MATHEMATISCHES INSTITUT
PROF. DR. ACHIM SCHÄDLE
MARINA FISCHER

HEINRICH HEINE
UNIVERSITÄT DÜSSELDORF

18.10.2018

Computergestützte Mathematik zur Analysis – 2. Übungsblatt

Hinweis: Bearbeiten Sie das Blatt in JUPYTER. Gehen Sie dazu wie in Aufgabe 0 von Blatt 1 vor.

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: (*Generatoren*)

Befehle: `* for * in *`, `format`

(a) Erstellen Sie die beiden Listen
`Wert = ['7','8','9','10','Bube','Dame','König','Ass']` und
`Symbol = ['Karo','Herz','Kreuz','Pik']`.

(b) Erstellen Sie nun mit Hilfe von Generatoren eine Liste `Deck`, deren Einträge die Kombinationen aus den Listen `Wert` und `Symbol` sind, d.h. eine Liste mit den 32 Einträgen
`Deck = ['Karo 7','Karo 8',...,'Karo Ass','Herz 7',...]`.

(c) Führen Sie die Befehle `from numpy.random import shuffle` und `shuffle(Deck)` aus. Lassen Sie sich dann `Deck` anzeigen. Was hat sich geändert?

(d) Erstellen Sie nun mit der neuen Liste `Deck` und Generatoren ein Dictionary mit den Einträgen

```
{ 0: 'Die Karte mit der Nummer 1 ist Kreuz König',
  1: 'Die Karte mit der Nummer 2 ist Pik 9',
  ...,
  31: 'Die Karte mit der Nummer 32 ist Herz König'}
```

Hierbei sollen die Zahlen $0, \dots, 31$ sowie $1, \dots, 32$ wie im Beispiel sortiert sein, die Reihenfolge der Karten im Deck kann sich jedoch unterscheiden.

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 auf der Homepage der CompLA.

(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:

Befehle: `while`, `assert`

Zur Bestimmung einer Näherung an eine Nullstelle der Funktion $f(x) = x^2 - a$ mit $a \geq 0$ kann das Newton-Verfahren verwendet werden, das Sie in Numerik 1 genauer kennenlernen werden. Für die gegebene Funktion und mit einem Startwert $x_0 \in \mathbb{R}$ lautet dessen Iterationsvorschrift

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)} = x_k - \frac{x_k^2 - a}{2x_k} \quad \text{für} \quad k = 0, 1, 2, \ldots.$$

(a) Schreiben Sie eine Funktion `x = mynewton(a,x0)`, die mit der obigen Vorschrift eine Approximation an eine Nullstelle von $f(x) = x^2 - a$ berechnet. Hierbei ist `a` $\in \mathbb{R}$ und `x0` $\in \mathbb{R}$ ist ein beliebiger Startwert. `mynewton` soll die Approximation `x` zurückgeben, falls $\left|\frac{\mathtt{x}^2-\mathtt{a}}{2\mathtt{x}}\right| < 10^{-8}$ erreicht ist. Prüfen Sie zu Beginn, ob `a` ≥ 0 gilt und geben Sie andernfalls eine sinnvolle Fehlermeldung aus.

(b) Lassen Sie nun in Ihrer Funktion `mynewton` einen Zähler mitlaufen, der die Schleifendurchläufe zählt, und geben Sie in jedem Schritt

Die Näherung in Schritt X ist Y.

aus. X ist hierbei die Nummer des jeweiligen Schleifendurchlaufs und Y die Näherung, die in diesem Schritt berechnet wurde.

(c) Testen Sie Ihre Funktion mit `a` $= 2$ und `x0` $= 2$. Geben Sie die Differenz zwischen der berechneten Näherung `x` und der exakten Lösung $x^* = \sqrt{a}$ aus.

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