Name:

Geburtsdatum:

Matrikelnummer:

Klausur
Informatische Werkzeuge in den Geistes- und Sozialwissenschaften 2

06. August 2020

	Nur zur Korrektur, bitte freilassen!												
Aufgabe	1.1	2.1	2.2	3.1	3.2	4.1	5.1	5.2	6.1	6.2	6.3	Summe	Note
Möglich	4	9	3	4	8	6	5	5	6	8	5	63	
Erreicht													

Klausurnote: Bonuspunkte: Endnote:

The „Lösungen“ der Aufgaben in diesem Dokument sollen den Studierenden als Anfangspunkt für die Beantwortung der Aufgaben dienen. Trotz aller Bemühungen kann es zu unvollständigen oder sogar Fehlern in den „Lösungen“ kommen. Da die Korrektur und Benotung der gestellten Aufgaben niemals lediglich auf einen „Vergleich mit der Musterlösung“ hinausläuft, ist dies auch nicht so schlimm. In jedem Fall sollten die Studierenden die Lösungen nachvollziehen und im Prinzip mittels des Lehrstoffs selbst verifizieren können.

Sollten Sie „Lösungen“ finden, die Sie nicht verstehen oder sogar für fehlerhaft halten diskutieren Sie diese am besten mit den Tutoren oder auf dem Kursforum und benachrichtigen Sie die Lehrenden.

1 Versionskontrolle

Aufgabe 1.1 Beatrice Beispiel führt in ihrer Shell folgende Befehle aus, um ein `git`-Repository zu klonen: 4 Pkt 4 min

```
git clone https://gitlab.cs.fau.de/iwgs-ss20/collaboration.git
cd collaboration
```

Sie verändert danach lokal die Datei "users.txt", indem sie einen Eintrag für sich selbst hinzufügt. Geben Sie alle Shell-Befehle an, die Beatrice nun durchführen muss um ihre Änderungen im remote-Repository zu veröffentlichen.

Lösung:Hier sind alle benötigten Befehle.

```
git add users.txt
git commit -m "Eintrag zu users.txt hinzugefuegt"
git pull
git push
```

Dabei ist `git pull` nur notwendig, wenn in der Zwischenzeit jemand anderes auch Änderungen gepusht hat

2 Datenbanken

Aufgabe 2.1 (Tabellenoperationen 1)
Gegeben ist folgende Tabelle `Animals`. Hierbei ist `ID` ein Primärschlüssel. 9 Pkt 9 min

ID	Name	Binomial	Diet	Status
1	Fiordland penguin	Eudyptes pachyrhynchus	Carnivore	Vulnerable
2	Brown-throated sloth	Bradypus variegatus	Herbivore	Least Concern
3	Common bottlenose dolphin	Tursiops truncatus	Carnivore	Least Concern
4	Green sea turtle	Chelonia mydas	Omnivore	Endangered
5	Great white shark	Carcharodon carcharias	Carnivore	Vulnerable

Geben Sie jeweils die korrekten SQL-Befehle für folgende Operationen an:

1. Sie wollen ein weiteres Tier, den karnivoren Riesen-Fischuhu, der Tabelle hinzufügen. Im Englischen wird dieser "Blakiston's fish owl" genannt. Der wissenschaftliche Name ist "Bubo Blakistoni".

   Der Riesen-Fischuhu ist die größte lebende Eulenart und sein Erhaltungszustand ist stark gefährdet ("Endangered").

2. Sie wollen den Erhaltungszustand (und *nur* den Erhaltungszustand) aller Karnivoren Tiere in der Datenbank einsehen.

3. Sie wollen die Tabelle dahingehend verändern, dass nur noch Tiere enthalten sind, deren Fortbestand zu einem gewissen Grad gefährdet oder bedroht ist. Dafür müssen alle Tiere mit dem Erhaltungszustand "Least Concern" entfernt werden.

Lösung:Hier sind sie entsprechenden Befehle:

```
1. INSERT INTO Animals VALUES (6, "Blakiston's Fish Owl", "Bubo Blakistoni",
                                  "Carnivore", "Endangered");
2. SELECT Status FROM Animals WHERE Diet = "Carnivore";
3. DELETE FROM Animals WHERE Status = "Least Concern";
```

Aufgabe 2.2 (Tabellenoperationen 2)

Ein Kollege von Ihnen möchte die Tabelle `Animals` aus Aufgabe 2.1 nach einer Rechtschreibreform anpassen. Der neue Name des ersten Tieres in der Datenbank ist nun nicht mehr "Fiordland penguin" sondern "Fjordland penguin". Er schlägt folgenden SQL-Befehl vor: 3 Pkt 3 min

```
UPDATE Animals SET Name = "Fjordland penguin";
```

Wo ist das Problem bei diesem Befehl? Was wäre der Effekt? Geben Sie auch den korrekten Befehl an.

Lösung:Der Befehl enthält keine WHERE-Klausel zum Einschränken der Operation, also würden *alle* Namen in der Tabelle geändert.

Korrekt wäre zum Beispiel:

```
UPDATE Animals SET Name = "Fjordland penguin"WHERE ID = 1;
```

3 Bild

Aufgabe 3.1 Wenn wir ein Farbbild (RGB) in Graustufen überführen, verwenden wir in der Regel eine gewichtete Summe der einzelnen Farbkanäle wie z.B.: 4 Pkt 4 min

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

Beschreiben Sie kurz, warum die einzelnen Kanäle hier unterschiedlich stark gewichtet werden, und warum gerade der Grüne Kanal so stark dominiert.

Lösung:Menschliche Augen sind besonders sensitiv gegenüber grünem Licht, dieses hat also den größten Einfluss auf unsere Farbwahrnehmung. Um ein Bild so in Graustufen umzuwandeln, wie wir es wahrnehmen würden, gewichten wir deshalb den grünen Farbkanal stärker.

Aufgabe 3.2 (Gespiegelte Graustufen)

Schreiben Sie eine python-Funktion fused_grey_mirror, die ein Bild `img` als Parameter nimmt und eine veränderte Kopie des Bildes zurück gibt, die sowohl horizontal gespiegelt als auch in Graustufen übersetzt wurde. 8 Pkt 8 min

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

```python
from PIL import Image

def fused_grey_mirror(img) :
    # Neues Bild erstellen. "L" bedeutet Graustufe.
    result = Image.new("L", (img.width, img.height), 0)

    # Ihr Code hier...

    return result
```

Hinweis: Ein paar nützliche Informationen finden Sie im Folgenden:

- 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)`

Lösung:Hier ist eine mögliche Implementation:

```python
from PIL import Image, ImageOps

def fused_grey_mirror(img):
    # Neues Bild erstellen. "L" bedeutet Graustufe.
    result = Image.new("L", (img.width, img.height), 0)

    for x in range(result.width):
        for y in range(result.height):

            # Farbwerte vom Original im Pixel (x,y)
            (r,g,b) = img.getpixel((x,y))

            # Umrechnen in Graustufe
            np = int(0.21 * r + 0.71 * g + 0.08 * b)

            # Wir setzen das Pixel ganz links im Original nach ganz rechts usw.
            # (-1) weil x-Werte bei einer Breite von 100 nur 0-99 annehmen.
            xneu = result.width - x - 1

            # Farbe im Ergebnis setzen
            result.putpixel((xneu,y), np)

    return result
```

---

4 Web-Applikationen

Aufgabe 4.1 6 Pkt

6 min

Schreiben Sie eine Route testrequest() für einen `bottle`-Server, die *keine* Argumente nimmt, dafür jedoch einen HTTP GET-Request verarbeitet und eine Zeile HTML zurück gibt (i.e. anzeigt). In diesem GET-Request werden zwei Argumente übergeben (size und payload).

Liegt `size` zwischen 1 und 6, so soll eine entsprechende HTML- Überschrift zurück gegeben werden (z.B. `<h3>...</h3>` falls size = 3). In allen anderen Fällen soll lediglich ein normaler Absatz (`<p></p>`) zurück gegeben werden.

In dem entsprechenden Tag soll der Inhalt (normalerweise also Text) des Arguments payload stehen. Sollte dieses leer sein, soll stattdessen "No text available!" angezeigt werden.

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('/testrequest')
def testrequest():

    # Request auslesen.
    size = request.query.size
    payload = request.query.payload

    # Falls size nicht vorhanden.
    if size == "":
        size = 0

    # Falls payload nicht vorhanden.
    if payload == "":
        payload = "No text available!"

    # Passendes HTML returnen.
    if int(size) > 0 && int(size) < 7:
        return "<h" + size + ">" + payload + "</h" + size + ">"
    else:
        return "<p>" + payload + "</p>"
```

5 Geistiges Eigentum & Datenschutz

Aufgabe 5.1 (CopyLeft)

Nennen Sie die „Copyleft"-Klausel in der GNU Public License oder in den Creative Commons Lizenzen, und erklären Sie kurz in eigenen Worten die Wirkungsweise. 5 Pkt

5 min

Lösung:Die „CopyLeft"-Klausel besagt, dass wen ein vom lizenzierten Werk abgeleitetes Werk weiterverbreitet wird, dann muss es mit genau der gleichen Lizenz lizensiert werden wie das Original.

Dadurch wird sichergestellt, dass alle, die auf ein so lizensiertes Werk aufbauen, sich entscheiden müssen, ob sie

- es weiterverbreiten wollen – dann müß das abgeleitete Werk dem „CopyLeft“ unterliegen und so dem gemeinfreien Bereich zugeordnet werden, oder
- nicht, dann muss das abgeleitete Werk nicht lizensiert werden, (aber die Schöpfer bekommen erzielen keinen Mehrwert aus Verbreitung oder Verkauf).

Aufgabe 5.2 (DSGVO)

Beschreiben Sie kurz in eigenen Worten wer oder was die DSGVO (*engl. = GDPR*) ist, wen sie primär betrifft und unter welchen Umständen sie auch für Dritte (außerhalb des primären Einflussgebietes) relevant ist.

5 Pkt
5 min

Lösung:DSGVO steht für Datenschutz-Grundverordnung. Das ist eine Verordnung der Europäischen Union, mit der die Regeln zur Verarbeitung personenbezogener Daten durch die meisten Datenverarbeiter, sowohl private wie öffentliche, EU-weit vereinheitlicht werden.

Primär betrifft diese nur Datenverarbeiter innerhalb der EU. Datenverarbeiter außerhalb der EU sind allerdings ebenfalls betroffen, falls sie Personen in der EU Waren oder Dienstleistungen anbieten oder deren Verhalten verfolgen.

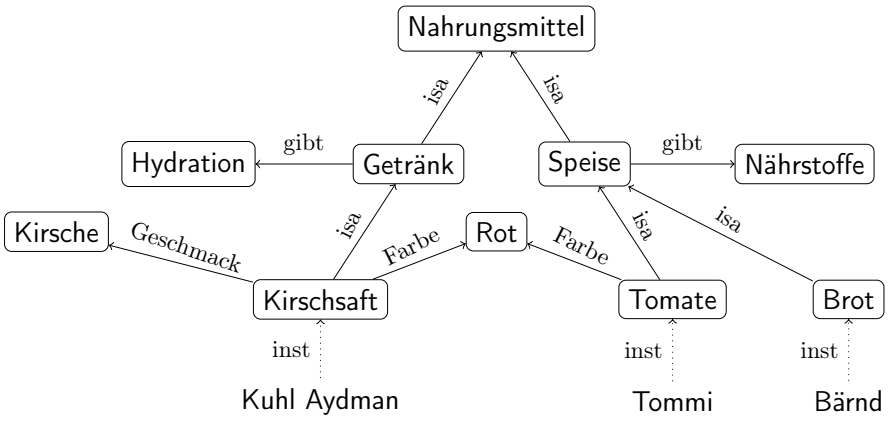
---

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.

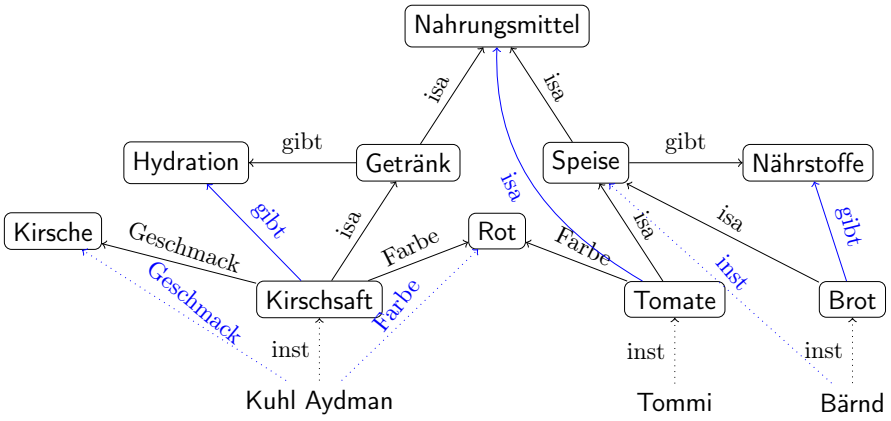
2. Zeichnen Sie jeweils drei Fakten und terminologische Axiome in das Netz ein, die im Netz inferiert werden können aber nicht explizit repräsentiert sind.

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

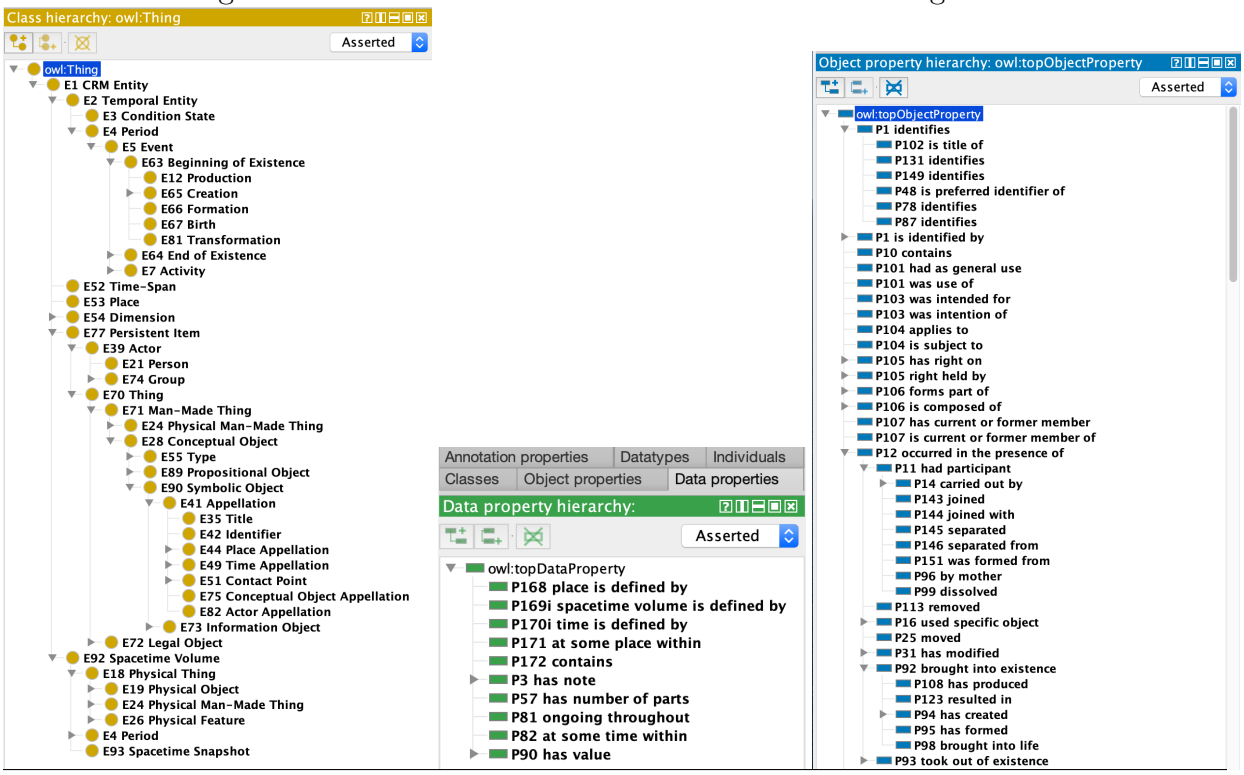
2. Wir zeichnen die neuen Relationen in blau ein:



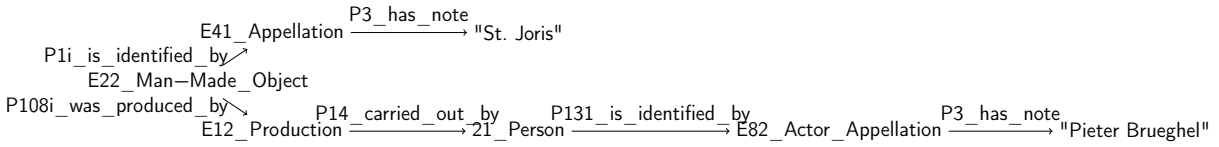
Aufgabe 6.2 (Modellierung in CIDOC-CRM)

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. 8 Pkt 8 min

Zur Orientierung drucken bilden wir Teile der CIDOC CRM Ontologie unten ab.



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.



Aufgabe 6.3 Beschreiben Sie in eigenen Worten das Konzept des “WissKI path builder” und welches Problem er löst. 5 Pkt 5 min

Lösung:In CIDOC CRM sind die Relationen sehr feingranular, daher entsprechen bei der Modellierung von Wissen über kulturelle Artefakte in CIDOC CRM oft schon einfache Fakten ganzen Ontologiepfaden, die viele Zwischenobjekte enthalten. Im WissKI path builder können Feldern in Erfassungsmasken Pfade zugeordnet werden, die bein Ausfüllen erzeugt werden.