

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:

Wir stellen die „Lösungen“ der Aufgaben in diesem Dokument den Studierenden als Ausgangspunkt zu Beantwortung von Fragen zur Verfügung. Obwohl wir uns bemühen, hilfreiche „Lösungen“ bereitzustellen, können diese doch unvollständig sein oder sogar Fehler enthalten.

Sollten Sie „Lösungen“ finden, die Sie nicht verstehen oder für fehlerhaft halten, sollten sie diese auf dem Kursforum oder mit ihren Tutoren diskutieren und die Referenten informieren.

Auf jeden Fall ist die Korrektur von Hausaufgaben oder Klausuren niemals nur ein Prozess in dem einfach die Antworten der Studierenden „mit der Musterlösung verglichen“ werden.

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
4 min

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.

Lösung: Hier sind alle benötigten Befehle.

```
git add DokumentA.py
git commit -m "Changed DokumentA"
git pull # sonst ist die Basis-Version veraltet
git push # Veroeffentlichung
```

Aufgabe 1.2 (Bilder in Versionskontrolle)

Warum ist `git` nicht gut geeignet für Binärdateien, wie zum Beispiel für Bilder? 4 Pkt

Lösung: `git`-Commits basieren immer auf dem vorherigen Commit. Dabei werden nur Änderungen zu diesem anderen Commit gespeichert, also normalerweise nur ein Teil des Dokuments. `git` findet diese Änderungen dabei Zeile für Zeile im Dokument. 4 min

Bei Bildern (und anderen Binärdateien) gibt es dieses Konzept von Zeilen nicht. Stattdessen wird bei einer Änderung immer das komplette Bild in den Commit aufgenommen. Da in der lokalen Kopie eines Repositoriums immer die komplette History heruntergeladen wird, wird diese dann schnell sehr groß.

Dazu kommt, dass `git` im Fall von Merge-Konflikten Zeichenketten in die Dateien einfügt. Diese ergeben in Binärdateien natürlich keinen Sinn und können damit das Bild zerstören.

2 Datenbanken

Aufgabe 2.1 (Snack-Datenbank: Tabelle erstellen)

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. 4 Pkt
4 min

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.

Lösung:Die Befehle sind wie folgt:

```
DROP TABLE IF EXISTS Snacks;
```

```
CREATE TABLE Snacks(  
  Name TEXT,  
  Hersteller TEXT,  
  Gruendungsjahr INTEGER,  
  Kilokalorien INTEGER);
```

Aufgabe 2.2 (Snack-Datenbank: Datenbank Werteingabe)

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

3 Pkt

3 min

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

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

Lösung:Das Problem ist, dass im Wert **Stay's** ein Apostroph vorkommt, welches als String-Delimiter interpretiert wird. Ihr Kollege sollte damit die einzufügenden Strings mit doppelten Anführungsstrichen markieren, also:

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

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

2 min

Lösung:

```
UPDATE Snacks SET Kilokalorien=180 WHERE Name="N&Ns";
```

3 Bild

Aufgabe 3.1 (Gedrehte Graustufen)

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

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:
 $grey = 0.21 \cdot R + 0.71 \cdot G + 0.08 \cdot B$

Lösung: Hier ist eine mögliche Implementation:

```
from PIL import Image

def rotate90(x, y, originalWidth, originalHeight) :
    rotated_x = y
    rotated_y = -x + originalWidth - 1
    return (rotated_x, rotated_y)

def fused_grey_rotate(img) :
    originalWidth = img.width
    originalHeight = img.height
    newWidth = originalHeight
    newHeight = originalWidth

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

    # Ihr Code hier...
    for y in range(0, originalHeight) :
        for x in range(0, originalWidth) :

            r, g, b = img.getpixel((x, y)) # Farbe im Inputbild.
            grey = int(0.21 * r + 0.71 * g + 0.08 * b) #

            rotatedCoordinates = rotate90(x, y, originalWidth, originalHeight) # Ergebnis ist ein Tupel.

            result.putpixel(rotatedCoordinates, grey)

    return result

img = Image.open("img.png")
rotated = fused_grey_rotate(img)
img.show()
rotated.show()
```

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
10 min

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.

Lösung:Eine mögliche Lösung ist die folgende:

```
@get('/roll')
def roll() :
    lowerBound = int(request.query.lowerBound)
    upperBound = int(request.query.upperBound)
    title = request.query.title

    # Wenn lowerBound >= upperBound ist, wird eine Fehlermeldung zurueckgegeben.
    if lowerBound >= upperBound :
        return "Falsche Eingabe"

    # Wenn title leer ist, ist firstPart leer. Ansonsten wird firstPart der fett gedruckte title und ein Doppelpunkt
    firstPart = ""
    if title != "" :
        firstPart = "<b>" + title + ":</b> "

    # Intervall ist oben exklusiv, also -1
    randomNumber = random.randint(lowerBound, upperBound - 1)

    return firstPart + string(randomNumber)
```

5 Geistiges Eigentum

Aufgabe 5.1 (Public Domain)

4 Pkt

4 min

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.

Lösung:

1. Ein Werk liegt in der Public Domain, wenn es keinerlei anderem Copyright unterliegt. Beispielsweise liegen viele Veröffentlichungen, die von US-Regierungsangestellten auf der Arbeit erstellt werden, in der Public Domain. Alte Werke (z.B. die Werke von Lovecraft) gehen in die Public Domain über, wenn das bestehende Copyright ausläuft.
 2. Es wäre nicht erlaubt, ein GPL-lizenziertes Stück Software verändert weiterzuverbreiten, ohne selbst eine kompatible Lizenz zu wählen und den veränderten Quellcode zugänglich zu machen.
-

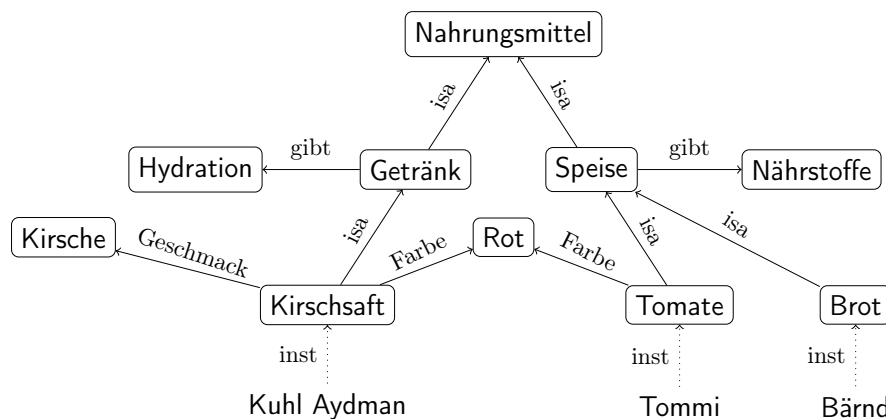
6 Semantic Web & Kulturelles Erbe

Aufgabe 6.1 (Semantische Netze zu Nahrungsmitteln)

Gegeben sei das folgende semantische Netz:

6 Pkt

6 min



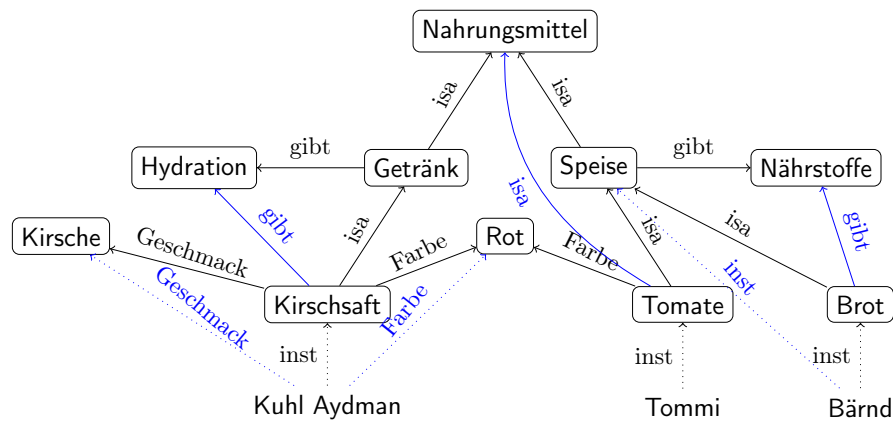
1. Geben Sie je drei Subjekt/Prädikat/Objekt-Tripel aus der TBox und der ABox dieses Netzes an.
2. Zeichnen Sie jeweils drei Fakten und terminologische Axiome in das Netz ein, die im Netz inferiert werden können aber nicht explizit repräsentiert sind.

Lösung:

1. Subjekt/Prädikat/Objekt

TBox	ABox
tomate/farbe/rot	tommy/farbe/rot
kirschsaft/geschmack/kirsche	kuhl aydman/inst/kirschsaft
brot/ist_ein/speise	bärnd/gibt/nährstoffe

2. Wir zeichnen die neuen Relationen in blau ein:



Aufgabe 6.2 (Modellierung in CIDOC-CRM)

8 Pkt

modellieren 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.

8 min

The image displays three OWL ontologies:

- Class hierarchy: owl:Thing**: A tree structure showing classes like E1 CRM Entity, E2 Temporal Entity, E3 Condition State, E4 Period, E5 Event, E63 Beginning of Existence, E12 Production, E65 Creation, E66 Formation, E67 Birth, E81 Transformation, E64 End of Existence, E7 Activity, E52 Time-Span, E53 Place, E54 Dimension, E77 Persistent Item, E39 Actor, E21 Person, E74 Group, E70 Thing, E71 Man-Made Thing, E24 Physical Man-Made Thing, E28 Conceptual Object, E55 Type, E89 Propositional Object, E90 Symbolic Object, E41 Appellation, E35 Title, E42 Identifier, E44 Place Appellation, E49 Time Appellation, E51 Contact Point, E75 Conceptual Object Appellation, E82 Actor Appellation, E73 Information Object, E72 Legal Object, E92 Spacetime Volume, E18 Physical Thing, E19 Physical Object, E24 Physical Man-Made Thing, E26 Physical Feature, E4 Period, and E93 Spacetime Snapshot.
- Data property hierarchy: owl:topDataProperty**: A tree structure showing data properties like P168 place is defined by, P169i spacetime volume is defined by, P170i time is defined by, P171 at some place within, P172 contains, P3 has note, P57 has number of parts, P81 ongoing throughout, P82 at some time within, and P90 has value.
- Object property hierarchy: owl:topObjectProperty**: A tree structure showing object properties like P1 identifies, P102 is title of, P131 identifies, P149 identifies, P48 is preferred identifier of, P78 identifies, P87 identifies, P1 is identified by, P10 contains, P101 had as general use, P101 was use of, P103 was intended for, P103 was intention of, P104 applies to, P104 is subject to, P105 has right on, P105 right held by, P106 forms part of, P106 is composed of, P107 has current or former member, P107 is current or former member of, P12 occurred in the presence of, P11 had participant, P14 carried out by, P143 joined, P144 joined with, P145 separated, P146 separated from, P151 was formed from, P96 by mother, P99 dissolved, P113 removed, P16 used specific object, P25 moved, P31 has modified, P92 brought into existence, P108 has produced, P123 resulted in, P94 has created, P95 has formed, P98 brought into life, and P93 took out of existence.

Lösung: Das Objekt um das es geht hat die Benennung "St. Joris" das gibt den oberen Ast im Diagramm unten. Ausserdem wurde das Objekt in einen Produktionsereignis geschaffen, das von einer Person mit der Benennung "Pieter Brueghel" ausgeführt wurde; dies gibt den unteren Ast.

