Name: Vorname:

Geburtsdatum: Matrikelnummer:

Platz:

Klausur
Informatische Werkzeuge in den Geistes- und Sozialwissenschaften 1

16. Februar 2023

	To be used for grading, do not write here										
prob.	1.1	1.2	1.3	2.1	2.2	3.1	3.2	4.1	4.2	Sum	grade
total	4	4	4	6	8	8	10	6	10	60	
reached											

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 (11 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, 16. Februar 2023 .....

(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 (Stellenwertsysteme) 4 Pkt

Menschen benutzen verschiedene Zahlensysteme. Manche davon sind *Stellenwertsysteme* („Positional Number Systems“). Erklären sie kurz (!), was ein Stellenwertsystem ausmacht. Nennen Sie außerdem zwei Zahlensysteme, die Stellenwertsysteme sind und ein Zahlensystem, welches keins ist.

Aufgabe 1.2 (URI-Encoding) 4 Pkt

Erklären Sie kurz (!), was beim *URI-Encoding* vor sich geht. Was genau wird encodiert? Warum/wann ist das nötig?

Aufgabe 1.3 (Bäume) 4 Pkt

Was ist ein *Baum* im Kontext von Informatik? Welche Bestandteile haben Bäume? Erklären Sie außerdem kurz die Begriffe *Wurzel* und *Blatt*.

2 Reguläre Ausdrücke

Aufgabe 2.1 (Regulärer Ausdruck für Uhrzeit) 6 Pkt

Geben Sie einen regulären Ausdruck an, der auf textuelle Angaben der Uhrzeit im Format `hh:mm:ss` (24 Stunden-Format mit Minuten und Sekunden) matcht.

Hier sind ein paar Uhrzeiten im korrekten Format:

```
09:30:59
13:12:00
22:49:22
```

Diese hier hingegen sind inkorrekt und sollen *nicht* gematcht werden.

```
12:04:11 AM
32:80:99
4:1:7
```

Falls Sie Gruppierungen verwenden, ist es egal ob diese *capturing* oder *non-capturing* sind.

Aufgabe 2.2 (Regulärer Ausdruck für DOIs) 8 Pkt

Um auf digitale Objekte (wie Journals, Berichte, Paper, Artikel, …) verlässlich verweisen zu können, wurde von der International Organization for Standardization (ISO) im Jahr 2012 ein Standard für *Digital Object Identifiers* (**DOI**) veröffentlicht.

- Ein **DOI** besteht aus einem Präfix und einem Suffix, die von einem „/“ getrennt werden.
- Das Präfix beginnt (für unsere Zwecke) immer mit der Zeichenkette „`10.`“ gefolgt von einer ganzen Zahl ≥ 1000.
- Das Suffix ist beliebig und unterliegt keinerlei Einschränkungen. In der Praxis ist es oft numerisch, der Standard erlaubt allerdings jede (nichtleere) Aneinanderreihung von Zeichen.

In normalen Texten ohne Hyperlinks wird ein **DOI** oft mit der Zeichenkette „`doi:`“ eingeführt, wie auch in den folgenden Beispielen:

```
doi:10.1000/182
doi:10.20347/WIAS.PREPRINT.2431
doi:10.1016/j.disc.2012.08.018
doi:10.1007/978-3-662-44199-2_5
doi:10.1007/s10817-012-9271-4
```

Geben Sie einen regulären Ausdruck an, der auf textuelle **DOI**-Ausdrücke wie oben beschrieben (also inklusive „`doi:`“) matcht. Falls Sie Gruppierungen verwenden, ist es egal ob diese *capturing* oder *non-capturing* sind.

3 Digitale Dokumente

Aufgabe 3.1 (XPath) 8 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) `//*[@price<42]`
   (b) `//img[@alt='Logo']`

Aufgabe 3.2 (HTML Seite mit CSS) 10 Pkt

Schreiben Sie den Quellcode für eine gültige *HTML*-Seite.

1. Diese Seite sollte im `body`-Teil mindestens drei *unterschiedliche* Elemente zeigen, mit denen Nutzer*Innen in irgendeiner Form interagieren können.

2. Die Seite sollte weiterhin im `head`-Bereich in angemessener Umgebung mindestens drei CSS-Regeln beinhalten. Diese Regeln sollen die Elemente im `body` visuell verändern. Achten Sie bitte dabei darauf, dass sich keine Regelelemente doppeln. Gegebene Selektoren und veränderte Attribute sollten also höchstens einmal auftauchen. Ästhetische Bedenken spielen in dieser Aufgabe *keine* Rolle.

3. Zuletzt soll außerdem irgendwo im Quellcode ein gültiger HTML-Kommentar vorkommen.

This page was intentionally left blank for extra space

4 Programmieren in Python

Aufgabe 4.1 (Python-Verständnisfrage) 6 Pkt

Machen Sie sich mit den in Abbildung 1 definierten Funktionen `a(s1,s2)` und `m(f1,f2)` vertraut. Welche gängigen Funktionen sind hier verschleiert implementiert worden? Was ist der Rückgabewert von `m(4,5)`?

```
# Slightly obfuscated code
def a(s1, s2):
    x = 0

    for _ in range(s1):
        x = x + 1

    for _ in range(s2):
        x = x + 1

    return x

def m(f1, f2):
    y = 0
    z = 0

    while z < f1:
        y = a(y, f2)
        z = z + 1

    return y
```

?figurename? 1: Welche bekannten Funktionen sind hier zu sehen?

Aufgabe 4.2 (Bottle Route für Reguläre Ausdrücke) 10 Pkt

Schreiben Sie eine Route `regex()` für einen `bottle`-Server, die zwei Argumente (Strings) über die URL annimmt: einen regulären Ausdruck und einen Text. Es soll getestet werden ob und wenn ja wie der RegEx auf den Text matcht.

Die Route soll konkret folgendes Verhalten zeigen:

- Matcht der RegEx an keiner Stelle im String, so soll `‘‘Kein Match!’’` zurück gegeben werden.
- Matcht der RegEx den gesamten String von Anfang bis Ende, so soll `‘‘Komplettes Match!’’` zurück gegeben werden.
- Matcht der RegEx nur partiell, so soll angegeben werden, an wie vielen Stellen dies passiert (z.B. `‘‘Genau 4 Matches!’’`). Nur die Anzahl ist wichtig, nichts weiter.

Sie können für diese Aufgabe davon ausgehen, dass alle benötigten `import`-Statements bereits vorhanden sind und das alle Strings wie getippt an die Route weiter gegeben werden.

This page was intentionally left blank for extra space

This page was intentionally left blank for extra space

This page was intentionally left blank for extra space