zwei `int`-Zahlen n und m einliest.

- Fall $n < m$: Das Programm soll die Summe der Zahlen zwischen n und m (**exklusive** n und **inklusive** m) **mit Hilfe einer `for`-Schleife** bestimmen. Bei $n = -2$ und $m = 2$ müssen Sie also die Summe $(-1) + 0 + 1 + 2 = 2$ berechnen und ausgeben.

  ```
  Eingabe: -2 2
  Ausgabe: 2
  ```

- Für den Fall $n \geq m$ soll keine Berechnung vorgenommen werden, sondern nur eine Fehlermeldung wie folgt ausgegeben werden (d.h. Ihr Ausgabetext wird entsprechend evaluiert):

  ```
  Eingabe: 100 10
  falsche Eingabe
  ```

Hinweis: Nutzen Sie eine zusätzliche Variable für die Summe, die mit 0 initialisiert wird.

Aufgabe 2 (Evaluation: Zahlen) (10 Punkte)

Im vorgegebenen Anfangsteil des Programmes wird zunächst eine ganze Zahl `anzahl` (mit Wert > 0) eingelesen und dann ein Array `int[] data` der Größe `anzahl` erstellt. Dieses Array wird dann mit zufällig bestimmten Zahlen gefüllt, wozu ein Zufallszahlengenerator mit einem zweiten eingelesenen Wert initialisiert wird. Die Zahlen im Array können dabei sowohl positiv als auch negativ sein. Bis hierhin ist das Programm vorgegeben und darf nicht verändert werden.

Sie sollen nun herausfinden, (a) wie viele der Zahlen im Array < 0 sind und (b) wie viele > 0 sind. Als Beispiel ergeben sich bei Eingabe von `3 3` als Array-Inhalt die drei Zahlen `4 -1 6` und damit

```
Anzahl: 3
Zufallsstart: 3
(a) Anzahl negativer Werte: 1
(b) Anzahl positiver Werte: 2
```

Aufgabe 3 (ohne Evaluationsmöglichkeit) (10 Punkte)

Schreiben Sie ein Programm, mit dem Sie (z.B. durch Suche mit einer `while`-Schleifen) die folgenden zwei Zahlen bestimmen können:

- die kleinste positive `int`-Zahl `n` (> 0), für die in Java der boolesche Ausdruck `(n*n*n)/(n*n) == n` den Wert `false` ergibt, sowie
- die kleinste positive `long`-Zahl `m` (> 0), für die in Java der Ausdruck `(m*m*m*m)/m == m*m*m` den Wert `false` ergibt.

Eine Eingabe wird hier nicht benötigt, Sie brauchen also z.B. keinen Scanner. Die Ausgabe soll im Wesentlichen die folgende Form haben:

```
int:  1234
long: 12345
```

Achtung: Die hier angegebenen Werte **1234** und **123456** sind nicht die korrekten Resultate, geben aber grob deren Größenordnung wieder. *Während der Miniklausur* ist auch kein korrekter Wert für die Evaluation vorgegeben, d.h. Sie können Ihr Programm nicht durch die Evaluation auf Korrektheit testen. Die korrekten Werte werden erst für die Korrektur in Moodle eingestellt.

Aufgabe 4 (Evaluation: Zahlen) (10 Punkte)

Programmieren Sie den folgenden Algorithmus:

Deklarieren Sie zwei Variablen: a vom Typ int und x vom Type double.
Lesen Sie in a eine int-Zahl vom Scanner ein. (Sie können annehmen, dass a positiv ist.)
Setzen Sie x auf den Wert 2.
Führen Sie genau zehnmal den folgenden Schleifenrumpf aus:
Geben Sie x aus.
Bestimmen Sie einen neuen Wert für x über die Formel x = (x + a/x) / 2

Es werden also genau 10 `double`-Zahlen ausgegeben. Beispiel:

```
Eingabe: 16
Ausgabe: 2.0 5.0 4.1 4.001219512195122 4.0000001858445895
4.000000000000004 4.0 4.0 4.0 4.0
```

Anmerkung: Dies ist das sogenannte “Heron”-Verfahren zur Berechnung der Quadratwurzel.

Aufgabe 5 (Evaluation: Zahlen/Text) (10 Punkte)

Schreiben Sie ein Programm, das drei `int`-Zahlen k, m und n einliest (in dieser Reihenfolge) und eine Liste der Zahlen zwischen m und n (jeweils inklusive) ausgibt. Allerdings soll dabei jede Zahl, die ohne Rest durch k teilbar ist, durch den Text `beep` ersetzt werden.

Für den Fall $m \leq n$ soll die Liste aufsteigend sortiert sein, wie im folgenden Beispiel:

```
Eingabe: 3 4 10
Ausgabe: 4 5 beep 7 8 beep 10
```

Für den Fall $m > n$ soll die Liste absteigend sortiert sein, wie im folgenden Beispiel:

```
Eingabe: 2 11 3
Ausgabe: 11 beep 9 beep 7 beep 5 beep 3
```

Tipp: Setzen Sie dazu den %-Operator geeignet ein. Sie dürfen davon ausgehen, dass der Wert k positiv ist. Trennen Sie die einzelnen Ausgaben zum Beispiel durch ein oder mehrere Leerzeichen oder Zeilenenden voneinander.