

Name:

Geburtsdatum:

Matrikelnummer:

Nachklausur Informatische Werkzeuge in den Geistes- und Sozialwissenschaften 2

29. Oktober 2020

	Nur zur Korrektur, bitte freilassen!											
Aufgabe	1.1	1.2	2.1	2.2	2.3	3.1	4.1	5.1	6.1	6.2	Summe	Note
Möglich	4	4	4	3	2	10	10	4	6	8	55	
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 55 Punkte erreichen, wenn Sie alle Aufgaben vollständig lösen. Allerdings zählen 50 Punkte bereits als volle Punktzahl, d.h. 5 Punkte sind Bonuspunkte.
8. Überprüfen Sie Ihr Exemplar der Klausur auf Vollständigkeit (19 Seiten inklusive Deckblatt und Hinweise) und einwandfreies Druckbild! **Ver-gessen 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, 29. Oktober 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 Versionskontrolle

Aufgabe 1.1 (Versionskontrolle via Kommandozeile)

Beatrice Beispiel hat ein `git`-Repository geklont und arbeitet jetzt an `DokumentA.py`, welches in dem Repository verwaltet wird. Nachdem sie mit ihren Änderungen zufrieden ist, möchte sie die Änderungen veröffentlichen, sodass andere Nutzer des Repositorys sie auch sehen können. 4 Pkt

In der Zwischenzeit hat ein Kollege von Beatrice Änderungen *an einer anderen Datei* durchgeführt und veröffentlicht.

Geben Sie alle Schritte an, die Beatrice in der Kommandozeile durchführen muss, um ihre Änderungen zu veröffentlichen.

Aufgabe 1.2 (Bilder in Versionskontrolle)

Warum ist `git` nicht gut geeignet für Binärdateien, wie zum Beispiel für Bilder?

4 Pkt

2 Datenbanken

Aufgabe 2.1 (Snack-Datenbank: Tabelle erstellen)

4 Pkt

Gegeben ist folgende Tabelle „Snacks“:

Name	Hersteller	Gründungsjahr	Kilokalorien
Venus	Venus Inc.	1925	230
N&Ns	Venus Inc.	1925	140
Stay's	Harret Food Company	1912	160

Die Spalte „Gründungsjahr“ beschreibt dabei das Jahr, in dem die Herstellerfirma gegründet wurde.

Geben Sie die SQL-Befehle zum Erstellen einer Tabelle unter dem Namen **Snacks** mit der obigen Struktur an. Sie müssen keine Einträge hinzufügen. Vermeiden Sie dabei eventuelle Probleme, falls bereits eine Tabelle namens **Snacks** existiert.

Aufgabe 2.2 (Snack-Datenbank: Datenbank Werteingabe)

Ihr Kollege hat Probleme beim Einfügen der dritten Zeile in die Datenbank. Er versucht 3 Pkt
es mit folgendem Befehl:

```
INSERT INTO Snacks VALUES('Stay's', 'Harret Food Company', 1912, 160);
```

Wo ist das Problem? Geben Sie auch den korrekten Befehl an.

Aufgabe 2.3 (Snack-Datenbank: Datenbank-Update)

Die Kilokalorien-Angaben verstehen sich pro Packung. Venus Inc. vergrößert nun die Packung ihrer N&Ns und damit hat eine Packung 180 kcal. Geben Sie den **SQL**-Befehl an, der den Eintrag in der Datenbank-Tabelle entsprechend ändert. 2 Pkt

Diese Seite wurde für mehr Platz absichtlich leer gelassen.

3 Bild

Aufgabe 3.1 (Gedrehte Graustufen)

Schreiben Sie eine python-Funktion `fused_grey_rotate(img)` die ein Bild `img` als Parameter 10 Pkt nimmt und eine veränderte Kopie des Bildes zurück gibt, die sowohl um 90° gedreht als auch in Graustufen übersetzt wurde.

Dabei wird verlangt, dass beide Operationen “auf einmal” durchgeführt werden. Erstellen Sie also nicht erst eine rotierte Kopie welche Sie dann in Graustufen umwandeln. Verwenden Sie also auch *nicht* die vorgefertigten Funktionen im Pillow-Modul `ImageOps`. Folgendes Codegerüst ist gegeben:

```
from PIL import Image

def rotate90(x, y, originalWidth, originalHeight) :
    # Implementierungsdetails ausgelassen. Sie muessen diese Funktion nicht implementieren!
    return (rotated_x, rotated_y)

def fused_grey_rotate(img) :
    originalWidth = img.width
    originalHeight = img.height
    newWidth = # TODO: Ergaenzen!
    newHeight = # TODO: Ergaenzen!

    # Neues Bild erstellen. "L" bedeutet Graustufe.
    result = Image.new("L", (newWidth, newHeight), 0)

    # Ihr Code hier...

    return result
```

Wie Sie sehen steht Ihnen eine Hilfsfunktion `rotate90` zur Verfügung. Sie müssen die Rotation also nicht selber implementieren. Die Funktion gibt ein Tupel der Form (x, y) zurück, welches für die Eingabewerte passende um 90° rotierte Koordinaten zurückgibt.

Hinweis: Ein paar nützliche Informationen finden Sie im Folgenden:

- Schreiben Sie eine Schleife über das *Eingabebild*, ermitteln Sie dort die Farbe pro Pixel und schreiben Sie das Ergebnis an die rotierten Koordinaten im *Ergebnisbild*.
- Sowohl der Parameter `img` als auch die Rückgabe Ihrer Funktion sind Pillow-Bilder.
- Das Bild `result`, welches Sie zurückgeben, ist in Graustufen, hat also nur einen Kanal.
- Sie können den Wert eines Pixels wie folgt auslesen:
`img.getpixel((x, y))` gibt ein 3-Tupel (r, g, b) zurück.
- Sie können den Wert eines Pixels wie folgt setzen:
`img.putpixel((x, y), value)`
- Die Umwandlung von R,G,B zu Graustufen funktioniert nach folgender Formel:
 $grey = 0.21 \cdot R + 0.71 \cdot G + 0.08 \cdot B$

Diese Seite wurde für mehr Platz absichtlich leer gelassen.

4 Web-Applikationen

Aufgabe 4.1 (Bottle Route)

Schreiben Sie eine Route `roll()` für einen `bottle`-Server, die keine Argumente nimmt, dafür jedoch einen HTTP GET-Request verarbeitet und eine Zeile HTML zurück gibt (also anzeigt). In diesem GET-Request werden drei Argumente übergeben (`lowerBound`, `upperBound` und `title`). 10 Pkt

Die Argumente `lowerBound` und `upperBound` sind ganze Zahlen. Ihre Route soll eine zufällige Zahl zwischen `lowerBound` und `upperBound` generieren. Dabei soll das Intervall nach oben hin exklusiv sein. Das bedeutet, dass der höchste erlaubte Wert von `upperBound` die Zahl -1 ist.

Beispiel: Für `lowerBound = 3` und `upperBound = 7` sind 3, 4, 5 und 6 möglich.

Die Route soll dann einen String zurückgeben. Der String soll folgendes Muster haben:

- Wenn `lowerBound` kleiner als `upperBound` ist, soll der übergebene `title` fett gedruckt (`...`) werden. Darauf folgt ein Doppelpunkt und dann die generierte Zufallszahl (nicht mehr fett gedruckt!).
- Wenn `title` ein leerer String ist, soll nur die Zufallszahl zurückgegeben werden.
- Wenn `lowerBound` größer oder gleich `upperBound` ist, soll „Falsche Eingabe“ zurückgegeben werden.

Zum Erzeugen von Zufallszahlen können Sie das `random`-Modul verwenden. Sie können davon ausgehen, dass dieses schon importiert ist. Eine Zufallszahl erstellen Sie mit der Funktion `random.randint(start, stop)`, welche eine zufällige Ganzzahl zurückgibt. Das Intervall ist hier inklusiv, also sowohl `start` als auch `stop` sind mögliche Rückgabewerte!

Hinweis: Liefert ein GET-Request das Argument `attribute` können Sie dieses mit dem Statement `request.query.attribute` auslesen. Dies funktioniert selbstverständlich auch mit jedem anderen Argumentnamen.

Diese Seite wurde für mehr Platz absichtlich leer gelassen.

5 Geistiges Eigentum

Aufgabe 5.1 (Public Domain)

4 Pkt

1. Wann wird davon gesprochen, dass ein Werk in der “Public Domain” liegt?
2. Nennen Sie eine Nutzung eines Werkes, die nicht erlaubt wäre, wenn das Werk mit der GNU General Public License (GPL) lizenziert wäre statt in der Public Domain zu liegen.

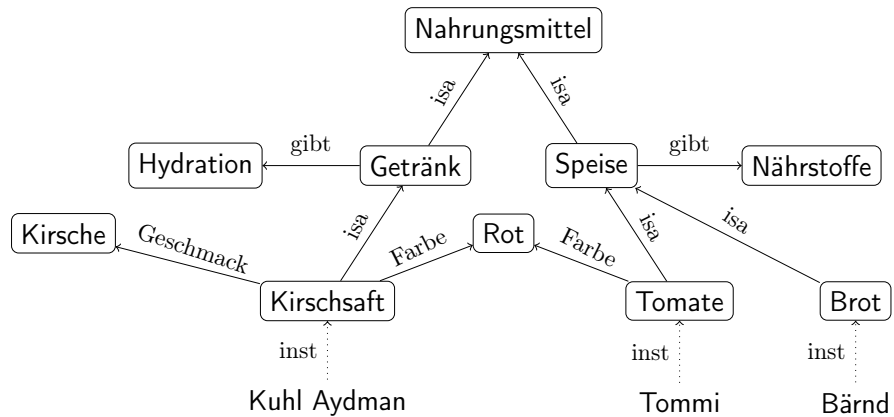
Diese Seite wurde für mehr Platz absichtlich leer gelassen.

6 Semantic Web & Kulturelles Erbe

Aufgabe 6.1 (Semantische Netze zu Nahrungsmitteln)

Gegeben sei das folgende semantische Netz:

6 Pkt



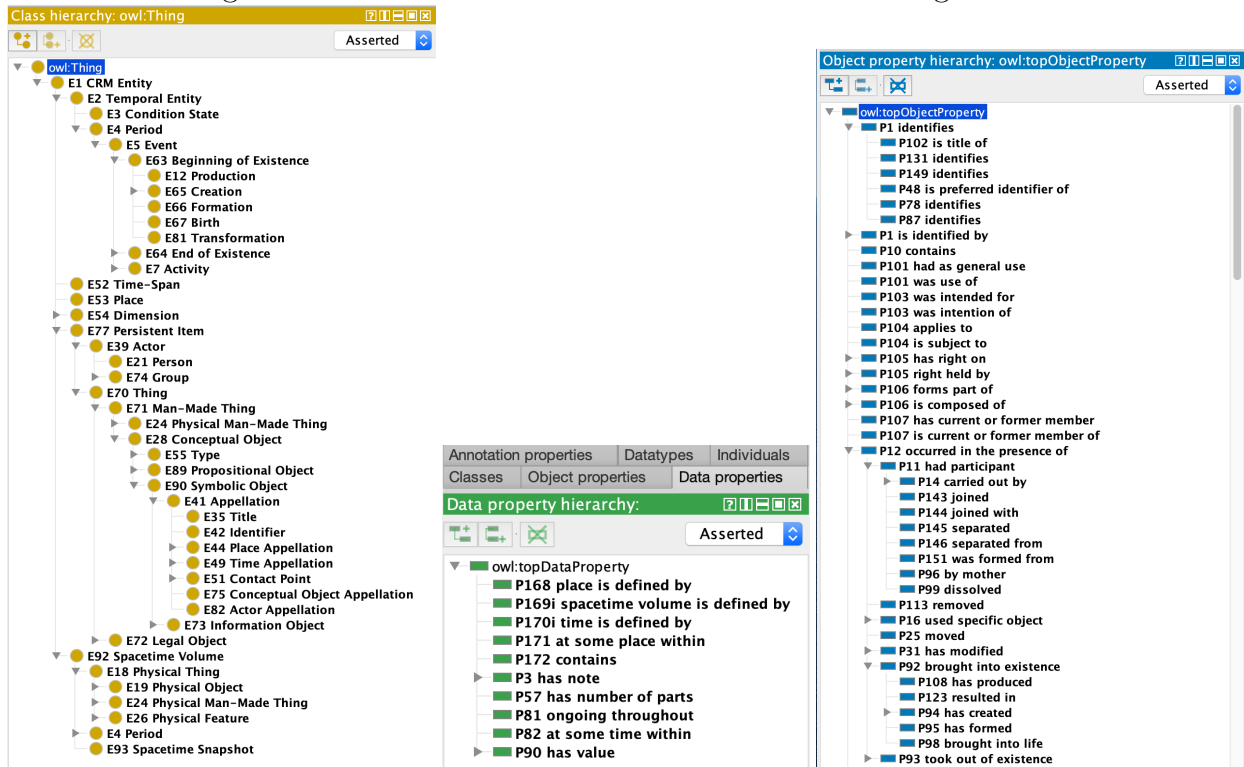
1. Geben Sie je drei Subjekt/Prädikat/Objekt-Tripel aus der TBox und der ABox dieses Netzes an.
2. Zeichnen Sie jeweils drei Fakten und terminologische Axiome in das Netz ein, die im Netz inferiert werden können aber nicht explizit repräsentiert sind.

Diese Seite wurde für mehr Platz absichtlich leer gelassen.

Aufgabe 6.2 (Modellierung in CIDOC-CRM)

8 Pkt

dellieren Sie in CIDOC CRM die Information “*Im Jahr 1618 hat Pieter Brueghel das Bild “St. Joris” gemalt*”. Zeichnen Sie das Netzwerk aus CIDOC CRM Klassen und Relationen. Zur Orientierung drucken bilden wir Teile der CIDOC CRM Ontologie unten ab.



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.