

Einführung in die Programmierung
Rechnerklausur WiSe 2024/2025 – 90 Minuten
12.02.2025

Bevor Sie mit der Bearbeitung der Aufgaben am Rechner beginnen, lesen Sie bitte sorgfältig die folgenden Hinweise zum Ablauf und die Bewertungskriterien auf der folgenden Seite:

1. Prüfen Sie Ihr Klausurexemplar auf Vollständigkeit. Die Klausur umfasst zusammen mit diesem Hinweisblatt **7 Seiten**.
2. Es gibt für Sie ein **eigenes Verzeichnis auf dem K-Laufwerk**. Ihre Python-Datei mit den Lösungen muss in diesem Verzeichnis gespeichert werden. Werden Dateien an einem anderen Ort gespeichert, werden diese nicht bewertet.
3. Ihre gesamte Datei sollte am Ende **ohne Fehler lauffähig** sein, ansonsten gibt es erhebliche Punktabzüge. Achten Sie also vor dem letzten Speichern darauf, dass es in der Datei keine Syntaxfehler gibt. Außerdem sollte es während der Ausführung keine Ausnahmen geben.
4. Auf dem Rechner, auf dem Sie arbeiten, dürfen während der Bearbeitungszeit **nur die Anwendung Spyder und Ihr Programm** ausgeführt werden. Falls während oder nach der Klausur, z.B. anhand von Log-Dateien, festgestellt wird, dass weitere Programme ausgeführt wurden, wird dies als Täuschungsversuch gewertet und hat die in der Prüfungsordnung festgelegten Konsequenzen.
5. Tragen Sie in die folgenden Felder **leserlich** Ihren Namen, Vornamen und Ihre Matrikelnummer ein und geben Sie die Klausur am Ende der Bearbeitungszeit ab.

Name	
Vorname	
Matrikelnummer	

Viel Erfolg!

Bewertung

Aufgabe	1	2	3	4	5	Σ
Max.Pkt	8	15	10	12	25	70
Ergebnis						

Bewertungskriterien:

1. **Korrektheit:** Die in der Aufgabenstellung angegebenen Anforderungen an die Berechnungsergebnisse werden erfüllt. Das umfasst mindestens die vorgegebenen Testfälle sowie zusätzliche Angaben in der Aufgabenstellung.
2. **Lesbarkeit und Verständlich:** Variablen sind angemessen benannt und Anweisungen angemessen kommentiert.
3. **Alle Funktionen haben einen Docstring** in dem die Schnittstelle der jeweiligen Funktion prägnant beschrieben ist.
4. Die **algorithmische Vorgehensweise** zur Lösung der Aufgabe ist plausibel und nachvollziehbar.

1. Aufgabe

(8 Punkte)

Schreiben Sie eine PYTHON-Funktion `vokale(w)`, die die Anzahl der Vokale in der Zeichenkette `w` zurückliefert. Zählen Sie alle Vorkommen von A, a, E, e, I, i, O, o, U, u.

Ergänzen Sie zusätzlich zu den vorhandenen Aufrufen noch mindestens 5 weitere Testfälle und notieren Sie deren Sollergebnis jeweils in einer Kommentarzeile.

2. Aufgabe

(15 Punkte)

Schreiben Sie eine PYTHON-Funktion `fib_liste(n)`, die eine Liste der ersten $n \geq 0$ Fibonacci-Zahlen zurückliefert und dabei **keine Rekursion** verwendet. Zur Erinnerung:

$$fib(0) = 0, fib(1) = 1 \text{ und } fib(n) = fib(n-1) + fib(n-2) \text{ für } n \geq 2$$

Schreiben Sie eine weitere PYTHON-Funktion `fib_gerade(k)`, die eine Liste der ersten $k \geq 1$ geraden Fibonacci-Zahlen zurückliefert. Überlegen Sie, wie Sie dabei die Funktion `fib_liste` verwenden können.

Bsp.: Die ersten $k = 4$ geraden Fibonacci-Zahlen sind $fib(0)$, $fib(3)$, $fib(6)$ und $fib(9)$.

3. Aufgabe

(10 Punkte)

Schreiben Sie eine **rekursive** PYTHON-Funktion `entferne_zeichen(w, c)`, die alle Vorkommen des Zeichens `c` aus dem String `w` entfernt und den resultierenden String zurückliefert. Verwenden Sie als Basisfall für `w` die leere Zeichenkette.

Probeklausur

4. Aufgabe

(12 Punkte)

Berechnen Sie in einer Funktion `pi_approx10()` eine Annäherung an π mit Hilfe der Formel

$$\pi^2 = 6 \cdot \sum_{n=1}^{\infty} \frac{1}{n^2} = 6 \cdot \left(\frac{1}{1^2} + \frac{1}{2^2} + \frac{1}{3^2} + \frac{1}{4^2} + \dots \right)$$

indem Sie nur die ersten 10 Summanden berücksichtigen und geben Sie das Ergebnis zurück.

Wie groß ist die Differenz zu `math.pi`?

Generalisieren Sie Ihre Funktion zu `pi_approx(k)`, indem Sie einen default-Parameter für die Anzahl `k` der Summanden einführen.

5. Aufgabe

(25 Punkte)

Am Ende der Saison liegen die Ergebnisse aller Heim- und Auswärtsspiele einer Mannschaft in einer Liste S vor (Beispiel siehe PYTHON-Datei). Nach dem Kennzeichen 'H' für Heim- oder 'A' für Auswärtsspiel folgen immer zuerst die erzielten Tore der jeweiligen Heimmannschaft, danach die erzielten Tore der Gastmannschaft. Schreiben Sie PYTHON-Funktionen zur Auswertung solcher Ergebnislisten:

1. Die Funktion `tordiff(S)` liefert die Differenz zwischen allen erzielten Toren und allen Gegentreffern.
2. Die Funktion `ergebnis(S)` liefert die Anzahl der gewonnen Spiele, die Anzahl der verlorenen Spiele und die Anzahl der Unentschieden.
3. Die Funktion `heimstark(S)` prüft, ob mehr Heimspiele als Auswärtsspiele gewonnen wurden.
4. In wievielen Spielen sind insgesamt zwei Tore gefallen? Schreiben Sie eine Funktion `torhist(S)`, die in einem Dictionary für alle in S auftretenden Gesamttore pro Spiel (Schlüssel) die Anzahl der Spiele mit dieser Toranzahl (Wert) zuordnet.