2. Aufgabenblatt zur Vorlesung

Konzepte der Programmierung

WiSe 2025/26

Kristin Knorr

Abgabe Abgabe 31.10.2025 23:59 Uhr

Aufgabe 1 Galgenmännchen *10 Punkte+2 Zusatzpunkte für ASCII art*

Programmieren Sie ein Galgenmännchenspiel für zwei Spieler:innen:

(1.) Nutzen Sie die `input` Funktion, um die Namen der Spieler:innen zu setzen. Weisen Sie dafür jedem Namen eine Variable zu.

(2.) Spieler:in 1 soll nun aufgefordert werden, ein Wort einzugeben. Es soll geprüft werden, ob die Eingabe der Vorgabe entspricht (Es sind nur Buchstaben zulässig). Nutzen Sie dafür die Funktion `isalpha()`. Diese kann mit `x.isalpha()` auf einem String `x` aufgerufen werden und gibt genau dann `True` zurück, wenn der String nur aus Buchstaben besteht.

(3.) Spieler:in 2 bekommt die Information, wie lang das Wort ist und soll das Wort erraten.

Bis hier hin reicht es aus alle Schritte einmalig auszuführen. Nun müssen wir den Vorgang des Ratens so lange wiederholen bis Spieler:in 2 gewonnen oder verloren hat. Nutzen Sie dafür eine Schleife. Wir begrenzen die Anzahl der Runden auf 10.

(4.) Ihr Programm soll nun abhängig vor der Eingabe des Spielers/der Spielerin 2 Hinweise geben. Dabei müssen 2 Fälle betrachtet werden (Sie können die Funktionen `x.lower()` oder `x.upper()` nutzen, um die Eingabe `x` in Kleinbuchstaben bzw. Großbuchstaben umzuwandeln.):

(a) Ein eingegebener Buchstabe kommt im Wort vor. Dann soll das Programm ausgeben, an welchen Positionen im Wort der Buchstabe vorkommt. Das kann geschehen, indem die Positionen benannt werden oder mittels einer optischen Ausgabe mit Unterstrichen. Es muss auch behandelt werden, wenn das Wort gelöst wurde.

(b) Wenn SpielerIn 2 einen Buchstaben eingibt, der nicht im Wort vorkommt, wird auch das mitgeteilt. Auch hier können gern kreative Lösungen gefunden werden beispielsweise ein Wort, das sich aufbaut, oder das Galgenmännchen in ASCII art.

(5.) Wenn SpielerIn 2 10-mal falsche Buchstaben geraten hat, hat er/sie verloren. Bei kreativen Lösungen kann die Anzahl falscher Buchstaben ggf. variieren.

Aufgabe 2 Listen und Tupel in Python *2+3+5*

Anna und Elsa wollen eine Party vorbereiten. Zuerst machen sie sich Gedanken über das Buffet und müssen Rezepte heraussuchen, die gut aus der Hand zu essen sind. Anschließend schreiben Sie einen Einkaufszettel mit den zusammengefassten Zutaten. Helfen Sie den beiden dabei, indem Sie die Vorgänge des Heraussuchens der Rezepte und des Schreiben des Einkaufszettels automatisieren. Gehen Sie dafür

davon aus, dass das Rezeptbuch eine Liste aus Tupeln der Form `(<Rezeptstichwort>,[(<Menge>,<Zutat>)])` ist. Nutzen Sie bei der Lösung der Aufgabenteile (a) und (b) mindestens eine `while`-Schleife.

(a) Schreiben Sie ein Programm, das eine Schleife nutzt, um aus der Liste alle Rezepte mit dem Rezeptstichwort `fingerfood` herauszusuchen und diese in einer Liste sammelt.

(b) Schreiben Sie nun unter Nutzung von Schleifen einen Ablauf, der eine Liste aller benötigten Zutaten erstellt (noch ohne Menge).

(c) Nutzen Sie die Listen aus (a) und (b) um mit Schleifen, die Einkaufsliste mit den aufsummierten Mengen zu erstellen (Liste aus Tupeln `(<Menge>,<Zutat>)`). Es ist hilfreich, zuerst eine Einkaufsliste mit Menge 0 für jede Zutat zu initialisieren. Beachten Sie beim Aktualisieren der Mengen, dass Tupel unveränderbar sind

Zum Testen können Sie die Liste `rezepte` aus `rezeptListe.py` nutzen.

Aufgabe 3 Syntaxbäume und Logik *5+5 Punkte*

In der Vorlesung haben wir einen Syntaxbaum zu einem arithmetischen Term erstellt. Nun wollen wir Syntaxbäume logischer Ausdrücke betrachten. Bei den logischen Operatoren nimmt die Bindungsstärke in der Reihenfolge `not`, `and`, `or` ab.

(a) Klammern Sie den Ausdruck `x or y and not(y or z)` vollständig und zeichnen Sie den zugehörigen Syntaxbaum. Geben Sie an wie der Term ausgewertet wird, wenn die Variablen, wie folgt, belegt sind: (i) x=`True`, y=`False`, z=`True` bzw. (ii) x=`False`, y=`False`, z=`False`

(b) Nun betrachten wir Ausdrücke, wie sie in Bedingungen im Code vorkommen. Hier gilt, dass Vergleichsoperatoren immer stärker binden als die Logikoperatoren. Klammern Sie den Ausdruck `x < y or y >= 10 and not z` vollständig und zeichnen Sie den zugehörigen Syntaxbaum. Geben Sie an wie der Ausdruck ausgewertet wird, wenn die Variablen, wie folgt, belegt sind: (i) x=20, y=15, z=`True` bzw. (ii) x=3, y=10, z=`False`