

Übungsblatt Nr. 2¹

Ausgabe: 25.10.2017
Abgabe: 04.11.2017
16.00 Uhr

Kontrollstrukturen

Hinweis: Achten Sie darauf die Variable `__author__` in **allen** Quellcode Dateien **korrekt** zu setzen. Aufgaben, die nicht dieser Vorgabe entsprechen, werden **nicht** bewertet! Quellcode muss als `.py` Datei und alles Weitere als `.pdf` Datei abgegeben werden. Bei Abgaben mehrerer Dateien müssen diese als `.zip` zusammengefasst werden. Abgaben, die nicht diesen Regeln entsprechen, werden ebenfalls **nicht** bewertet!

Σ ____ / 10

Aufgabe 2.1: Warm-Up mit Kontrollstrukturen

Punkte: ____ / 5

- (a) (2 Punkte) Schreiben Sie ein Python 3.6 Programm, welches einen Satz vom Benutzer entgegen nimmt und bestimmt, ob dieser ein Pangramm ist, d.h. ob er alle Buchstaben des Alphabets (ohne Sonderzeichen) mindestens einmal enthält – unabhängig von Groß- oder Kleinschreibung. Nutzen Sie built-in String Funktionen! (Die Dokumentation solcher Funktionen finden Sie hier: <https://docs.python.org/3/library/stdtypes.html#text-sequence-type-str>)
- (b) (3 Punkte) Schreiben Sie ein Python 3.6 Programm, das eine ganze Zahl im Intervall [0,999] vom Benutzer entgegen nimmt und nach folgenden Regeln die Ausgabe erzeugt:
 - 1. Wenn die Zahl durch 3 teilbar ist oder eine Ziffer der Zahl 3 ist, soll `fizz` ausgegeben werden.
 - 2. Bei jeder Zahl, die durch 7 teilbar ist oder eine Ziffer 7 ist, soll `buzz` ausgegeben werden.
 - 3. Falls (1) **und** (2) beide erfüllt sind, soll `fizzbuzz` ausgegeben werden.
 - 4. Ist keine der drei Bedingungen erfüllt, soll die eingegebene Zahl ausgegeben werden.

Überlegen Sie genau, was der Nutzer Ihres Programms alles falsch machen könnte und fangen Sie dies ab. Falls etwas falsch eingegeben wird, geben Sie `ERROR` aus.

Aufgabe 2.2: Professor Ungerecht

Punkte: ____ / 5

Professor Ungerecht hat eine sehr eigene Bewertungsmethode.

Zunächst ermittelt er die Vornoten der Prüfungsleistungen für Module nach den Vorgaben des § 26 Absatz 2 bis 4 der Bachelorordnung Informatik, siehe http://www.cs.uni-frankfurt.de/images/pdf/informatik/bachelor2/bachelorordnung_neu.pdf. Dann „korrigiert“ er die Note:

¹Es 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.

- Ist die Person weiblich, verbessert er die Note um 0.4 Notenwerte.
- Ist die Person größer als 180 cm (der Körpergröße von Professor Ungerecht), so addiert er für jeden weiteren cm 0,05 Notenpunkte auf den bereits ermittelten Notenwert.
- Abschließend rundet er die Note gemäß der Vorgaben der Prüfungsordnung.

Als Hardcore-Informatiker hat sich Professor Ungerecht zur Berechnung ein Assembler-Programm geschrieben, das als Eingabe einen String der Form

Vornote Geschlecht Körpergröße (alles in einer Zeile, getrennt jeweils durch ein Leerzeichen, ohne prompt).

von der Konsole erwartet, mit folgenden Einschränkungen:

- Die Vornote wird als Float-Literal notiert, immer mit genau drei Zeichen, z.B. 2.3 oder 3.0 Geschlecht wird mit einem Zeichen kodiert: w steht für weiblich, jedes andere Zeichen steht für nicht weiblich.
- Die Körpergröße wird als Integer in cm angegeben, immer dreistellig im Intervall [100,300], z.B. 185 für 0,25 Zuschlag zum Notenwert.
- Als Ergebnis gibt das Programm dann die gerundete „korrigierte“ Note aus.

Tun Sie es Professor Ungerecht nach. Schreiben Sie auch ein solches Programm in Python 3.6. Sie dürfen davon ausgehen, dass die Syntax der Eingabe präzise eingehalten wird, möglicherweise aber nicht die Wertebereiche.

Überlegen Sie genau, was der Nutzer Ihres Programms alles falsch machen könnte und fangen Sie diese Fehler ab. Überprüfen Sie dies durch geeignete Testfälle (entsprechend des Überblicks zum Testen, welcher in der Vorlesung gegeben wird). Falls unzulässige Werte eingegeben werden, geben Sie ERROR aus.