

Probeklausur

Informatische Werkzeuge in den Geistes- und Sozialwissenschaften I

11. Januar 2020

Diese Probeklausur ist absolut freiwillig und soll Ihnen dazu dienen die Klausur besser einzuschätzen.

Falls gewünscht, *können* Lösungen an die Tutoren zur Korrektur eingereicht werden. In jedem Fall wird nächste Woche eine Musterlösung bereitgestellt.

Die vorliegenden Übungen sind in etwa vergleichbar mit tatsächlichen Klausuraufgaben, daher haben wir auch am Rand (hoffentlich realistisch geschätzte) Zeiten angegeben.

Bitte beachten Sie die folgenden Regeln, um keine Punkte zu verlieren:

- Wenn Sie eine Antwort auf einer anderen Seite fortsetzen, geben Sie bitte die Nummer der Aufgabe auf der neuen Seite mit an und verweisen Sie auf der alten Seite auf die neue.
- Begründen Sie Ihre Aussagen, wenn angebracht (wir würden gerne Teilpunkte für unvollständige Antworten geben). Wenn nicht explizit darum gebeten, antworten Sie möglichst nicht einfach mit „Ja“, „Nein“ oder „42“.

1 Grundlagen & Begrifflichkeiten

Aufgabe 1.1 Erklären Sie die Begriffe *Bit*, *Byte*, *Kilobyte* und *Mebibyte*. Erläutern Sie außerdem kurz die Beziehungen, die diese Begriffe untereinander haben. 5 Pkt
5 min

Aufgabe 1.2 In digitalen Dokumenten, was ist der Unterschied zwischen einem *Plaintext*-Format und einem *Markup*-Format? 5 Pkt
5 min

Aufgabe 1.3 Erläutern Sie kurz, was es mit der python-Funktion `input()` auf sich hat. Ist es möglich, Argumente an diese Funktion zu übergeben? Wenn ja: was passiert mit diesen Argumenten? Wenn nein: was passiert, wenn jemand es trotzdem tut? Worauf ist zu achten, wenn Zahlen eingelesen werden sollen? 5 Pkt
5 min

2 Digitale Dokumente

Aufgabe 2.1 (HTML Ping Pong)

Geben Sie den Quellcode für zwei HTML-Seiten (`ping.html` und `pong.html`) an. 15 Pkt

Beide Seiten sollten eine Überschrift, einen kurzen Text und einen Knopf (nicht einfach nur einen Link) enthalten. Beim Klicken des Knopfes soll auf die jeweils andere Seite weiter geleitet werden, so dass durch wiederholtes Klicken zwischen den Seiten hin und her gesprungen werden kann. 15 min

3 Reguläre Ausdrücke

Aufgabe 3.1 (Reguläre Ausdrücke I)

Geben Sie einen regulären Ausdruck an, der *genau* folgende Strings komplett matcht (also alle diese und keine anderen): 5 Pkt
5 min

`cat4, hat4, mat4, bat4, apollo`

Aufgabe 3.2 (Reguläre Ausdrücke für Adressen)

Geben Sie einen regulären Ausdruck an, der gegen Adresszeilen der Form „91058 Erlangen“ oder „33649 Bielefeld“ matcht. 5 Pkt
5 min

Diese bestehen immer aus einer fünfstelligen PLZ, einem Leerzeichen und einem Ortsnamen. Ortsnamen sind für den Rahmen dieser Aufgabe zwischen einem und 20 Zeichen lang, beginnen mit einem Großbuchstaben (inkl. Umlauten) gefolgt von lediglich Kleinbuchstaben (inkl. Umlauten + ß), ohne Leerstellen oder Interpunktion.

Aufgabe 3.3 (Reguläre Ausdrücke für IP-Adressen)

Geben Sie einen regulären Ausdruck an, der gegen IPv4-Adressen matcht. Diese Adressen bestehen aus genau vier Blöcken, die durch Punkte voneinander getrennt sind. Jeder Block enthält zwischen einer und drei Ziffern von 0 bis 9. 5 Pkt
5 min

Beispiele:

- 127.0.0.1
- 131.188.6.20
- 203.000.113.000

Hinweis: Blöcke in tatsächlichen IPv4-Adressen nehmen nur Werte zwischen 0 und 255 an, nicht jede beliebige Kombination aus drei Ziffern wie z.B. 999. Diesen Fakt ignorieren wir in dieser Aufgabe.

4 Programmieren in Python

Aufgabe 4.1 (Listen von Listen)

Schreiben Sie eine python-Funktion `longestString`, die aus einer *Liste von Listen von Strings* 10 Pkt
den längsten String findet und zurück gibt. 10 min

Beispiel: Angewendet auf die Liste `[["sloth", "elephant", "cat"], ["singing"], ["banana", "kiwi"]]`
sollte Ihre Funktion `"elephant"` zurück geben.

Diese Liste von Listen wird Ihrer Funktion als einziges Argument übergeben werden.

Sie können für diese Aufgabe annehmen, dass alle Längen der Strings in den Listen nur
einmal vorkommen (d.h. es gibt keine zwei Strings, die gleich lang sind).

Aufgabe 4.2 Schreiben Sie ein python-Programm, das das Würfeln von 10.000 sechssei- 5 Pkt
tigen Würfeln simuliert, die Augenzahlen aufaddiert und den Durchschnittswert (die Summe 5 min
geteilt durch die Anzahl Durchläufe) ausgibt.

Ihnen steht dafür eine Funktion `wuerfel()` zur Verfügung, die eine zufällige Zahl zwischen
1 und 6 (inklusive) zurück gibt.