

Name:

Geburtsdatum:

Matrikelnummer:

Nachklausur Informatische Werkzeuge in den Geistes- und Sozialwissenschaften 1

18. Juni 2020

	Nur zur Korrektur, bitte freilassen!											
Aufgabe	1.1	1.2	1.3	2.1	2.2	3.1	3.2	4.1	4.2	4.3	Summe	Note
Möglich	4	4	5	5	8	6	4	5	9	10	60	
Erreicht												

Klausurnote:

Bonuspunkte:

Endnote:

Organisatorisches

Bitte lesen die folgenden Anweisungen genau und bestätigen Sie diese mit Ihrer Unterschrift.

1. Bitte legen Sie Ihren Studentenausweis und einen Lichtbildausweis zur Personenkontrolle bereit!
2. Die angegebene Punkteverteilung gilt unter Vorbehalt.
3. Es sind keine Hilfsmittel erlaubt außer eines handgeschriebenen "Spickzettels" von 1 Seite A4 einseitig.
4. Die Lösung einer Aufgabe muss auf den vorgesehenen freien Raum auf dem Aufgabenblatt geschrieben werden; die Rückseite des Blatts kann mitverwendet werden. Wenn der Platz nicht ausreicht, können bei der Aufsicht zusätzliche Blätter angefordert werden.
5. Wenn Sie die Prüfung aus gesundheitlichen Gründen abbrechen müssen, so muss Ihre Prüfungsunfähigkeit durch eine Untersuchung in der Universitätsklinik nachgewiesen werden. Melden Sie sich in jedem Fall bei der Aufsicht und lassen Sie sich das entsprechende Formular aushändigen.
6. Die Bearbeitungszeit beträgt 60 min.
7. Sie können 60 Punkte erreichen, wenn Sie alle Aufgaben vollständig lösen. Allerdings zählen 55 Punkte bereits als volle Punktzahl, d.h. 5 Punkte sind Bonuspunkte.
8. Überprüfen Sie Ihr Exemplar der Klausur auf Vollständigkeit (17 Seiten inklusive Deckblatt und Hinweise) und einwandfreies Druckbild! **Vergessen Sie nicht, auf dem Deckblatt die Angaben zur Person einzutragen und diese Erklärung zu unterschreiben!**

Erklärung: Durch meine Unterschrift bestätige ich den Empfang der vollständigen Klausurunterlagen und die Kenntnisnahme der obigen Informationen.

Erlangen, 18. Juni 2020

.....
(Unterschrift)

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

Aufgabe 1.1 In digitalen Dokumenten, was ist der Unterschied zwischen einem *Plaintext*-Format und einem *Markup*-Format? 4 Pkt

Aufgabe 1.2 4 Pkt
Was ist in der Programmiersprache `python` der Unterschied zwischen einer *Liste* und einem *Dictionary*?

Aufgabe 1.3 Was ist ein *Dateipfad*? Wie unterscheidet sich dieser von einem *Dateinamen*? 5 Pkt
Was bedeuten in diesem Kontext *relativ* und *absolut*?

Diese Seite wurde für mehr Platz absichtlich leer gelassen.

2 Reguläre Ausdrücke

Aufgabe 2.1 (RegEx für Nummernschilder)

Autokennzeichen (Nummernschilder) in Deutschland bestehen aus einem “Unterscheidungszeichen” (ein bis drei Großbuchstaben für den Ort, an dem das Fahrzeug registriert wurde) und einer “Kennnummer” (ein bis zwei Großbuchstaben und ein bis vier Ziffern *ohne* führende Null). 5 Pkt



Unterscheidungszeichen und Kennnummer sowie Buchstaben der Kennnummer und Ziffern der Kennnummer werden entweder mit einem Doppelpunkt, einem Bindestrich oder einem Leerzeichen getrennt.

Geben Sie einen regulären Ausdruck an, der Autokennzeichen nach dem obigen Muster matcht. Es ist dabei keine Voraussetzung, dass alle Trennzeichen in einem Autokennzeichen gleich sind.

Beispiele: “KA-PA-777”, “BI KJ 4990”, “LIP Q 9” und “B:AN:1337”

Aufgabe 2.2 (RegEx mit Konsonant Vokal und Klammerung)

Geben Sie einen regulären Ausdruck an, der eine ununterbrochene Aneinanderreihung beliebiger Länge von Ausdrücken matcht, die jeweils mit einer öffnenden Klammer (also “(”, “{”, “[” oder “<”) beginnen, gefolgt von genau einem Konsonanten, genau einem Vokal und genau einer Ziffer (in dieser Reihenfolge) und mit einer schließenden Klammer (“)”, “}”, “]” oder “>”) beendet werden. 8 Pkt

Es ist keine Voraussetzung, dass die öffnende Klammer zur schließenden Klammer passen muss. Ebenfalls gibt es keine Vorgabe dafür, ob eventuelle Gruppierungen “capturing” oder “non-capturing” sein sollen.

Beispiele: "(no2)", "[na0]", "[zi5]", "[ne5>[re0]"

3 HTML und CSS

Aufgabe 3.1 (HTML-Seite mit Textfeld)

Schreiben Sie eine gültige HTML-Datei, die in einem Browser etwa so dargestellt wird wie 6 Pkt in folgendem Bild zu sehen:



Die genaue Art der Überschrift können Sie frei wählen. Der Hyperlink im zweiten Listenelement soll auf `http://jupyter.kwarc.info/` weiterleiten.

Diese Seite wurde für mehr Platz absichtlich leer gelassen.

Diese Seite wurde für mehr Platz absichtlich leer gelassen.

4 Programmieren in Python

Aufgabe 4.1 (Python-Verständnisfrage)

Was gibt das Programm in Abbildung 1 aus, wenn es ausgeführt wird?

5 Pkt

```
def isCorrectPassword(pw):  
    return pw == "friend"  
  
inputs = ["admin", "12345", "password", "friend", "swordfish"]  
counter = 0  
passed = False  
  
while counter < 3:  
    passed = isCorrectPassword(inputs[counter])  
    if passed:  
        break  
    else:  
        counter = counter + 1  
  
if passed:  
    print("ACCESS GRANTED!")  
else:  
    print("ACCESS DENIED!")
```

Abbildung 1: Was macht dieses Programm?

Diese Seite wurde für mehr Platz absichtlich leer gelassen.

Aufgabe 4.2 (Python-Programmieraufgabe File-IO)

Schreiben Sie eine python-Funktion `reverseLines`, die einen Dateipfad als Argument entgegen nimmt, diese Datei mit `open(filename, mode)` öffnet, zeilenweise ausliest und in einer zweiten Datei namens `reversed.txt` alle Zeilen aus der ersten Datei in umgekehrter Reihenfolge abspeichert. 9 Pkt

Diese Seite wurde für mehr Platz absichtlich leer gelassen.

Aufgabe 4.3 (Python-Programmieraufgabe RegEx)

Schreiben Sie eine python-Funktion `deleteByRegex`, die einen regulären Ausdruck (also einen String) als Argument entgegen nimmt und alle Dateien im aktuellen Verzeichnis (".") löscht, deren Dateiname von dem regulären Ausdruck gematcht wird. 10 Pkt

So sollte zum Beispiel nach dem Aufruf von `deleteByRegex("\\d+\\.txt")` eine Datei namens `31415.txt` gelöscht werden, eine Datei namens `survives.txt` allerdings nicht.

Hinweis: Ihnen könnten hierzu die Bibliotheksfunktionen `os.listdir(directory)`, `os.remove(filename)` und `re.fullmatch(regex,string)` nützlich erscheinen.

Denken Sie auch an alle benötigten **import**-Statements.

Diese Seite wurde für mehr Platz absichtlich leer gelassen.

Diese Seite wurde für mehr Platz absichtlich leer gelassen.

Diese Seite wurde für mehr Platz absichtlich leer gelassen.

Diese Seite wurde für mehr Platz absichtlich leer gelassen.