

Name: _____ Abzugeben bis: Mi, 23.10.2002 12:00

Matrikelnummer: _____ Bearbeitungsdauer in Stunden: _____

Nummer d. Übungsgruppe: _____ Name des Tutors: _____

Name d. Übungsleiters: _____ Punkte: _____

Aufgabe 1: Größte Werte

3 + 4 + 4 + 1 Punkte

Entwickeln Sie einen Algorithmus, der folgendes leistet: Es sollen beliebig viele nicht-negative Zahlen von einem Eingabemedium gelesen werden. Die Eingabe ist beendet, wenn eine abschließende negative Zahl gelesen wird. Von den eingelesenen Zahlen sollen die zwei größten Werte berechnet und ausgegeben werden. Wenn keine oder nur eine Zahl eingegeben wurde, soll eine entsprechende Fehlermeldung ausgegeben werden.

Beispiele:

Eingabe: 1.5 2 0 4.25 1 -1
Ausgabe: Groesste Zahlen: 4.25 und 2

Eingabe: 5.25 -3.2
Ausgabe: Zuwenig Zahlen eingegeben!

- Beschreiben Sie den von Ihnen entwickelten Algorithmus in *stilisierter Prosa*.
- Stellen Sie diesen Algorithmus als *Ablaufdiagramm* dar.
- Stellen Sie diesen Algorithmus als *Struktogramm* dar.
- Vergleichen Sie die drei Darstellungsformen. Welche gefällt Ihnen am besten? (*Mit Begründung!*)

Hinweis: Für die Punkte b) und c) können Sie zum Einlesen der Zahlen `read(↑number)` verwenden. Die Variable `number` enthält nach so einem Aufruf immer die nächste Zahl. Die Ausgabe des Ergebnisses erfolgt mit `write(↓result)`. Mit einem Aufruf der Form `write(↓"Das ist ein Text")` können Texte ausgegeben werden.

Aufgabe 2: Primfaktorenberechnung

8 + 4 Punkte

Entwickeln Sie einen Algorithmus, der die Primfaktoren einer Zahl berechnet. Die Primfaktoren sollen dabei aufsteigend und mit ihrer Häufigkeit ausgegeben werden. Die Ausgabe soll in dem, in den Beispielen angegebenen Format erfolgen (d.h. mit "*" und "^"). Zum Einlesen können Sie die Funktion `read(↑number)`, zum Ausgeben die Funktion `write(↓number)` verwenden.

Beispiele:	Eingabe	Ausgabe	Eingabe	Ausgabe
	10	2 ¹ * 5 ¹	6534	2 ¹ * 3 ³ * 11 ²
	256	2 ⁸	13332	2 ² * 3 ¹ * 11 ¹ * 101 ¹

- Stellen Sie diesen Algorithmus als *Ablaufdiagramm* oder als *Struktogramm* dar. Tipp: Entwickeln Sie den Algorithmus schrittweise, z.B. erst nur Primfaktoren, dann mit Potenzen, ...
- Versuchen Sie, den unter a) entwickelten Algorithmus in ein lauffähiges Java-Programm zu übersetzen. Vergessen Sie nicht, das Programm zu testen und die Ein-/Ausgabe für die Tests mit abzugeben.

Hinweis: Sie können die 0. Übung als Vorlage verwenden und die `main`-Methode entsprechend anpassen.