

Name:

Vorname:

Geburtsdatum:

Matrikelnummer:

Platz:

Nachhol-Klausur
Informatische Werkzeuge in den
Geistes- und Sozialwissenschaften 1

21. April 2022

Nur zur Korrektur, bitte freilassen!												
prob.	1.1	1.2	1.3	2.1	2.2	3.1	3.2	4.1	4.2	4.3	Summe	Note
total	4	4	6	4	4	6	6	4	12	10	60	
reached												

Klausurnote:

Bonuspunkte:

Endnote:

Organisatorisches

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

1. Bitte legen Sie Ihren Studierendenausweis und einen Lichtbildausweis zur Personenkontrolle bereit!
2. Die angegebene Punkteverteilung gilt unter Vorbehalt.
3. 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.
4. Es sind keine Hilfsmittel erlaubt außer eines handgeschriebenen "Spickzettels" von 1 Seite A4 einseitig.
5. Die Bearbeitungszeit beträgt genau 60 min.
6. Schreiben Sie die Lösungen direkt auf die ausgeteilten Aufgabenblätter. Andere Blätter werden nicht bewertet.
7. 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.
8. Überprüfen Sie Ihr Exemplar der Klausur auf Vollständigkeit (16 Seiten exklusive 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, 21. April 2022

.....
(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 und Verständnis

Aufgabe 1.1 (Dateipfade)

4 Pkt

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

Aufgabe 1.2 (Server und Clients)

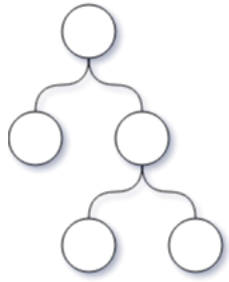
4 Pkt

Beschreiben Sie kurz die Rollen von „Servern“ und „Clients“ im World Wide Web
und wie diese typischerweise miteinander interagieren und zu welchem Zweck.

Aufgabe 1.3 (Bäume)

6 Pkt

Welche der folgenden Graphen zeigen korrekte Bäume im mathematischen Sinne? Für nicht-Bäume begründen Sie bitte Ihre Entscheidung kurz (!).



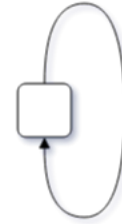
A



B



C



D

2 Reguläre Ausdrücke

Aufgabe 2.1 (IBANs)

4 Pkt

In dieser Aufgabe geht es um einen regulären Ausdruck (Regex) für korrekte International Bank Account Numbers (IBANs). Die genaue Form von IBANs ist abhängig vom Land, wir suchen nach einem regulären Ausdruck für korrekte IBANs aus allen DACH-Ländern (d.h. Deutschland, Österreich und Schweiz).

Deutsche IBANs sind die Buchstaben "DE" gefolgt von 20 Ziffern, Österreichische IBANs sind die Buchstaben "AT" gefolgt von 18 Ziffern und Schweizer IBANs sind die Buchstaben "CH" gefolgt von 19 Ziffern.

Beispiele:

- DE89370400440532013000 (Deutschland)
- AT611904300234573201 (Österreich)
- CH9300762011623852957 (Schweiz)

Geben Sie einen regulären Ausdruck an, der genau alle DACH-IBANs erfasst.

Aufgabe 2.2 (Regulärer Ausdruck für Cryptocurrency Wallets)

4 Pkt

Als Teil einer Ermittlung gegen Geldwäsche werden Sie damit beauftragt, in einem großen Textkorpus alle potentiellen Wallet-Adressen für die Cryptowährung „PollutionCoin“ zu finden. So eine Adresse besteht für unsere Zwecke aus zwei Teilen (in dieser Reihenfolge), und zwar...

- ...dem String "1", "3" oder "pc1"
- ...zwischen 27 und 34 (inklusive) Kleinbuchstaben, Großbuchstaben oder Ziffern. Dabei kommen die Buchstaben I, l, O und die Ziffer 0 nicht vor, um Verwechslungen auszuschließen.

Hier sehen Sie ein paar Beispiele für diese Adressen:

```
12cbQLTFMXRnSzktFkuoG3eHoMeFtpTu3S
1FeexV6bAHb8ybZjqQMjJrcCrHGW9sb6uF
3FfsbQntU2BAkVy2xaaRpSmLEDMEUwPbG1
pc1eHhgW6vquBYhwMPHQ668PjxTtpvZGPC
```

Geben Sie einen regulären Ausdruck an, der auf Wallet-Adressen wie oben beschrieben matcht. Falls Sie Gruppierungen verwenden, ist es egal ob diese *capturing* oder *non-capturing* sind.

3 Digitale Dokumente

Aufgabe 3.1 (Mentales CSS)

6 Pkt

Gegeben ist der Quelltext der folgenden **HTML**-Seite:

```
<!DOCTYPE html>
<html>

  <head>
    <style>
      p      {text-align: center;}
      body   {background-color: yellow;}
      h1     {font-size: 100pt;}
      span   {color: red;}
      div    {background-color: green;}
    </style>
  </head>

  <body>

    <h1 style="font-size: 5pt;">Headline</h1>

    <p>Lorem ipsum dolor sit amet...</p>

    <div>
      Lorem ipsum dolor sit amet...

      <form>
        <input type="submit" value="Useless_button!">
      </form>

    </div>

  </body>
</html>
```

Beschreiben Sie, welche Elemente auf der fertig dargestellten **HTML**-Seite zu sehen sind, wie sie angeordnet sind und welche Farbe sie haben.

Diese Seite wurde für mehr Platz absichtlich leer gelassen.

Aufgabe 3.2 (XPath)

6 Pkt

1. Im Kontext von XML-Dokumenten, was versteht man unter einem *XPath*?
2. Beschreiben Sie außerdem für jeden der unten angegebenen XPaths, was genau damit erreicht werden kann.
 - (a) `//*[not(*)]`
 - (b) `//*[self::subject or self::note][2]/text()`

4 Programmieren in Python

Aufgabe 4.1 (Python-Verständnisfrage)

4 Pkt

Gibt das folgende Programm etwas aus, wenn es ausgeführt wird? Oder wird die Schleife niemals gebrochen (und folglich nichts ausgegeben)? Begründen Sie Ihre Antwort.

```
einnahmen = 1001
ausgaben = 1000
angestellte = 1

quotient = einnahmen / ausgaben
profit = 0

while True:

    for _ in range(angestellte):
        quotient = quotient * quotient

    einnahmen = einnahmen * quotient
    ausgaben = ausgaben * (quotient / quotient)

    profit = einnahmen - ausgaben

    if profit > 0:
        angestellte += 1
    else:
        angestellte -= 1

    if profit > 1000000:
        break

print("Ziel_erreicht!")
```

Diese Seite wurde für mehr Platz absichtlich leer gelassen.

Aufgabe 4.2 (Lotto)

12 Pkt

Beatrices Großvater hat mitbekommen, dass sie jetzt „Computer“ studiert und bittet sie um Hilfe beim Ausfüllen seines Lottoscheins. Leider lässt er sich nicht von ihren mathematischen Argumenten überzeugen, dass Lotterien wie diese statistisch identisch dazu sind, das Geld gleich zum Fenster hinaus zu werfen.

Basierend auf seinem Geburtsdatum (04.09.1947) und dem ihrer Großmutter (29.02.1948) möchte er die Zahlen 4,9,47,29,2,48 spielen. Er möchte allerdings wissen, ob diese Zahlen in genau dieser Kombination schon einmal gezogen wurden, und welche Superzahl am häufigsten gezogen wird.

Ihnen liegt eine CSV-Datei `lotto_archiv.csv` mit vergangenen Ergebnissen vor, die wie folgt formatiert ist:

```
KW;Datum;Gewinnzahlen;Zusatzzahl;Superzahl;Tag
1;01.01.2000;1,2,12,30,40,47;41;0;Samstag
2;08.01.2000;4,6,20,29,31,37;33;8;Samstag
3;15.01.2000;22,31,32,33,35,38;14;8;Samstag
4;22.01.2000;10,15,25,28,30,40;32;7;Samstag
5;29.01.2000;19,21,30,34,41,45;16;0;Samstag
[...]
```

Schreiben Sie ein `python`-Programm, das ausgibt, ob genau diese sechs Zahlen, die Beatrices Großvater spielen möchte, schon einmal so gezogen wurden. Wenn ja, dann soll auch das entsprechende Datum ausgegeben werden. Das Programm soll außerdem überprüfen (und ausgeben), welche Superzahl in der Datengrundlage am häufigsten vorkommt.

Hinweis:

- Die Reihenfolge der gezogenen Zahlen spielt bei dieser Lotterie für den Gewinn keine Rolle, für Ihre Abfrage allerdings schon.
 - Sollten die Zahlen mehrmals vorgekommen sein oder mehrere Superzahlen gleich häufig erschienen sein, ist es egal welches Datum von den passenden Ziehungen / welche der häufigsten Superzahlen Sie ausgeben.
 - Um einen String an einem Trennzeichen zu teilen, steht Ihnen die Funktion `exampleString.split(trennzeichen)` zur Verfügung.
Beispiel: `"3-1-4-1-5".split("-") == ["3", "1", "4", "1", "5"]`
-

Diese Seite wurde für mehr Platz absichtlich leer gelassen.

Aufgabe 4.3 (Python-Programmieraufgabe bottle / Liste)

10 Pkt

Geben Sie den python-Quellcode für eine `bottle`-Route an, die auf dem Pfad

`/echo/...`

ansteuerbar ist und zwei Parameter entgegen nimmt..

Die Route soll eine valide HTML-Seite zurückgeben, deren `<body>`-Tag lediglich eine nicht nummerierte Liste enthält. Diese Liste soll so viele Elemente haben, wie der zweite Parameter angibt. Diese Elemente sollen jeweils den ersten Parameter als Text enthalten.

Dabei soll das erste Element eine Schriftgröße haben, die dem zweiten Parameter gleicht, das zweite eine Schriftgröße kleiner und so weiter bis runter auf 1.

Beispiel: `/echo/Hallo/15/` würde also eine HTML-Seite mit einer Liste zeigen, die 15 Elemente mit dem Text „Hallo“ hat, in den Schriftgrößen 15, 14, 13, ..., 3, 2 und 1.

Hinweis: Zur Erinnerung: Sie können CSS-Regeln für einzelne HTML Knoten wie folgt über das `style`-Attribut des Knotens setzen:

```
<div style="color:red;">Here's some red text in a div!</div>
```

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.