

Name:

Geburtsdatum:

Matrikelnummer:

Nachklausur Informatische Werkzeuge in den Geistes- und Sozialwissenschaften 2

09. Oktober 2019

	Nur zur Korrektur, bitte freilassen!										
Aufgabe	1.1	1.2	2.1	3.1	3.2	3.3	4.1	5.1	6.1	Summe	Note
Möglich	4	4	20	4	6	4	8	4	6	60	
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 Beispielperson 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-ss19/collaboration.git
cd collaboration
```

Sie verändert danach lokal die Datei “users.txt” und führt folgende Befehle durch um ihre Änderungen zu veröffentlichen.

```
git add users.txt
git push
```

Leider hat dies nicht den gewünschten Effekt. Was hat Beatrice falsch gemacht? Welcher Befehl hätte ausgeführt werden müssen? Erklären Sie kurz, warum.

Lösung: Beatrice hat alles richtig gemacht, sie hat lediglich vergessen, ihre Änderungen auch zu comitten, nachdem sie sie gestaged hat.

Nach dem `git add`-Befehl hätte sie noch einen Befehl wie `git commit -m "changes to users.txt"` ausführen müssen.

Aufgabe 1.2 Was ist ein “Diff” zwischen zwei Textdateien? Welche Rolle spielen Diffs in der Versionskontrolle? 4 Pkt
4 min

Lösung: Ein *Diff* zwischen zwei Dateien ist eine Liste von Unterschieden (engl.: *differences*) (auf Zeilenbasis). Ein Diff enthält also alle Informationen darüber, wie die erste Datei geändert werden müsste, um identisch zur Zweiten zu sein.

Hier ist ein Beispiel:

```
jbetzend@turing :: 11:15:05 :: ~/Desktop > # Here are some files
jbetzend@turing :: 11:15:12 :: ~/Desktop > cat a.txt
This line is in both files.
This one isn't.
This one is a bit different.
jbetzend@turing :: 11:15:16 :: ~/Desktop > cat b.txt
This line is in both files.
This one is slightly different.
jbetzend@turing :: 11:15:18 :: ~/Desktop > # And now the Diff
jbetzend@turing :: 11:15:25 :: ~/Desktop > diff a.txt b.txt
2,3c2
< This one isn't.
< This one is a bit different.
---
> This one is slightly different.
```

Versionskontrollsysteme benutzen Diffs um anzugeben, welche Änderungen eine Datei erfahren hat. So benötigt ein Repository deutlich weniger Speicherplatz als wenn jede Version der Datei (Snapshot) gespeichert werden müsste.

2 Datenbanken

Aufgabe 2.1 (Buchverwaltung)

Gegeben ist folgender Datensatz, der den Bestand einer Bibliothek modelliert:

20 Pkt

20 min

BuchID	Titel	AusgeliehenVon	Kundennummer
123	Bürgerliches Gesetzbuch	Magdalena Musterfrau	00334
137	Die Physiker	-	-
055	Darm mit Charme	-	-
249	Die Verwandlung	Jan Modaal	75187
005	Faust	Jan Modaal	75187

Die **BuchID** identifiziert ein Buch eindeutig (Primary Key). Sie können davon ausgehen, dass es jedes Buch nur ein Mal in der Bibliothek gibt.

Sie wissen außerdem, dass für einen gleichen Eintrag in der Spalte **AusgeliehenVon** auch der Eintrag in der Spalte **Kundennummer** gleich sein wird.

Ein Bindestrich in den Spalten **AusgeliehenVon** und **Kundennummer** signalisiert, dass dieses Buch derzeit nicht ausgeliehen ist (NULL bzw. None).

- 1. Tabelle erstellen (5 Punkte):** Geben Sie die SQL-Befehle zum Erstellen einer Tabelle unter dem Namen „Books“ mit der obigen Struktur an. Sie müssen keine Einträge hinzufügen. Vermeiden Sie dabei eventuelle Probleme, falls bereits eine Tabelle namens „Books“ existiert.
- 2. Daten anpassen (4 Punkte):** San Zhang (Kundennummer 67673) möchte das Buch *Die Physiker* ausleihen. Geben Sie den SQL-Befehl an, der die obige Tabelle entsprechend anpasst.
- 3. Redundanz (3 Punkte):** In der obigen Tabelle werden einige Daten mehrfach gespeichert. Begründen Sie, warum das (gerade für größere Datensätze) suboptimal ist.
- 4. Praxis (8 Punkte):** Schreiben Sie ein python-Programm, welches den Nutzer wiederholt nach einem Buchtitel fragt. Verwenden Sie hierfür die python-Funktion `input`. Ihr Programm soll dann den Namen der Person ausgeben (`print`), welche das Buch derzeit ausgeliehen hat. Fall das Buch nicht ausgeliehen ist, oder der Name des Buchs nicht in der Datenbank gefunden werden kann, soll eine entsprechende Meldung ausgegeben werden. Die beschriebene Funktionalität soll solange ausgeführt werden, bis der Benutzer `quit` tippt.

Lösung:

1. Folgende Befehle erstellen die Datenbank korrekt:

```
DROP TABLE IF EXISTS Books;
CREATE TABLE Books(
BuchID INTEGER PRIMARY KEY,
Titel TEXT,
AusgeliehenVon TEXT,
Kundennummer INTEGER);
```

2. Der Befehl lautet wie folgt:

```
UPDATE Books
SET AusgeliehenVon = "San Zhang", Kundennummer = "67673"
WHERE BookID = 137;
```

3. In der Tabelle wird sowohl AusgeliehenVon als auch die Kundennummer festgehalten, obwohl bekannt ist, dass diese immer korrelieren. Diese Informationen könnten also (z.B. in eine zweite Tabelle "Kunden") ausgelagert werden.

Die doppelte Vorhaltung der Daten ist problematisch weil so leichter Fehler / Inkonsistenzen entstehen und mehr Speicherplatz verbraucht wird, als nötig.

4. Hier ist ein solches Programm:

```
import sqlite3

userin = ""

db = sqlite3.connect("klausur.db")

while userin != "quit" :
    userin = input("Titel:")
    if userin != "quit" :
        cursor = db.cursor()
        cursor.execute("SELECT AusgeliehenVon FROM Books Where Titel=?", (userin,))
        result = cursor.fetchone()
        if result :
            if result[0] :
                print(result[0])
            else :
                print("Nicht ausgeliehen")
        else :
            print("Buch nicht gefunden.")
```

3 Bild

Aufgabe 3.1 (CSS-Anonymisierung)

Angenommen, Sie entwickeln eine Webseite, auf der die persönlichen Daten von vielen 4 Pkt

4 min

Benutzern sichtbar sind. Aus Datenschutzgründen entscheiden Sie sich dazu, diese Daten unlesbar darzustellen, indem Sie einen *Blur*-Filter auf den Text und die Bilder anwenden:

```
<style>
    .anonym {
        filter: blur(10px);
    }
</style>

<!-- Alle persönlichen Daten haben die Klasse 'anonym'. -->
<p class='anonym'>Max Mustermann</p>
<img class='anonym' src='...' />

...
```

Sind die Daten mit dieser Methode für Nutzer Ihrer Webseite ausreichend geschützt? Begründen Sie Ihre Antwort.

Lösung: Nein! Die Daten werden immernoch als Klartext an den Client geschickt. Erst der Browser sorgt dafür, dass die Texte und Bilder verwaschen werden. Benutzer können die Daten einsehen, indem Sie den Seitenquelltext anschauen, oder entsprechende Texte einfach markieren und kopieren, bzw. die Bilder abspeichern.

Aufgabe 3.2 (Bilder invertieren)

Schreiben Sie eine python-Funktion `invert(image)`. Das Argument ist ein Bild, repräsentiert als Liste von Listen (von Pixeln). 6 Pkt
6 min

- Die Einträge der inneren Listen sind alle jeweils ein Pixel, hier ein Tupel im Format (R,G,B).
- Die Farbwerte in jedem Kanal liegen zwischen 0 und 255.
- Ihre Funktion soll `image` invertieren. Das bedeutet, dass ein Farbwert von 0 nach dem Invertieren den Wert 255 haben soll, 1 soll zu 254 konvertiert werden, und so weiter.

Lösung:

```
example = [[(1,11,21),(1,21,41)],[(32,16,1),(52,26,1)]]
```

```
def invert(pixels):
```

```
    # Das gesamte Bild ist eine Liste von Listen.
```

```
    # Dies ist unsere äussere neue Liste.
```

```
    newImg = []
```

```
    for i in pixels:
```

```
        # Innere neue Liste fuer jede in Liste in der äusseren.
```

```
        newList = []
```

```
        for j in i:
```

```
            # Invertieren.
```

```
            newPixel = (255 - j[0], 255 - j[1], 255 - j[2])
```

```
            # Haengt das neue Pixel an die neue innere Liste an.
```

```

newList.append(newPixel)
# Haengt die neue innere Liste an die neue aeuessere an.
newlmg.append(newList)
return newlmg

print(invert(example))

```

Aufgabe 3.3 (Vektorgrafiken)

Vektorgrafiken speichern Bildinformationen nicht in Form eines Rasters von Pixeln, sondern als Liste von Primitiven (Kreise, Rechtecke, Polygone, etc). 4 Pkt
4 min

Erläutern Sie, wie diese Grafiken trotzdem auf Raster-Bildschirmen angezeigt werden können. Was bedeutet in diesem Kontext der Begriff *Anti-Aliasing*?

Lösung: Vektorgrafiken müssen vor der Anzeige auf einem Rasterbildschirm rasterisiert werden. Dabei evaluiert ein Programm für jede Form (Rechteck, Kreis, Polygon, ...) ob ein gegebener Pixel innerhalb der Form oder außerhalb der Form ist und weist dem Pixel gegebenenfalls die jeweilige Farbe zu.

Bei binären Entscheidungen (Pixel ist innerhalb der Form oder außerhalb, ja oder nein) entstehen dabei unschöne Treppenstufen bei diagonalen Kanten. Das liegt daran, dass Pixel keine Punkte sind, sondern eine gewisse Ausdehnung haben. Um die visuelle Qualität zu verbessern, versuchen Anti-Aliasing-Algorithmen zu entscheiden, wieviel Prozent eines Pixels eine Form überlappt. Die zugewiesene Form ist dann eine gewichtete Summe aus der Farbe der Form und des Hintergrundes. Dadurch werden Kanten weicher.

4 Web-Applikationen

Aufgabe 4.1 (Bottle-Route für Namen)

Gegeben ist ein simpler Webserver über `bottle` in der Datei `name-server.py` (In dieser Aufgabe ist kein HTML und kein Templating erforderlich). 8 Pkt
8 min

Vervollständigen Sie folgendes Code-Gerüst, sodass auf der Webseite ein String mit vier Namen, die durch Kommata und Leerstellen getrennt sind angezeigt wird. Die Namen sollen „Michael“, „Philipp“, „Jonas“ sowie ein weiterer Name sein, der als Argument der Route übergeben wird. Weiterhin sollen alle vier Namen *in alphabetischer Reihenfolge* erscheinen.

Beispiel: An der Route `/iwgs/Mustermensch` sollte also folgender String angezeigt werden:

Jonas, Michael, Mustermensch, Philipp

Hinweis: In `python` ist die übliche lexikographische Ordnung auf Strings bereits in den Operatoren `<` und `>` implementiert. So evaluiert zum Beispiel der Ausdruck `"Jacqueline"<"Michael"` zu `True` und der Ausdruck `"Jacqueline">="Philipp"` zu `False`, weil „Jacqueline“ alphabetisch vor den beiden anderen Namen kommt.

```
# DATEI: name-server.py
```

```
@bottle.route('/iwgs/<name>')  
def sortedNames(name) :  
    # YOUR CODE HERE
```

Lösung: Hier ist die vollständige `name-server.py`:

```
import bottle
```

```
@bottle.route('/')  
def landingPage():  
    link = "<a href=\"./iwgs/Mustermensch\">IWGS names</a> route?"  
    return "Hello! Are you looking for the " + link
```

```
@bottle.route('/iwgs/<name>')  
def sortedNames(name) :  
    if name > "Philipp":  
        return "Jonas, Michael, Philipp, " + name  
    elif name > "Michael":  
        return "Jonas, Michael, " + name + ", Philipp"  
    elif name > "Jonas":  
        return "Jonas, " + name + ", Michael, Philipp"  
    else:  
        return name + ", Jonas, Michael, Philipp"
```

```
bottle.run(host='localhost', port=8080, reloader=True)
```

Wohlgemerkt war es nicht Teil der Aufgabe, die `/`-Route und den Befehl zum Start des Bottle-Servers anzugeben, die sind lediglich der Vollständigkeit halber hier auch angeführt.

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.

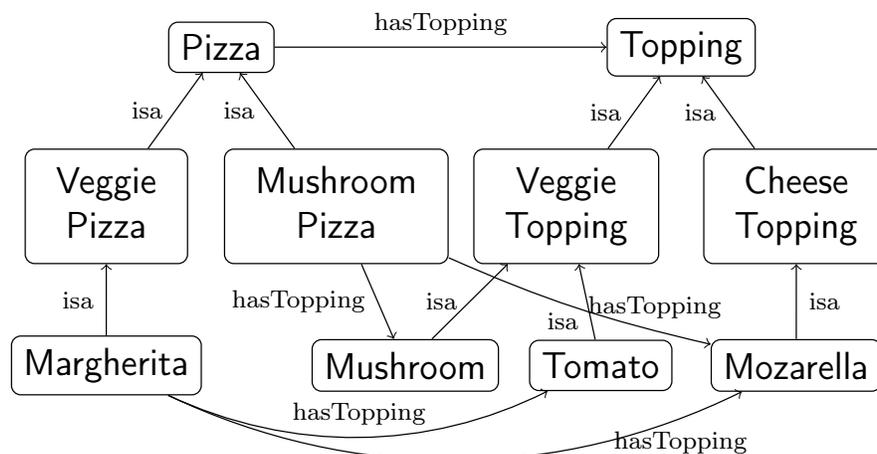
6 Semantic Web

Aufgabe 6.1 (Semantische Netze (Pizza))

Gegeben sei das folgende semantische Netz:

6 Pkt

6 min



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

Lösung:

- Subjekt/Prädikat/Objekt

1. in TBox	2. inferiert
Margherita/hasTopping/Mozzarella	Margherita/hasTopping/CheeseTopping
Mozzarella/isa/Cheese Topping	Mozzarella/isa/Topping
Pizza/hasTopping/Topping	Margherita/hasTopping/Topping