Informatikklausur Nr. 2
Klasse 10
Schuljahr 2016/2017, 2. Halbjahr
2. Mai 2017

Aufgabe 1:
Herr Riecken unterrichtet gerade eine 5. Klasse in Chemie. Er ist vollkommen verzweifelt: Die meisten Versuche in der Klasse klappen nicht, weil die Schülerinnen und Schüler entweder nicht zuhören oder die Anleitung nicht lesen. Als letzten Ausweg sieht Herr Riecken nur einen Ablaufdiagramm für diesen Versuch.

1. Lies dir die Versuchsbeschreibung (Material 1) zu diesem Versuch genau durch.
2. Entwickle ein übersichtliches PAP-Diagramm (Programmablaufdiagramm) für die Durchführung dieses Versuchs.

Raute = Verzweigung / Parallelogramm = Eingabe / Rechteck = sonstiger Programmablaufschritt

Aufgabe 2:
In Material 2 findest du eine Beschreibung des sogenannten Bubblesortalgorithmus.

a) Führe die Tabelle aus dem Material fort, indem du die Inhalte der nächsten 10 Zeilen voraussagst.
b) In folgendem Code ist der Bubblesortalgorithmus in Python realisiert:

```
010  def bubbleSort(alist):
020       for passnum in range(len(alist)-1,0,-1):
030            for i in range(passnum):
040                 if alist[i]>alist[i+1]:
050                      temp = alist[i]
060                      alist[i] = alist[i+1]
070                      alist[i+1] = temp

080  alist = [54,26,93,17,77,31,44,55,20]
090  bubbleSort(alist)
100  print(alist)
```

Erkläre das Programm, indem du für jede Zeile notierst, was der Algorithmus hier tut.

Ganz zentral ist hier die range()-Funktion:

```
range([Startwert], Abbruchwert[, Endwert])
```

Aufgabe 3:
In Material 3 findest du gängige Schreibungen für reguläre Ausdrücke.

a) Gibt jeweils drei Beispiele für Zeichenketten an, die auf folgende reguläre Ausdrücke passen:
   (1) \d\d.\d\d.\d\d
   (2) (abc)*[\d]
   (3) (abc).[\d]
   (4) [1-9][0-9]{3}
   (5) ^[123456][abcde]*
   (6) $[(com),(de)]

b) Formuliere einen regulären Ausdruck, der die folgende Bedingungen erfüllt:

   (1) Eine fünfstellige Zahl vor einem a oder b am Anfang einer Zeichenkette
   (2) Eine gültige De-Domain (endet auf „.de“ und muss mit einem Kleinbuchstaben beginnen)
   (3) Eine gültige Schulnote (1-6, darf ein „-“ oder „+“ vorangestellt haben)
   (4) Ein Datum im Format tt.mm.jjjj (Tag und Monat zweistellig, Jahr vierstellig)

c) Bei Aufgabe 3b.(4) sind an bestimmtem Stellen nur bestimmte Zahlen zulässig, z.B. darf die erste Ziffer des Tages nur zwischen 0 und 3 liegen. Korrigiere deinen Ausdruck so weit, wie es möglich ist und beschreibe zusätzlich, welche Problemfälle weiterhin auftreten.

d) Schüler deines Kurses schlagen folgende Ausdrücke zur Überprüfung einer E-Mailadresse vor:

   (1) [@][a-z]{2,10}[\.][a-z]{2,6}
   (2) [a-z.+{6,30}@.+\.[^\.][com,de]
   (3) ^[A-Za-z]*[@][A-Za-z]*[.][a-z][a-z]

   Der Backslash „\“ sorgt dafür, dass der Punkt als Punktzeichen und nicht als regulärer Ausdruck gewertet wird.

   Diskutiere die jeweiligen Vor- und Nachteile der Schülerlösungen.

Aufgabe 4:

a) Was macht dieses Programm? Beschreibe jede Zeile.

```
10    import re
20    text = input("Gib deine Postleitzahl an: ")
30    pattern = "[0-9]*5"
40    matcher = re.search(pattern, text)
50    if len(text) == 5 and matcher :
60        print("Gültig")
70        f = open('text.txt', 'a')
80        if f:
90            f.write(text+"\n")
100           f.close()
110           f = open('text.txt', 'r')
120           x = f.read()
130           print(x)
140           f.close()
150       else:
160           print("Fehler. Datei konnte nicht erzeugt werden")
170   else:
180       print("Leider konnten wir deine Postleitzahl nicht finden. Versuch es
          nochmal. ")
```

b) Modifiziere das Programm so, dass mehrere Eingaben hintereinander möglich sind.

Material 1:

Versuch zur Bestimmung der Löslichkeit

Stelle einen Erlenmeyerkolben auf eine Waage und betätige die Tara-Taste. Fülle ein Reagenzglas mit fünf Millilitern deionisiertem Wasser. Gib nun eine kleine Portion Kochsalz hinzu und schüttele so lange, bis sich das Salz vollständig aufgelöst hat. Wiege nun das Reagenzglas mit Inhalt, indem du es in den Erlenmeyerkolben auf der Waage stellst und notiere das angezeigte Gewicht. Drücke danach nicht die Tarataste! Wiederhole den Versuch so lange, bis sich die hinzugegebene Kochsalzportion nicht mehr auflöst. Arbeite unbedingt immer mit kleinen Mengen, sodass sich die Stoffportion auch immer vollständig auflösen kann.
Wenn du fertig bist, wiederhole den gesamten Versuch mit Haushaltszucker (Saccharose).

Material 2:
In der Bubble-Phase wird die Eingabe-Liste von links nach rechts durchlaufen. Dabei wird in jedem Schritt das aktuelle Element mit dem rechten Nachbarn verglichen (grau). Falls die beiden Elemente das Sortierkriterium verletzen, werden sie getauscht.

Beispiel:

Element 1	Element 2	Element 3	Element 4	Element 5	Element 6	Element 7	Element 8	Element 9	Element 10
6	5	3	1	8	7	9	0	2	4
5	6	3	1	8	7	9	0	2	4
5	3	6	1	8	7	9	0	2	4
5	3	1	6	8	7	9	0	2	4
5	3	1	6	8	7	9	0	2	4

Am Ende der Phase steht bei auf- bzw. absteigender Sortierung das größte bzw. kleinste Element der Eingabe am Ende der Liste.

Die Bubble-Phase wird solange wiederholt, bis die Eingabeliste vollständig sortiert ist. Dabei muss das letzte Element des vorherigen Durchlaufs nicht mehr betrachtet werden, da die restliche zu sortierende Eingabe keine größeren bzw. kleineren Elemente mehr enthält.

Je nachdem, ob auf- oder absteigend sortiert wird, steigen die größeren oder kleineren Elemente wie Blasen im Wasser (daher der Name) immer weiter nach oben, das heißt, an das Ende der Liste. Auch werden immer zwei Zahlen miteinander in „Bubbles“ vertauscht.

Quelle: *https://de.wikipedia.org/wiki/Bubblesort*

Material 3:

Regulärer Ausdruck	Erläuterung	Beispiel

a	Das Zeichen „a“	`re.search(a,“Hallo“)`
		
		`„Hallo“ auf das Vorhandensein von „a“ prüfen`

. (Punkt)	Ein beliebiges Zeichen	`re.search(a.,“Hallo“)`
		
		`„Hallo“ auf das Vorhandensein von „a“ gefolgt von einem beliebigen Zeichen prüfen`

[abc]	Ein beliebieges Zeichen aus der Menge {a,b,c}	
\d	Eine Ziffer [0-9]	
\w	Buchstabe, Ziffer oder Unterstrich [a-zA-Z_0-9]	
X|Y	X oder Y	
X+	Mindestens einmal X	
$X	X am Ende	
^X	X am Anfang	
{n}	Der vorangehende Ausdruck muss exakt n mal vorkommen	
{n,m}	Der vorangehende Ausdruck muss mindestens n mal vorkommen und darf höchstens m mal vorkommen	
(xyz)	Gruppierung xyz müssen miteinander vorkommen	

*	Beliebige Zeichen	`re.search(a*,“Hallo“)`
		
		`„Hallo“ auf das Vorhandensein von „a“ gefolgt von beliebig vielen Zeichen prüfen`

+	Und	`re.search(a+b,“Hallo“)`
		
		`In „Hallo“ müssen die Zeichen „a“ und „b“ vorkommen`