

Name: _____ Abzugeben bis: Mi, 20.10.2010 **12:00**
Matrikelnummer: _____ Bearbeitungsdauer in Stunden: _____
Nummer d. Übungsgruppe: _____ Name des Tutors: _____
Name d. Übungsleiters: _____ Punkte: _____

Aufgabe 1: Mittelwertberechnung

2 + 3 + 3 + 1 Punkte

Entwickeln Sie einen Algorithmus, der folgendes leistet: Es sollen beliebig viele Zahlen (aber mindestens eine Zahl) von einem nicht näher spezifizierten Eingabemedium gelesen werden. Die Eingabe ist beendet, wenn eine abschließende 0 gelesen wird. Dieses spezielle Abschlusszeichen wird nicht mehr mitgerechnet. Es soll von den eingelesenen Zahlen der Mittelwert (Summe aller Zahlen dividiert durch die Anzahl der Zahlen) berechnet und ausgegeben werden.

Beispiele:

Eingabe: 1 12 8 9 0

Ausgabe: Gesamtanzahl: 4 Mittelwert: 7.5

Eingabe: 2 0

Ausgabe: Gesamtanzahl: 1 Mittelwert: 2

- Beschreiben Sie den von Ihnen entwickelten Algorithmus in *Prosa*.
- Stellen Sie diesen Algorithmus als *Ablaufdiagramm* dar.
- Stellen Sie diesen Algorithmus als *Struktogramm* dar.
- Vergleichen Sie die drei Darstellungsformen, geben Sie deren Unterschiede an und begründen Sie, welche Darstellungsform sich am besten für welchen Typ von Algorithmus eignet.

Hinweise: Zum Einlesen der Zahlen können Sie `read(↑number)` verwenden. Die Variable `number` enthält nach so einem Aufruf immer die nächste Zahl. Die Ausgabe des Ergebnisses geschieht bspw. mit `write(↓result)`. Mit einem Aufruf der Form `write(↓"Das ist ein Text")` können auch Texte ausgegeben werden.

Aufgabe 2: Fibonacci-Zahlen

2 + 4 Punkte

Entwerfen Sie einen (nicht rekursiven) Algorithmus, der alle Fibonacci-Zahlen im Bereich von 1 bis 100 ausgibt. Die Fibonacci-Reihe ist wie folgt definiert:

$$f(0) = 1$$

$$f(1) = 1$$

$$f(n) = f(n-1) + f(n-2)$$

- Geben Sie eine Lösungsidee (textuell) für den Algorithmus an.
- Stellen Sie den Algorithmus als *Struktogramm* dar.

Hinweis: Zum Ausgeben können Sie wiederum die Funktion `write(↓number)` verwenden (siehe Aufgabe 1).

Aufgabe 3: Zinsberechnung

4 + 5 Punkte

Es ist ein Algorithmus gesucht, der folgendes leistet: Für einen bestimmten Geldbetrag und einen bestimmten Zinssatz soll ausgegeben werden, wie sich dieser Wert Jahr für Jahr erhöht (jeweils Ausgabe von Jahr und Wert). Betrag, Zinssatz sowie Anzahl der auszugebenden Jahre sollen bereits beim Programmstart (jeweils ganzzahlig) eingelesen werden. Ist eine der eingelesenen Zahlen 0 oder negativ so soll der Algorithmus mit Ausgabe der „Wert(e) ungültig, keine Berechnung möglich!“ beendet werden.

Für die „Geldvermehrung“ kann folgende einfache Formel verwendet werden:

$$\mathbf{Kapital = Kapital * (100 + Zinssatz)/100}$$

Beispiele:

Eingabe: 100
5
3

Ausgabe: 1 105.0 2 110.25 3 115.7625

Eingabe: 50
-2
1

Ausgabe: Wert(e) ungültig, keine Berechnung möglich!

- Stellen Sie diesen Algorithmus als *Ablaufdiagramm* dar.
- Versuchen Sie den unter 3a) entwickelten Algorithmus in ein lauffähiges Java-Programm zu übersetzen. **Für Benutzereingaben (read-Operationen) ist (bis inkl. Übung 5) zwingend die Input-Klasse zu verwenden (Download im Begleitmaterialordner zur Übung!).** Überprüfen Sie die Funktionsweise zumindest mit den oben angegebenen Testfällen, schließen Sie der Abgabe ein Testprotokoll an (Ein-/Ausgaben des Programms).

Hinweis: Zum Einlesen können Sie die Funktion `read(↑number)`, zum Ausgeben die Funktion `write(↓number)` verwenden (siehe Aufgabe 1).

Das 2. Übungsblatt baut auf Aufgabe 3 (Zinsberechnung) auf – bitte kopieren Sie sich deshalb Ihre abgegebene Lösung (Übung 1 ist mit hoher Wahrscheinlichkeit im Bearbeitungszeitraum von Übung 2 noch nicht korrigiert bzw. retourniert).