

Belegaufgaben für den Studiengang Keramik-, Glas- und Baustofftechnik (KGB) zur Lehrveranstaltung "Grundlagen der Informatik"

1) Hinweise zu den Belegaufgaben

- Die Studentinnen und Studenten des Studiengangs

Keramik-, Glas- und Baustofftechnik (KGB)

müssen *gemäß der Prüfungsordnung des Studiengangs KGB* einen Beleg anfertigen und verteidigen, um zur Prüfung (Klausur) zugelassen zu werden.

- Als Beleg ist selbstständig ein Python-Programm zu entwickeln. Dazu kann aus drei Aufgaben (siehe Abschnitt 2) eine zu lösende Aufgabe selbstständig ausgewählt werden. Der Beleg ist vor dem entsprechenden Übungsleiter zu verteidigen (CD bzw. Diskette mit Lösungsdatei mitbringen).
→ **Bitte vereinbaren Sie dazu rechtzeitig einen Termin mit dem Übungsleiter.**

- Der **Beleg muss spätestens 7 Tage vor der Klausur verteidigt worden sein**, um zur Klausur zugelassen zu werden.

- Jede Studentin und jeder Student meldet die gewählte Aufgabe bitte zur Registrierung bei Frau Schüttauf an. Zu melden sind Name, Matrikelnummer, Studiengang und die Nummer des Belegs. Die Meldung kann persönlich, telefonisch (03731 / 39 29 44) oder per Email (schuetti@informatik.tu-freiberg.de) erfolgen.

- Die Aufgabentexte können entweder bei Frau Schüttauf kopiert, auf dem WWW-Server des Instituts für Informatik (gleiche WWW-Seite wie die Aufgaben und Lösungen zur Lehrveranstaltung) oder im bekannten Verzeichnis *Y:\Lehre* der PC-Pools aufgerufen werden.

2) Belegaufgaben

Die folgenden Belegaufgaben sind jeweils für eine Studentin/einen Student vorgesehen. Wählen Sie eine Aufgabe aus den folgenden 3 Aufgaben aus:

Aufgabe 1: Volumen-, Oberflächen- und Massenberechnung

Für 5 unterschiedliche geometrische Körper (Quader, Kugel, Kegel,...) und 6 Materialien (Eisen, Holz, Wasser, ...) sind **gesonderte Funktionen** (!!) zu implementieren, in denen jeweils Volumen, Oberfläche und Masse für einen speziellen Körper und alle Materialien berechnet werden.

Alle für die Berechnung erforderlichen Eingaben, Funktionsaufrufe und Resultatausgaben sind im Hauptprogramm vorzunehmen. Nach jeder Berechnung und Resultatausgabe ist der Nutzer zu fragen, ob er Berechnungen für einen weiteren Körper vornehmen will oder nicht.

Es ist nicht mit globalen, sondern nur mit lokalen Variablen zu arbeiten.

Die Aufgabe kann fakultativ beliebig erweitert (, aber nicht eingeschränkt) werden.

Aufgabe 2: Ratespiel („1 aus 100“):

Geben Sie ein vollständiges Python-Programm für folgendes Ratespiel an:

Spielschritt 1: Mit dem Zufallszahlengenerator ist zunächst eine zu erratende ganze Zahl a aus dem Bereich $1 \leq a \leq 100$ zu bestimmen, die der Spieler natürlich nicht kennen darf.

Spielschritt 2: - Der Spieler muss a nun durch Eingabe einer ganzen Zahl b zu erraten versuchen.

- Solange er *falsch* rät:
 - Hinweis durch Ausgabe, ob b zu groß oder zu klein ist.
 - Danach muss der Spieler erneut raten.
- Wenn richtig geraten:
 - Dann ist auf den Bildschirm **und** außerdem in ein Festplattenfile auszugeben:
 - Anzahl n der benötigten Rateversuche
 - Außerdem eine verbale Leistungseinschätzung in Abhängigkeit von n und zwar:

--- $n = 1$ oder 2 :	Super
--- $n = 3$ oder 4 :	Sehr gut
--- $n = 5$:	Gute Leistung
--- $n = 6, 7$ oder 8 :	Mittelmäßige Leistung
--- $n > 8$:	Schwache Leistung

- Nach Ende einer Spielrunde: Fragen, ob Spieler noch einmal spielen will.

Wenn ja, dann Fortsetzung bei Spielschritt 1, sonst Spielende.

(Die Ausgaben nach jeder Raterunde sind in das gleiche Festplattenfile zu schreiben.)

Der Spieler ist durch aussagekräftige Ausgaben auf dem Bildschirm durch das Spiel zu führen.

Es ist nicht mit globalen, sondern nur mit lokalen Variablen zu arbeiten.

Die Aufgabe kann fakultativ beliebig erweitert (, aber nicht eingeschränkt) werden.

Aufgabe 3: Äquidistante Klassifizierung von reellen Messwerten

(Die Klassifizierung von Messwerten ist eine wichtige Anwendungsaufgabe in allen Wissenschaftsgebieten, vor allem in den Ingenieurwissenschaften.)

Gegeben sei eine unbekannte Anzahl reeller Messwerte, die von einem Messgerät in ein ASCII-Festplattenfile geschrieben worden ist (ein Messwert pro Zeile). Diese auf der Festplatte vorliegenden Messwerte sollen nun in einem vollständigen Python-Programm gemäß einer äquidistanten Klasseneinteilung („äquidistant“ heißt: identische Klassenbreite) klassifiziert werden.

Die Klassifizierung ist in folgenden Schritten vorzunehmen:

- zunächst sind zur Orientierung des Nutzers zu bestimmen und auszugeben: die Anzahl der Messwerte, der minimale und der maximale Wert der Messwerte sowie der Mittelwert.
- dann ist die Klasseneinteilung über Tastatur einzugeben (z.B. durch die unterste Klassengrenze ug , die Klassenbreite kb und die Klassenanzahl ka); eine Klasse soll dabei ein links abgeschlossenes und rechts offenes Intervall sein.
- die absoluten und relativen Klassenhäufigkeiten sind zu berechnen (es empfiehlt sich, vorher die Messwerte der Größe nach zu sortieren)
- außer den Häufigkeiten für die vom Nutzer eingegebenen Klassen ist vom Programm auch zu berechnen, wieviele Messwerte links von der untersten bzw. rechts von der obersten Klasse liegen, d.h. es sind auch die absoluten und relativen Häufigkeiten für folgende Klassen zu ermitteln:

- Klasse Nr. 0 : $(-\infty, u_g)$
- Klasse Nr. $ka+1$: [oberste Klassengrenze, $+\infty$)
- auf dem Bildschirm sind **tabellarisch** für jede Klasse auszugeben:
 - die Klassennummer
 - die Klassengrenzen
 - die absolute und relative Klassenhäufigkeit

Die einzelnen Moduln des Programms (Berechnung der Klassenhäufigkeiten, Tabellenausgabe, evtl. Sortierung, ...) sind als **gesonderte Funktionen** zu schreiben und jeweils vom Hauptprogramm aus aufzurufen.

Es ist nicht mit globalen, sondern nur mit lokalen Variablen zu arbeiten.

Die Aufgabe kann fakultativ beliebig erweitert (, aber nicht eingeschränkt) werden.

Beispiel (zum besseren Verständnis der Aufgabe):

- Im Festplattenfile sollen folgende Messwerte stehen (eine Zahl pro Zeile, n unbekannt):

x[0]=1.2, x[1]= -10.0, x[2]=5.0, x[3]=3.1, x[4]=3.9,
 x[5]=4.1, x[6]= 3.1, x[7]=0.0, x[8]=4.6, x[9]=2.3

- Daraufhin wird vom Programm berechnet und ausgegeben:
 anzahl = 10, min = -10.0, max = 4.6, mw = 1.73

- Als Klasseneinteilung werde nun eingegeben:
 ug=0.0, kb=2.5, ka=3.0

- Dann könnte die Tabelle zum Beispiel wie folgt aussehen:

Klassennummer.	Klassenintervall	Klassenhäufigkeit	
		absolut	relativ
0	(-unendl., 0.0)	1	0.010
1	[0.0, 2.5)	3	0.030
2	[2.5, 5.0)	5	0.050
3	[5.0, 7.5)	1	0.010
4	[7.5, +unendl.)	0	0.000

Wir wünschen Ihnen viel Erfolg!

(Ri 10/2006)