

Unit 10a

Dr. Günter Kolousek

21. Juli 2015

Lege wiederum ein Verzeichnis an. Nennes es `14_unit10a`! In diesem Verzeichnis sollen alle Dateien der jeweiligen Einheit abgelegt werden.

1 Schulübungen

1. Schreibe ein Programm `pi.py`, das die Zahl π berechnet.

$$\pi = \frac{4}{1} - \frac{4}{3} + \frac{4}{5} - \frac{4}{7} + \dots$$

Das Programm soll insgesamt `n` Summanden aufsummieren. `n` wird wieder vom Benutzer eingegeben.

Am Ende soll der Fehler berechnet werden, indem der Absolutbetrag der Differenz zur im `math`-Modul eingebauten Konstanten `pi` berechnet wird.

Welche Gesetzmäßigkeit lässt sich in der Formel bzgl. der Zähler und Nenner der Brüche erkennen? Welche Gesetzmäßigkeit bzgl. der Vorzeichen?

2. Schreibe eine Funktion `changeE(s)`, die einen String als Parameter bekommt und einen String zurückliefert, der alle enthaltenen `e` in Großbuchstaben gewandelt hat.

Für die Spezialisten: Nein, die `replace` Methode darf jetzt nicht verwendet werden.

3. Schreibe eine Funktion `easter(year)`, die sowohl das Monat als auch den Tag des Ostersonntags berechnet und zurückliefert. Verwende dazu den Algorithmus von Carl Friedrich Gauß (wird als einer der größten Mathematiker bezeichnet) aus dem Jahr 1816:

```
a = Jahr mod 19
b = Jahr mod 4
c = Jahr mod 7
k = Jahr div 100
p = (8k + 13) div 25
q = k div 4
```

```

M = (15 + k - p - q) mod 30
d = (19a + M) mod 30
N = (4 + k - q) mod 7
e = (2b + 4c + 6d + N) mod 7
Wenn 22 + d + e kleiner oder gleich 31:
    Ostern = (22 + d + e)ter März
anderenfalls:
    Ostern = (D+E-9).April

```

Beispiel:

```
monat, tag = easter(2011)
```

Denke daran, dass die Python Coding Conventions eingehalten werden, auch wenn der in Pseudocode formulierte Algorithmus diese nicht einhält.

4. Verwende folgenden in Pseudocode verfassten Algorithmus zum schnellen Potenzieren zweier Zahlen um eine Python Funktion `pow(a, b)` zu schreiben:

```

x := a; y := b; z := 1;
while y > 0 do
begin
    if odd(y) then z := z*x;
    y := y div 2;
    x := x*x;
end;

```

In der Variablen `z` befindet sich danach das Ergebnis.

Hinweise zur Notation dieses Pseudocodes:

- a) Zuweisung durch `:=` anstatt durch `=` wie in Python
 - b) Blöcke werden durch `do`, `begin` und `end` gekennzeichnet und nicht wie in Python durch `:` und Einrückung.
 - c) Jede Anweisung wird durch ein `;` abgeschlossen. In Python können einfache Anweisungen durch ein `;` abgeschlossen werden, müssen jedoch nur, wenn mehrere einfache Anweisungen in einer Zeile stehen.
5. Schreibe eine Funktion `count_chars(s, c)`, die die Anzahl der Buchstaben im String `s` zurückliefert, die durch den Buchstaben im Parameter `c` festgelegt worden sind.

D.h. wenn die Funktion mit `count_chars("abcabcabc", "a")` aufgerufen wird, dann liefert dieser Funktionsaufruf 3 zurück.

6. Schreibe eine Funktion `egyptian_mul(a, b)`, die den Algorithmus der ägyptischen Multiplikation implementiert.

Der Algorithmus ist folgendermaßen formuliert:

- a) Man schreibt die beiden zu multiplizierenden Zahlen nebeneinander.
 - b) Auf der linken Seite werden die Zahlen jeweils halbiert (bei ungeraden Zahlen wird abgerundet). Die Ergebnisse werden untereinander geschrieben, bis man zur 1 gelangt.
 - c) Auf der rechten Seite werden die Zahlen verdoppelt und untereinander geschrieben. Die rechts stehenden (verdoppelten) Zahlen werden gestrichen, wenn die links stehende Zahl gerade ist.
 - d) Die Summe der nicht gestrichenen rechts stehenden Zahlen ergibt das gesuchte Produkt.
7. Schreibe eine Funktion `is_subset(a, b)`, die `True` zurückliefert, wenn die Menge `a` eine Teilmenge der Menge `b` ist. Der Operator `<=` bzw. `<` darf nicht verwendet werden.
8. Schreibe eine Funktion `union(a, b)`, die die Vereinigungsmenge von `a` und `b`, also $a \cup b$ zurückliefert.

Der Operator `|` darf nicht verwendet werden. Ein Element `e` kann man einer Menge `m` mittels der Methode `m.add(e)` hinzugefügt werden.

9. Schreibe eine Funktion `intersect(a, b)`, die die Schnittmenge von `a` und `b`, also $a \cap b$ zurückliefert.

Der Operator `&` darf nicht verwendet werden.

10. Schreibe eine Funktion `diff_set(a, b)`, die die Differenzmenge von `a` und `b`, also $a \setminus b$ zurückliefert.

Der Operator `-` darf nicht verwendet werden.