Probeklausur
Informatische Werkzeuge in den Geistes- und Sozialwissenschaften II

16. Juli 2020

Diese Probeklausur ist absolut freiwillig und soll Ihnen dazu dienen die Klausur besser einzuschätzen.

Falls gewünscht, *können* Lösungen an die Tutoren zur Korrektur eingereicht werden. In jedem Fall wird nächste Woche eine Musterlösung bereitgestellt.

Die vorliegenden Übungen sind in etwa vergleichbar mit tatsächlichen Klausuraufgaben, daher haben wir auch am Rand (hoffentlich realistisch geschätzte) Zeiten angegeben.

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 (Dezentrale Versionskontrolle)
Erklären Sie kurz (!) das Konzept von dezentralen Versionskontrollsystemen. Können `git`-Repositorien auch ohne einen Server wie `GitHub` oder `GitLab` betrieben werden? 4 min

Aufgabe 1.2 (git Beispielablauf)
Betrachten Sie folgende beispielhafte Zeitlinie. Markieren Sie jede Zeile jeweils mit “Kein Problem” oder “Problem”, falls bei der Ausführung eine Fehlermeldung auftreten sollte. 4 min

Falls ein Problem auftritt, beschreiben Sie es und geben Sie den Lösungsweg an.

1. Person A klont das Repositorium R,
2. Person B klont auch R,

3. Person A verändert eine Datei `foo.txt` in Zeile 10,
4. Person A führt einen `git add foo.txt` und `git commit` aus,
5. Person B verändert die Datei `foo.txt` in Zeile 10,
6. Person B führt `git add foo.txt`, `git commit` und `git push` aus,
7. Person A führt `git push` aus,
8. Person A verändert die Datei `foo.txt` in Zeile 23,
9. Person A führt `git push` aus.

Aufgabe 1.3 (Definition von pull)
Was ist gemeint, wenn im Kontext von Versionskontrolle von "`pull = fetch + merge`" gesprochen wird? 4 min

2 Datenbanken

Aufgabe 2.1 (Datenbanktabelle)
Sie möchten eine Datenbanktabelle für die Mitarbeitenden der Capitalist Corporation anlegen, die `PersonalID`, `Name`, `Vorname`, `Jobtitel` und `Jahresgehalt` erfasst. Der Name der Tabelle soll "Angestellte" sein. Die `PersonalID` ist eine ganze Zahl und ein Primärschlüssel. 4 min

Geben Sie den entsprechenden `SQL`-Befehl an.

Aufgabe 2.2 (Gleiches Gehalt)
Das Jahresgehalt ergibt sich in der Capitalist Corporation ausschließlich durch den Jobtitel, das heißt alle Mitarbeitenden mit dem selben Titel verdienen auch das selbe Gehalt. 4 min

1. Was ist das Problem bei der Tabelle aus Aufgabe 2.1?
2. Was wäre in diesem Fall die effizientere Weise, die Daten der Mitarbeitenden zu speichern?
3. Welche Vorteile hätte die Alternative?

Sie müssen die Alternative nicht implementieren, nur beschreiben.

Aufgabe 2.3 (Nils Nichtig Heiratet Nelly Null)
Der Mitarbeiter Nils Nichtig (PersonalID 1729) heiratet Nelly Null und nimmt ihren Nachnamen an. Wie könnte dies zu Problemen führen, wenn der entsprechende Eintrag in der Datenbank geändert wird? 3 min

Aufgabe 2.4 (Capitalist Corporation Ändert den Namen)
Geben Sie den *korrekten* `SQL`-Befehl an, der den Nachnamen des Mitarbeiters Nils Nichtig aus Aufgabe 2.3 ändert, nachdem er Nelly Null geheiratet hat. 2 min

Aufgabe 2.5 (Noch gleicheres Gehalt)
Capitalist Corporation wird umstrukturiert zu Community Collective. Alle Mitarbeitenden sollen ab sofort das gleiche Gehalt bekommen. Schreiben Sie ein python-Programm was alle Gehälter (und nur die Gehälter) abfragt, aufsummiert, durch die Anzahl der Mitarbeitenden teilt und allen dieses Gehalt gibt. 10 min

3 Bilder und Bildmanipulation

Aufgabe 3.1 (Beatrices Beispiel-App)
Beatrice Beispiel hat eine App programmiert, mit der BenutzerInnen Bilder zeichnen können und diese mit ihren Freundinnen und Freunden teilen können. Diese wiederum können die Bilder wieder verändern und zurückschicken und so weiter. Die Bilder werden im JPEG-Format verschickt. 4 min

Nach kurzer Zeit bemerken BenutzerInnen seltsame Artefakte in den Bildern. Was ist das Problem?

Aufgabe 3.2 (Raster- und Vektorgrafiken)
Erklären Sie kurz den Unterschied zwischen Vektor- und Rastergrafiken. Warum macht das Konzept von "Auflösung" nur bei einem der beiden Sinn? 4 min

Aufgabe 3.3 (Kantenerkennung)
Beschreiben Sie kurz in eigenen Worten, wie Kantenerkennung auf Bildern funktioniert. 5 min

Aufgabe 3.4 (Kantenerkennung mit Sobel-Filter)
Schreiben Sie eine python-Funktion edge_detection, die ein Bild `img` und eine Zahl `threshold` als Parameter nimmt. 12 min

Ihre Funktion soll einen Sobel-Filter (wie in der Vorlesung) implementieren und ein neues Bild zurückgeben, in welchem für jeden Pixel gespeichert ist, ob es Teil einer Kante ist oder nicht. Somit sollte in jedem Pixel, durch den eine Kante geht, der Wert 255 gespeichert sein und in jedem anderen Pixel der Wert 0.

Verwenden Sie *nicht* die `Pillow`-Funktion img.filter. Folgendes Codegerüst ist gegeben:

```
from PIL import Image

def edge_detection(img, threshold) :
    # Neues Bild erstellen. "L" bedeutet grayscale.
    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
- Beide Bilder sind Graustufen-Bilder, haben also nur einen Kanal

- Sie können den Wert eines Pixels wie folgt auslesen:
  `value = img.getpixel((x, y)) # Nur ein Wert pro Pixel, denn das Bild ist in Graustufen`
- Sie können den Wert eines Pixels wie folgt setzen:
  `img.putpixel((x, y), value)`
- Der Sobel-Filter hat folgende Gewichte:

für horizontale Kanten	für vertikale Kanten
$\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$	$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$

- Da der Filter auf Nachbar-Pixel zugreift, müssen Sie beachten, dass Sie nicht auf Pixel außerhalb des Bildes zugreifen. Führen Sie dazu den Filter und damit die Kanten-Entscheidung nur auf Pixeln durch, die nicht am Rand des Bildes liegen.
