**WS 23 Übung 3 GIN1IL\_T1**

# Abgabefrist: 10.11 23:55 Erreichte Punkte: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

8 h

Katrin Szikora

Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Zeitaufwand in h: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*Beachten Sie die Abgabekriterien! (siehe LVA Übersicht)*

# Online Bibliothek 5 Punkte

Erstellen Sie ein Use Case Diagramm für die Website einer Online Filmebewertung. Die Use Cases folgender Aktoren sollen abgedeckt werden:

* User, die Filmebewertungen lesen und Zuschauerbewertungen abgeben können
* KritikerInnen, die Kritiken verfassen können
* Admins der Website, die neue Filme hinzufügen können

Überlegen Sie, welche typischen Use Cases es für die genannten Aktoren in einer Online Filmebewertung gibt. Das Gesamtsystem soll aus mind. 10 Use Cases bestehen.

Folgendes soll erstellt werden:

1. Lösungsidee
2. Use Case Diagramm

# Abstraktionen in der realen Welt 2 Punkte

Finden oder erstellen Sie 6 verschiedene Abstraktionen. Dokumentieren Sie den Ausgangszustand der Idee/des Konzepts/… und dessen Abstraktion. Dokumentieren Sie ebenfalls welche Details entfernt und welche mit Absicht in der Abstraktion behalten wurden.

Folgendes soll erstellt werden:

1. Lösungsidee
2. Für jede Abstraktion:
   1. Ausgangszustand
   2. Abstraktion dessen
   3. Entfernte Details
   4. Nicht-entfernte Details

# Raute 8 Punkte

Erstellen Sie ein Programm, das eine Raute aus Sternen zeichnet. Der User kann als Input angeben, wie breit die Raute sein soll. Das Programm soll anhand dieses Inputs die Raute zeichnen. Die Breite, die angegeben werden kann, kann beliebig groß werden (z.B. auch weiter über 100 sein).

Beispiele:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Breite: 1 | Breite: 2 | Breite: 3 | Breite: 4 | … |
|  |  |  |  | … |

Folgendes soll erstellt werden:

1. Lösungsidee
2. Das Programm, das die Raute generiert
3. Testfälle

# Todo-Liste 10 Punkte

Erstellen Sie ein Programm für eine Todo-Liste.

Jedes Feature soll in eine Funktion ausgelagert werden. Folgende Features sollen unterstützt werden:

* Eingegebenen Text als neues Todo anlegen
* Todoliste mit offenen Todos ausgeben
* Bestimmtes Todo abschließen
* Todoliste mit abgeschlossenen Todos ausgeben
* Text von offenem Todo ändern

Folgendes soll erstellt werden:

1. Lösungsidee
2. Das Programm, das die Todo-Liste verwaltet
3. Testfälle für alle Features

**1. Onlinebibliothek**

**Lösung**

1. Lösungsidee

Ich habe zu Beginn die drei vorgegebenen Aktoren aufgezeichnet und die bereits vorgegebenen Use Cases hinzugefügt. Diese habe ich dann um weitere Use Cases ergänzt, die mir als sinnvoll erscheinen. Dabei habe ich mich vom Aufbau der IMDb Website inspirieren lassen.

1. Use Case Diagramm

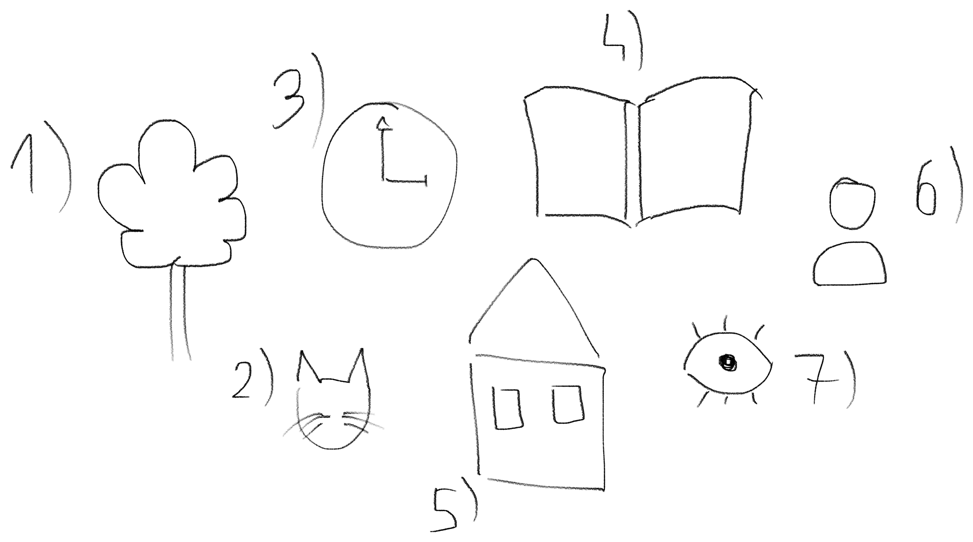
**2. Abstraktion in der realen Welt**

**Lösung**

1. Lösungsidee

Um 6 verschiedene Abstraktionen zu finden habe ich kurz meine Umgebung betrachtet und überlegt, welche der Objekte man vereinfacht darstellen könnte.

1. a.



b.

c. 1) Baum: Blätter, Äste d) Stamm, Baumkrone

2) Katze: Augen, Nase d) Kopf, Ohren, Schnurrhaare

3) Uhr: Ziffernblatt d) Rahmen, Zeiger

4) Buch: Wörter d) Seiten

5) Haus: Tür d) Dach, Fenster

6) Person: Gesicht, Haare, Hals d) Kopf, Oberkörper

7) Auge: Lid, Iris, Augenbraue d) Wimpern, Pupille

**3. Raute**

**Lösung**

1. Lösungsidee

Der User muss zuerst die gewünschte Breite der Raute eingeben. Danach wird überprüft, ob die angegebene Breite eine gerade Zahl ist, wenn ja wird die Zahl um 1 erhöht, sonst wäre die Raute nicht symmetrisch. Um die Raute zu zeichnen habe ich zwei for-Schleifen gewählt, jeweils eine für den oberen und den unteren Teil. Bei der ersten Schleife werden die Sterne im 2er Schritt erhöht bis zur jeweiligen Breite und bei der zweiten Schleife verringern diese sich so lange bis die Anzahl wieder 1 ist.

1. Implementierung

Ein Bild, das Text, Screenshot enthält.

Automatisch generierte Beschreibung

1. Testfälle

Testfall 1

Input:

width = 6

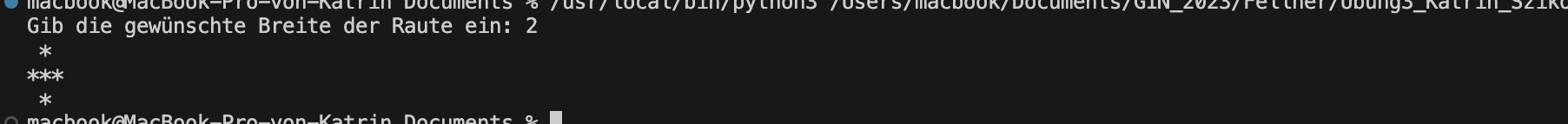
Ein Bild, das Text, Schrift, Screenshot enthält.

Automatisch generierte Beschreibung

Testfall 2

Input:

width = 2



Testfall 3

Input:

width = 9

Ein Bild, das Text, Screenshot enthält.

Automatisch generierte Beschreibung

Testfall 4

Input:

width = 17

Ein Bild, das Text, Screenshot, Schwarz, Design enthält.

Automatisch generierte Beschreibung

Testfall 5

Input:

Ein Bild, das Text, Screenshot enthält.

Automatisch generierte Beschreibungwidth = 25

**4. Todo-Liste**

**Lösung**

1. Lösungsidee

Das Programm für die To Do Liste beginnt mit einer Funktion für ein Menü, mit welchem die verschiedenen Features, anhand ihrer Nummer ausgewählt werden können. Für jedes einzelne Feature habe ich folgend eine eigene Funktion erstellt. Die To Do Aufgabe, welche der User hinzufügt, werden im Array « myTasks » gespeichert. Werden Aufgaben als erledigt markiert, wandern sie in den Array « tasksDone ». Diese können, genauso wie die zu erledigen Aufgaben, auf Verlangen abgerufen werden.

1. Ein Bild, das Text, Screenshot enthält.

   Automatisch generierte BeschreibungImplementierung

Ein Bild, das Text, Screenshot, Software, Multimedia-Software enthält.

Automatisch generierte Beschreibung

1. Testfälle

Testfall 1

Input:

choice = 1

Ein Bild, das Text, Screenshot, Schrift, Design enthält.

Automatisch generierte Beschreibungtask = Wäsche waschen, Hausübungen erledigen, Staubsaugen

Testfall 2

Input:

 choice = 2

Testfall 3

Input:

choice = 3

Ein Bild, das Text, Screenshot, Schrift, Schwarz enthält.

Automatisch generierte BeschreibungtaskNumber = 2

Testfall 4

Input:

choice = 2

Ein Bild, das Text, Screenshot, Schwarz, Schrift enthält.

Automatisch generierte Beschreibung

Testfall 5

Input:

Ein Bild, das Text, Screenshot, Schwarz, Schrift enthält.

Automatisch generierte Beschreibungchoice = 4

Testfall 6

Input:

choice = 5

taskNumber = 1

newText = Wäsche aufhängen

Ein Bild, das Text, Screenshot, Schrift, Schwarz enthält.

Automatisch generierte Beschreibung

Testfall 7

Input:

Ein Bild, das Text, Screenshot, Schrift, Schwarz enthält.

Automatisch generierte Beschreibungchoice = 2