Grundlagen der Programmierung 1

Prof. Dr. Detlef Krömker
Alexander Wolodkin

GOETHE UNIVERSITÄT FRANKFURT AM MAIN

Übungsblatt Nr. 4

Ausgabe: 08.11.2017
Abgabe: 17.11.2017
9.30 Uhr

Datentypen und Kontrollstrukturen

Hinweis: *Es sind grundsätzlich Rechenwege anzugeben, es sei denn es findet sich ein expliziter Hinweis, dass dies nicht nötig ist. Es dürfen keine Lösungen aus dem Skript, dem Internet oder anderen Quellen abgeschrieben werden. Diese Quellen dürfen nur mit Quellenangaben verwendet werden und es muss ein hinreichend großer Eigenanteil in den Lösungen deutlich zu erkennen sein. Digitale Abgaben, die nicht im Format .pdf für Texte oder .py für Code erfolgen, werden nicht bewertet. Bei Abgaben mehrer Dateien müssen diese als .zip zusammengefasst werden. Abgaben, die nicht diesen Regeln entsprechen, werden nicht bewertet! Achten Sie darauf die Variable* `__author__` *in allen Quellcode Dateien korrekt zu setzen. Abgaben, die nicht dieser Vorgabe entsprechen, werden nicht bewertet. Außerdem muss Ihr Name in jeder abgegebenen .pdf Datei zu finden sein. Ababen, die vollständig per Hand geschrieben und eingescannt werden, sind nur in zuvor abgesprochenen Ausnahmefällen erlaubt.*

Σ __ / **9**

Aufgabe 4.1: **IEEE-754** Punkte: ____ / 5

In der Vorlesung wurde der IEEE-754 Standard zur Darstellung von Gleitpunktzahlen vorgestellt.

(a) (1 Punkt) Welchen Wert hat die Gleitpunktzahl (nach IEEE-754, single precision)

0 10000111 10000010100101010000010

als Dezimalzahl?

(b) (1 Punkt) Was ist die betragsmäßig kleinste Ganzzahl, die nicht mehr fehlerfrei von `int` nach `double` (64-Bit) konvertiert werden kann? Wie kommen Sie zu diesem Ergebnis?

(c) (1 Punkt) Im Falle der einfachen Genauigkeit wird ein *Bias* von 127, bzw. 1023 bei doppelter Genauigkeit, verwendet. Welcher wesentlich Vorteil ergibt sich hieraus im Gegensatz zur Kodierung des Exponenten im Zweierkomplement?

(d) (1 Punkt) Warum ist es unerlässlich für das halblogarithmisch IEEE Verfahren auch negative Exponenten darzustellen?

(e) (1 Punkt) Beim Rechnen mit Gleitkommazahlen können Fehler auftreten. Auf was sollte man daher bei der Verwendung von Gleitkommazahlen achten? Wie können wir uns vor diesen Fehlern schützen?

Aufgabe 4.2: **Kontrollstrukturen** Punkte: ____ / 4

(a) (1 Punkt) Überführen Sie die Ackermannfunktion

$$ack(n,m) = \begin{cases} m+1 & ,n = 0 \\ ack(n-1,1) & ,m = 0 \wedge n \neq 0 \\ ack(n-1, ack(n, m-1)) & \end{cases}$$

in eine rekursive Python Funktion.

(b) (1 Punkt) Geben Sie eine iterative Implementierung der folgenden Funktion an.

$$f(n) = \begin{cases} n & , n \leq 1 \\ f(n-1) + f(n-2) & \end{cases}$$

(c) (1 Punkt) Schreiben Sie ein Programm, welches Wörter „null“, „eins“, „zwei“, „drei“, „vier“, „fuenf“, „sechs“, „sieben“, „acht“ und „neun“ durch Komma getrennt und ohne Leerzeichen entgegen nimmt und diese in ihre Dezimalzahlendarstellung wandelt und mit print() ausgibt: „eins,zwei,drei“ = 123

Geben Sie das Ergebnis aus. Beschränken Sie sich hierbei auf maximal 4 Dezimalstellen.

(d) (1 Punkt) Entfernen Sie die print-Anweisung aus dem Programm des Aufgabenteils (*c*) und ergänzen Sie ihr Programm um ein Analogon einer (*fußgesteuerten*) do-while Schleife.

Tipp: Die Schleife läuft mindestens einmal durch.

Bei jedem Durchlauf:
• Dekrementieren Sie die entgegengenommene ganzzählige Zahl.
• Beenden Sie die Schleife genau dann, wenn das Ergebnis einem Wert von $''False + 2''$ entspricht. Geben Sie in diesem Fall $''True''$ aus.
Tipp: Schauen Sie sich die Übung PRG1 03 genauer an...
• Fangen Sie mit einer Ausgabe "Out of range" Zahlenwerte ab, die die Abbruchbedingung nicht erreichen würden.