MATHEMATISCHES INSTITUT
PROF. DR. ACHIM SCHÄDLE
MARINA FISCHER

HEINRICH HEINE
UNIVERSITÄT DÜSSELDORF

18.10.2018

Computergestützte Mathematik zur linearen Algebra – 2. Übungsblatt

Hinweis: Bearbeiten Sie das Blatt in SPYDER. Erstellen Sie für jede Aufgabe ein neues Skript und speichern Sie diese als *AufgabeX.py*, wobei X die jeweilige Aufgabennummer ist.

Aufgabe 5: (*Einführung Kontrollstrukturen*)

Befehle: `break`, `elif`, `else`, `format`, `if`, `for`, `while`, `randint`
Tick, Trick und Track wollen ihre 50 Kekspackungen auf einer Straße mit 20 Häusern verkaufen. Sie gehen dafür die Häuser der Reihe nach ab und verkaufen an jeder Tür maximal 5 Packungen.
Simulieren Sie diese Situation mit Schleifen und `if`-Abfragen folgendermaßen:

- Die Bewohner von Haus X kaufen Y Packungen. Hier soll Y zufällig zwischen 0 und 5 sein (siehe Hinweis unten). Das wird durch folgende Ausgabe angezeigt:
  Die Bewohner von Haus X kaufen Y Packungen Kekse.
  Bei *einer* Packung wird
  Die Bewohner von Haus X kaufen eine Packung Kekse.
  ausgegeben. Sollten an einem Haus keine Kekse verkauft werden, so wird stattdessen ausgegeben:
  Die Bewohner von Haus X möchten keine Kekse.
- Wenn alle Kekse verkauft wurden, hören Tick, Trick und Track auf. Dann wird ausgegeben:
  Alle Kekspackungen wurden verkauft!
  Ansonsten soll
  Es sind noch X Kekspackungen übrig.
  ausgegeben werden. Achten Sie auch hier wieder auf den Sonderfall mit einer Kekspackung.
- Achten Sie auch darauf, dass man nur so viele Packungen verkaufen kann, wie man auch wirklich noch hat.

Hinweis: Nach der Ausführung des Befehls `from numpy.random import randint` erhalten Sie mit `randint(6)` eine ganze Zufallszahl (gleichverteilt in $[0, 6)$).

Aufgabe 6: (*arithmetisches Mittel*)

Befehle: `for`, `assert`, `isinstance`, `*`
Schreiben Sie zwei Funktionen `amean` und `bmean` zur Berechnung des arithmetischen Mittels. Hierbei soll

(a) die Funktion `amean` als Eingabe beliebig viele Elemente, und

(b) die Funktion `bmean` als Eingabe eine Liste mit beliebig vielen Elementen

erhalten. Als Ausgabe sollen beide Funktionen jeweils das arithmetische Mittel der Elemente liefern. Prüfen Sie sowohl in `amean` als auch in `bmean` mit dem Befehl `isinstance(X, (int,float,complex))`, ob es sich bei jedem betrachteten Element `X` tatsächlich um eine gültige Zahl handelt und geben Sie andernfalls eine sinnvolle Fehlermeldung aus.
Verwenden Sie beide Funktionen um den Mittelwert der ganzen Zahlen von 1 bis 15 zu bestimmen.

b.w.

Aufgabe 7: (*Fibonacci-Zahlen*)

Gegeben sei folgender Schnipsel PYTHON-Code:

```
def myPrint(ebene, msg):
    print(("*" * ebene) + "␣" + str(msg));
def fib(n):
    if n <= 1:
        myPrint(n, 1);
        return 1;
    F_nm2 = fib(n - 2);
    F_nm1 = fib(n - 1);
    F = F_nm1 + F_nm2;
    myPrint(n, "{0}␣=␣{1}␣+␣{2}".format(F, F_nm1, F_nm2));
    return F;
```

`fib(n)` soll die n-te Fibonacci-Zahl rekursiv berechnen. Den obigen Code zum Rauskopieren und die Definition der *Fibonacci-Zahlen* finden Sie im Vorlesungsskript.

(a) Korrigieren Sie den kleinen Fehler.

(b) Kommentieren Sie jede Zeile der Funktion `fib` sinnvoll.

(c) Erläutern Sie die Ausgaben der Aufrufe `fib(1)`, `fib(3)` und `fib(5)`.

Aufgabe 8: (*Maschinengenauigkeit*)

Befehle: `while`, `format`

(a) Bestimmen Sie die kleinste positive Zahl `eps`, so dass in PYTHON $1 + \texttt{eps} > 1$ gilt.
Beginnen Sie dafür bei $\texttt{eps} = 1$ und halbieren Sie Ihren Kandidaten so oft wie nötig.

(b) Lassen Sie sich `eps` in der Form 2^{-n}, $n \in \mathbb{N}$, ausgeben.

Besprechung in den Übungen vom 22.-26. Oktober 2018.