

Name:

Vorname:

Geburtsdatum:

Matrikelnummer:

Platz:

Nachklausur
Informatische Werkzeuge in den Geistes-
und Sozialwissenschaften 2

13. Oktober 2022

	To be used for grading, do not write here										
prob.	1.1	2.1	2.2	2.3	2.4	3.1	3.2	3.3	4.1	Sum	grade
total	15	4	6	8	6	7	4	5	5	60	
reached											

Organisatorisches

Bitte lesen die folgenden Anweisungen genau und bestätigen Sie diese mit Ihrer Unterschrift.

1. Bitte legen Sie Ihren Studierendenausweis und einen Lichtbildausweis zur Personenkontrolle bereit!
2. Die angegebene Punkteverteilung gilt unter Vorbehalt.
3. Sie können 60 Punkte erreichen, wenn Sie alle Aufgaben vollständig lösen. Allerdings zählen 55 Punkte bereits als volle Punktzahl, d.h. 5 Punkte sind Bonuspunkte.
4. Es sind keine Hilfsmittel erlaubt außer eines handgeschriebenen "Spickzettels" von 1 Seite A4 einseitig.
5. Die Bearbeitungszeit beträgt genau 60 min.
6. Schreiben Sie die Lösungen direkt auf die ausgeteilten Aufgabenblätter. Andere Blätter werden nicht bewertet.
7. 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.
8. Überprüfen Sie Ihr Exemplar der Klausur auf Vollständigkeit (16 Seiten exklusive Deckblatt und Hinweise) und einwandfreies Druckbild! **Vergessen 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, 13. Oktober 2022

.....
(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 Datenbanken

Aufgabe 1.1 (Verwaltung einer Spieledatenbank)

15 Pkt

Gegeben ist die folgende SQL-Tabellen `Mitarbeitende`. Die Spalte `PersonID` ist dabei ein *Primärschlüssel* (“Primary Key”).

PersonID	Name	Abteilung	Gehalt
0	Gertrud Gründer	Management	80000
1	Franziska Finanz	Management	90000
2	Detlef Designer	Produktentwicklung	40000
3	Sabrina Server	IT	35000
4	Beatrice Beispiel	Praktikum	0

- (3 Punkte)* Geben Sie einen SQL-Befehl an, der die Namen aller festen Mitarbeitenden (`Abteilung` darf also nicht “Praktikum” sein) abfragt.
- (3 Punkte)* Geben Sie einen SQL-Befehl an, mit dem die Gehälter aller der Mitarbeitenden auf 0 gesetzt werden, die aktuell ein Gehalt von mehr als 60000 haben.
- (3 Punkte)* Geben Sie einen SQL-Befehl an, mit dem *alle* Einträge in der Tabelle gelöscht werden. Die Tabelle selbst soll bestehen bleiben.
- (3 Punkte)* Geben Sie einen SQL-Befehl an, mit dem eine neue Tabelle `Produkte` angelegt wird, falls diese noch nicht existiert. Diese Tabelle soll die Spalten `ProductID` (Integer, Primärschlüssel), `Bezeichnung` (Text), `Preis` (Integer) und `Verantwortung` (Integer) haben.
Dabei soll die Spalte `Verantwortung` ein *Fremdschlüssel* (“Foreign Key”) sein, der auf `PersonID` in `Mitarbeitende` verweist.
- (3 Punkte)* Gehen Sie nun davon aus, dass die Tabelle `Produkte` einige Einträge erhalten hat und geben Sie einen SQL-Befehl an, der alle Bezeichnungen von Produkten und den Namen der jeweils verantwortlichen Person abfragt. Führen Sie dafür einen „Inner Join“ von `Produkte` und `Mitarbeitende` auf der Spalte `PersonID/Verantwortung` durch.

This page was intentionally left blank for extra space

2 Bild

Aufgabe 2.1

4 Pkt

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

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

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

Aufgabe 2.2 (Abgesicherter Zugriff auf Pixel)

6 Pkt

Schreiben Sie eine python-Funktion `getPixelSafe()`, die ein (Pillow-)Bild `img`, sowie zwei Integer (für X- und Y-Koordinate) als Parameter entgegen nimmt.

Sind beide Koordinaten passend für das Bild, so sollen die RGB-Werte des Bildes an der gegebenen Koordinate zurück gegeben werden.

Ist jedoch mindestens eine der Koordinaten zu groß oder zu klein für das gegebene Bild (z.B. eine X-Koordinate von -5 oder 400 bei einem Bild, das nur 200 Pixel breit ist), so soll `(0,0,0)` zurück gegeben werden.

Hinweis: Ein paar nützliche Informationen finden Sie im Folgenden:

- Sie können den Wert eines Pixels in Pillow-Bildern wie folgt auslesen:
`img.getpixel((x, y))` Gibt ein 3-Tupel `(r,g,b)` zurück.
 - Die Breite bzw. Höhe eines Bildes können Sie mit `img.width` bzw. `img.height` abfragen.
-

Aufgabe 2.3 (Helligkeit der Nachbarschaft eines Pixels)

8 Pkt

Schreiben Sie eine python-Funktion `neighbourhood_brightness()`, die ein Bild `img`, sowie zwei Integer (für X- und Y-Koordinate) als Parameter entgegen nimmt und die Helligkeit der Nachbarschaft um die gegebene Koordinate zurück gibt.

Wir verwenden hier der Einfachheit halber eine sehr naive Interpretation von „Helligkeit“. Um die Helligkeit eines Pixels zu ermitteln, addieren Sie bitte einfach die Werte der RGB-Kanäle des Pixels auf. Die Helligkeit einer Nachbarschaft entspricht der Summe der Helligkeit aller Pixel.

Die Nachbarschaft eines Pixels ist das Pixel selbst und alle benachbarten Pixel, horizontal, vertikal und diagonal. Wenn eine Nachbarschaft eines Pixels über die Grenzen des eigentlichen Bildes hinaus geht, so haben diese Pixel eine Helligkeit von 0.

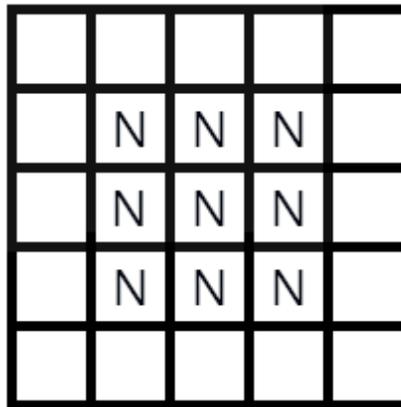


Abb. 1: Raster mit Koordinaten (0,0) bis (4,4). Für die Koordinate (2,2) ist die Nachbarschaft genau die Felder, die mit N markiert sind.

Hinweis: Sie können für die Lösung dieser Aufgabe davon ausgehen, dass Ihnen die Funktion `getPixelSafe()` aus Aufgabe 2.2 zur Verfügung steht (auch wenn Sie diese nicht erfolgreich bearbeitet haben).

This page was intentionally left blank for extra space

Aufgabe 2.4 (Finden der hellsten Nachbarschaft)

6 Pkt

Schreiben Sie eine python-Funktion `brightest_area()`, die ein (Pillow-)Bild `img` als Parameter entgegen nimmt. Der Rückgabewert soll die Koordinate des Pixels mit der hellsten Nachbarschaft des Bildes sein, in der Form eines Tupels `(x,y)`.

Sollte der hellste Wert von mehr als einer Koordinate erreicht werden, ist egal, welche davon Sie zurück geben.

Hinweis: Sie können für die Lösung dieser Aufgabe davon ausgehen, dass Ihnen die Funktion `neighbourhood_brightness()` aus Aufgabe 2.3 zur Verfügung steht (auch wenn Sie diese nicht erfolgreich bearbeitet haben).

Erinnerung: Die Breite bzw. Höhe eines Bildes können Sie mit `img.width` bzw. `img.height` abfragen.

This page was intentionally left blank for extra space

3 Semantic Web & Kulturelles Erbe

Aufgabe 3.1 (Semantische Netze)

7 Pkt

Zeichnen Sie ein semantisches Netz nach Ihren eigenen Vorstellungen. Das Netz soll mindestens 8 Knoten und 10 Kanten enthalten, davon mindestens jeweils 3 Inklusionen (*isa*-Kanten) und Instanziierungen (*inst*-Kanten).

Geben Sie außerdem jeweils drei Subjekt/Prädikat/Objekt-Tripel aus der TBox und der ABox Ihres Netzes an.

Aufgabe 3.2 (Semantic Web Inferenz)

4 Pkt

Gegeben sind die folgenden Tripel:

`("Lars", "inst", "Eisbär")`

`("Eisbär", "is_a", "Tier")`

Warum können wir in diesem Kontext *nicht* folgern, dass `("Lars", "is_a", "Tier")`?
Was wäre stattdessen richtig?

Aufgabe 3.3 (SPARQL-Anfrage)

5 Pkt

Geben Sie eine vollständige SPARQL-Query für dbpedia oder eine vergleichbare Quelle an, die alle Object Properties auflistet, die den Artikel für *Kreuzberg* aus der Quelle mit dem Artikel für *Berlin* verbindet. In anderen Worten suchen wir die Prädikate aller Subjekt-Prädikat-Object-Tripel in denen Kreuzberg das Subjekt und Berlin das Objekt ist.

Hinweis: Wir erwarten nicht, dass Sie alle relevanten Ontologien auswendig können oder mitgebracht haben.

Sie können also die fiktive Ontologie UCO (für “*Universally Convenient Ontology*”) benutzen und relevante Object Properties etc. dafür erfinden. So könnte zum Beispiel Berlin als `uco:Berlin` modelliert sein.

4 Geistiges Eigentum

Aufgabe 4.1 (Kombination von CC-lizenzierten Werken)

5 Pkt

Beatrice hat einen bösen Brief von einem Anwalt bekommen. Sie hat ein Buch geschrieben und es unter der Lizenz „Creative Commons - Attribution“ (CC-BY) veröffentlicht. In diesem Buch verwendet sie ein Bild eines Faultiers, welches unter „Creative Commons - Attribution NonCommercial ShareAlike“ (CC-BY-NC-SA) veröffentlicht wurde. Der Brief vom Anwalt sagt nun, dass sie dies nicht hätte tun dürfen. Beatrice ist verwirrt und sagt, sie hätte vorher extra in folgender Tabelle nachgesehen, dass die Lizenzen kompatibel wären.

	 PUBLIC DOMAIN	 PUBLIC DOMAIN	 CC BY	 CC BY SA	 CC BY NC	 CC BY ND	 CC BY NC SA	 CC BY NC ND
 PUBLIC DOMAIN	✓	✓	✓	✓	✓	✗	✓	✗
 PUBLIC DOMAIN	✓	✓	✓	✓	✓	✗	✓	✗
 CC BY	✓	✓	✓	✓	✓	✗	✓	✗
 CC BY SA	✓	✓	✓	✓	✗	✗	✗	✗
 CC BY NC	✓	✓	✓	✗	✓	✗	✓	✗
 CC BY ND	✗	✗	✗	✗	✗	✗	✗	✗
 CC BY NC SA	✓	✓	✓	✗	✓	✗	✓	✗
 CC BY NC ND	✗	✗	✗	✗	✗	✗	✗	✗

Abb. 2: Kompatibilität von Creative Commons Lizenzen

Erklären Sie, wie es sein kann, dass Beatrice rechtlichen Ärger für ihre Verwendung des Faultierbildes bekommt, obwohl die Tabelle die beiden relevanten Lizenzen als kompatibel listet.

This page was intentionally left blank for extra space

This page was intentionally left blank for extra space

This page was intentionally left blank for extra space

This page was intentionally left blank for extra space