

Aufgabe 1

- 1) Nennen Sie jeweils ein **Beispiel** für einen festprogrammierten und einen speicherprogrammierten Rechner. Was sind die fundamentalen Unterschiede zwischen den beiden Rechnerarten?
- 2) Benennen Sie **Vorteile** von festprogrammierten Rechnern im Vergleich zu speicherprogrammierten Rechnern. In welchem Umfeld können festprogrammierte Rechner einen Vorteil gegenüber speicherprogrammierten Rechnern ausspielen?
- 3) Was bedeutet, dass eine Programmiersprache Turing-vollständig ist? Welche **Auswirkung** hat das auf die Wahl der Programmiersprache **in der Praxis**?

Aufgabe 2

Schreiben Sie folgendes Programm mit einem iterativen Ansatz: Geben Sie die Summe aller ungeraden Zahlen von 1 bis n aus. n ist hierbei Teil der Summe, sofern n auch ungerade ist.

Beispiel:

Summe jeder ungeraden Zahl von 1 bis 11: $1 + 3 + 5 + 7 + 9 + 11 = 36$

Aufgabe 3

Schreiben Sie folgendes Programm mit einem iterativen Ansatz: Finden und benennen Sie alle natürlichen Zahlen zwischen 1 und n , die durch die Zahl 5 ohne Rest teilbar sind. n ist ein vom Benutzer angegebener Wert. Weiterhin soll n auch benannt werden, sofern n ohne Rest durch die Zahl 5 teilbar ist.

Beispiel:

Natürliche Zahlen zwischen 1 und 20, die durch 5 ohne Rest teilbar sind: 5, 10, 15, 20

Aufgabe 4

Palindrome sind Zeichenketten, die von vorne und von hinten gelesen das identische Wort ergeben. Beispiele sind Wörter wie *Lagerregal*, *Elle*, *Ebbe*, *Ehe*, *Neffen*, *Retsina-Kanister*, *Reliefpfeiler* oder auch *Rentner*.

Schreiben Sie bitte eine Funktion, die prüft, ob es sich (ohne Berücksichtigung der Groß- und Kleinschreibung) bei einer Zeichenkette um ein Palindrom handelt.

Aufgabe 5

Schreiben Sie folgende kleinere Programme. Der Nutzer kann hierbei einen String `input_string` angeben.

- 1) Bestimmen Sie die Anzahl Konsonanten in `input_string`
- 2) Geben Sie alle vorkommenden Zeichen von `input_string` in umgekehrt alphabetischer Reihenfolge aus

Aufgabe 6

Geben Sie auf der Kommandozeile einen vereinfachten Stern mit Zahlen aus. Die Zahlen laufen von 0 bis 9 und beginnen dann wieder von vorne. Hierfür wird der Nutzer um Angabe der maximalen Breite/Höhe (B/H) des Sterns gebeten. Die minimale Breite beträgt 5. Zur Vereinfachung dürfen Sie davon ausgehen, dass nur ungerade Zahlen vorgegeben werden.

Im Folgenden drei Beispielausgaben für unterschiedliche Breiten/Höhen (B/H):

Stern mit B/H 5:	Stern mit B/H 7:	Stern mit B/H 9:
<pre> 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 </pre>	<pre> 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 </pre>	<pre> 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 </pre>

Aufgabe 7

Zwei vom Nutzer angegebene Zahlen `input_number_a` und `input_number_b` sollen – ohne Verwendung des Divisionsoperators von Python – geteilt werden. Zur Vereinfachung nehmen wir an, dass `input_number_a ≥ input_number_b > 1`. Schreiben Sie hierfür bitte folgende zwei Programme:

- 1) Division mit Hilfe der linearen Suche
- 2) Division mit Hilfe der binären Suche

Aufgabe 8

Das Volumen einer Kugel ergibt sich aus dem Radius durch folgende Gleichung:

$$V = \frac{4}{3} \cdot \pi \cdot r^3$$

Der Nutzer möchte nun zu einem gegebenen Volumen V den zugehörigen Radius r bestimmen.

Hierbei soll in der Lösung jedoch auf die Verwendung des Wurzel-Operators in Python verzichtet werden. Die Lösung soll durch ein Näherungsverfahren bestimmt werden. Schreiben Sie daher bitte folgende zwei Funktionen:

- 1) Bestimmung des Radius r zu einem Volumen V mit Hilfe der linearen Suche
- 2) Bestimmung des Radius r zu einem Volumen V mit Hilfe der binären Suche

Erstellen Sie einen einfachen Benchmark: Wie viele Iterationen benötigen Ihre zwei Implementierungen?

Aufgabe 9

Implementieren Sie das Finden der Kubikwurzel mithilfe des Newton-Raphson-Verfahrens und erstellen Sie einen vergleichenden Benchmark mit der Implementierung auf Basis der binären Suche.

Aufgabe 10

Gegeben sei die Funktion

$$f(x) = x^3 - 2x + 3x - 5$$

Finden Sie eine Nullstelle dieser Funktion ausgehend vom Startwert $x_0 = 2$ mithilfe der binären Suche sowie mithilfe des Newton-Raphson-Verfahrens.

Aufgabe 11

Die Potenz einer positiven Zahl lässt sich folgendermaßen als Produkt darstellen:

$$a^b = \underbrace{a \cdot a \cdots a}_{b \text{ mal}} = a \cdot a^{b-1}$$

Schreiben Sie ein rekursives Programm, welches die Potenz a^b durch einen rekursiven Ansatz berechnet, wobei a die Basis und b der Exponent ist.

Aufgabe 12

Erläutern Sie kurz die Grundidee von rekursiven Algorithmen. Welche Vorteile bringen diese mit sich und was sind mögliche Herausforderungen, die es für Entwickler zu betrachten gibt.

Schreiben Sie eine Funktion, die zwei vom Nutzer angegebene Zahlen (`input_number_a` und `input_number_b`) mithilfe eines rekursiven Ansatzes miteinander multipliziert. Zur Vereinfachung nehmen wir an, dass der Wert beider Zahlen ≥ 0 ist.

Aufgabe 13

Modellieren Sie folgende Zusammenhänge in UML:

Eine Bibliothek hat mehrere Regale. Regale haben eine Kapazität und eine Identifikationsnummer. In den Regalen stehen unterschiedliche Medien. Medien können Bücher, DVDs oder Zeitschriften sein. Medien haben folgende Attribute: Identifikationsnummer, Titel, Erscheinungsjahr, Anschaffungspreis. Weiterhin hat die Bibliothek Nutzer mit folgenden Attributen: Name und NutzerID.

- 1) Modellieren Sie die Zusammenhänge als UML-Modell

- 2) Implementieren Sie die Klassen Ihres UML-Modells
- 3) Implementieren Sie folgende zwei Funktionen und führen diese beispielhaft aus:
 - a. Einstellen eines Buchs in ein Regal
 - b. Ausleihen desselben Buchs aus dem Regal

Zur Vereinfachung können Sie annehmen, dass es aktuell nur ein Regal in der Bibliothek gibt.

Aufgabe 14

Gegeben sei die folgende – offensichtlich nicht sehr robust implementierte – Funktion namens `get_ratio_at_position`, die zwei Listen von Zahlen und einen Index als Parameter entgegennimmt. Die Funktion soll den Quotienten der Zahlen in den Listen am gegebenen Index zurückgeben.

prog_ss23/src/aufgabe03.py

```
def get_ratio_at_position(list_a, list_b, pos_in_list):
    """Assumes: list_a and list_b are lists of equal length of numbers
    Returns: List_a[pos_in_list]/List_b[pos_in_list]"""

    if (
        not isinstance(list_a, list)
        or not isinstance(list_b, list)
        or not isinstance(pos_in_list, int)
    ):
        raise TypeError

    return float(list_a[pos_in_list]) / float(list_b[pos_in_list])

# Beispielaufruf und Überprüfung
assert get_ratio_at_position([2, 3, 6, 9], [1, 2, 3, 4], 2) == 2

assert not get_ratio_at_position([2, 3, 6, 9], [1, 2, 3, 4], 3) == 3
```

Bitte überarbeiten Sie die **Funktion** `get_ratio_at_position` um folgendes Verhalten:

- a) Sofern die beiden ersten Parameter keine Liste oder der dritte Parameter kein Integer ist, wird aktuell intern eine Exception vom Typ **TypeError** erzeugt. Bitte fangen Sie die Exception ab und geben eine benutzerdefinierte Fehlermeldung aus.
- b) Ergänzen Sie die Funktion um die Behandlung von Exceptions vom Typ **IndexError**, wenn der angegebene Index außerhalb des Bereichs der Liste liegt.
- c) Ergänzen Sie die Funktion um die Behandlung von Exceptions vom Typ **ZeroDivisionError**, wenn eine Division durch Null erfolgt.
- d) Wie können Sie den Fehler abfangen, wenn die Listen nicht nur Zahlen enthalten, sondern z.B. einen String und dadurch die Division fehlschlägt?
- e) Nur wenn **keine Exception** auftritt, geben Sie den modifizierten String zurück.

Wirtschaft
Hauptcampus

H O C H
S C H U L E
T R I E R

Beenden Sie das Programm immer mit einer Nachricht, die bestätigt, dass die Funktion `get_ratio_at_position` ausgeführt wurde, unabhängig davon, ob eine Exception aufgetreten ist oder nicht.