WS 23 Übung 3 GIN1IL_T1

Abgabefrist: 10.11 23:55 **Erreichte Punkte: _____**

Name: Katrin Szikora Zeitaufwand in h: 8 h

Beachten Sie die Abgabekriterien! (siehe LVA Übersicht)

1) 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:

a) Lösungsidee
b) Use Case Diagramm

2) 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.

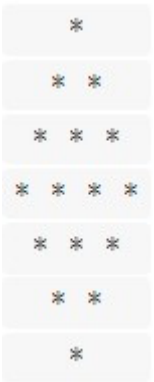
Folgendes soll erstellt werden:

a) Lösungsidee
b) Für jede Abstraktion:
    a. Ausgangszustand
    b. Abstraktion dessen
    c. Entfernte Details
    d. Nicht-entfernte Details

3) 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:

a) Lösungsidee
b) Das Programm, das die Raute generiert
c) Testfälle

4) 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:

a) Lösungsidee
b) Das Programm, das die Todo-Liste verwaltet
c) Testfälle für alle Features

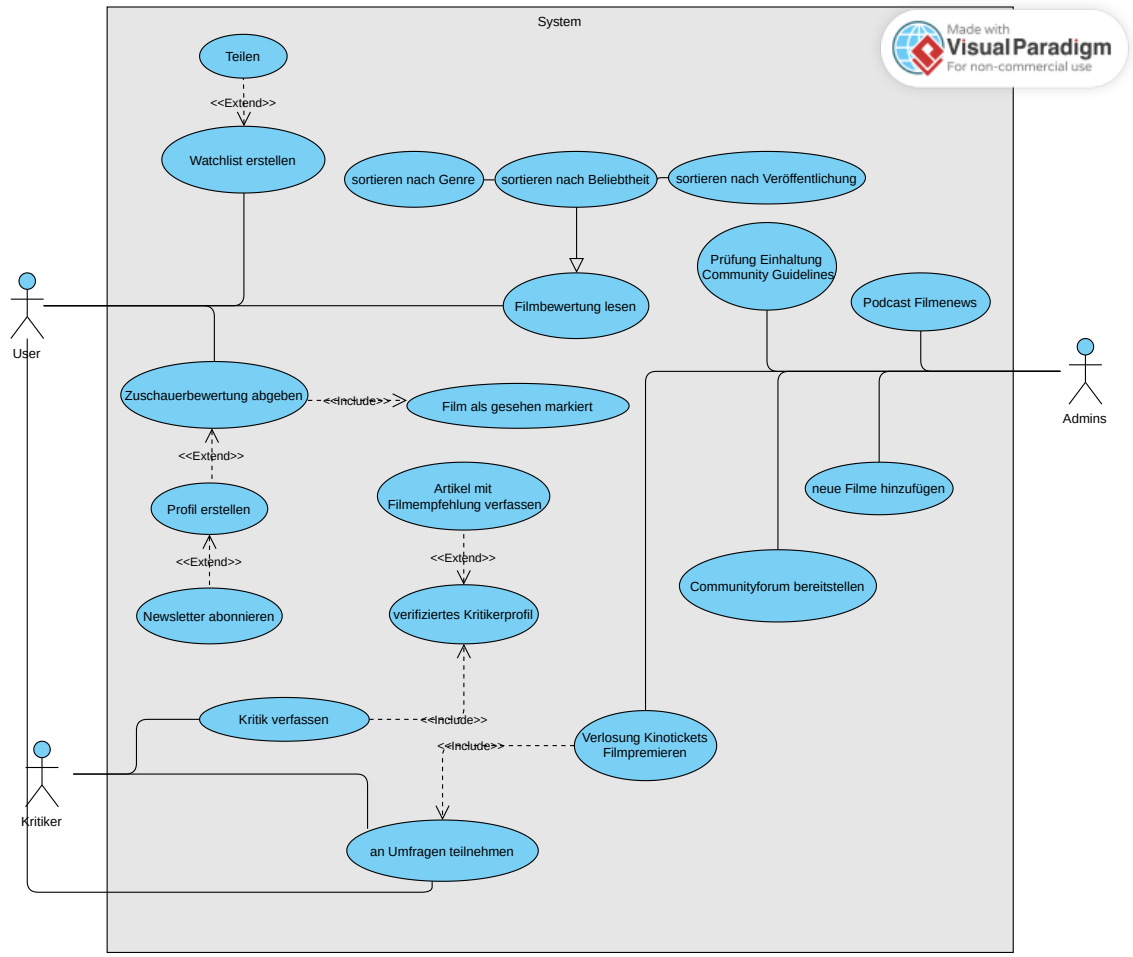
1. Onlinebibliothek

Lösung

a) 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.

b) Use Case Diagramm



2. Abstraktion in der realen Welt

Lösung

a) Lösungsidee

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

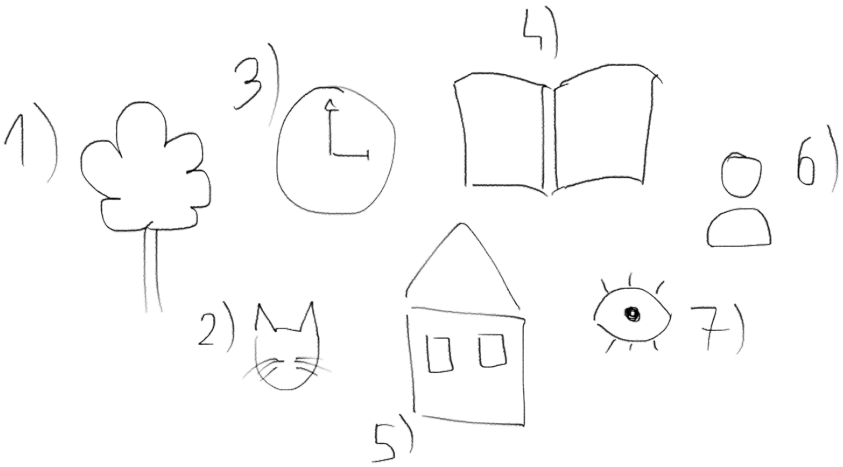
b) a.







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

a) 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.

b) Implementierung

```python
width = int(input("Gib die gewünschte Breite der Raute ein: "))

if width % 2 == 0: # % = modulo operator, berechnet Rest einer Division
    width += 1  # Breite muss ungerade sein, damit die Raute symmetrisch ist

# obere Hälfte der Raute
for i in range(1, width + 1, 2):  # Parameter range-Funktion = start, stop, step
    spaces = " " * int((width - i) / 2)
    stars = "*" * i
    print(spaces + stars)

# untere Hälfte der Raute
for i in range(width - 2, 0, -2):
    spaces = " " * int((width - i) / 2)
    stars = "*" * i
    print(spaces + stars)

```

c) Testfälle

Testfall 1

Input:
width = 6

```
macbook@MacBook-Pro-von-Katrin Documents % /usr/local/bin/python3 /Users/macbook
/Documents/GIN_2023/Fellner/Übung3_Katrin_Szikora/3b_Raute.py
Gib die gewünschte Breite der Raute ein: 6
   *
  ***
 *****
*******
 *****
  ***
   *
macbook@MacBook-Pro-von-Katrin Documents %
```

Testfall 2

Input:
width = 2

```
Gib die gewünschte Breite der Raute ein: 2
 *
***
 *
```

Testfall 3

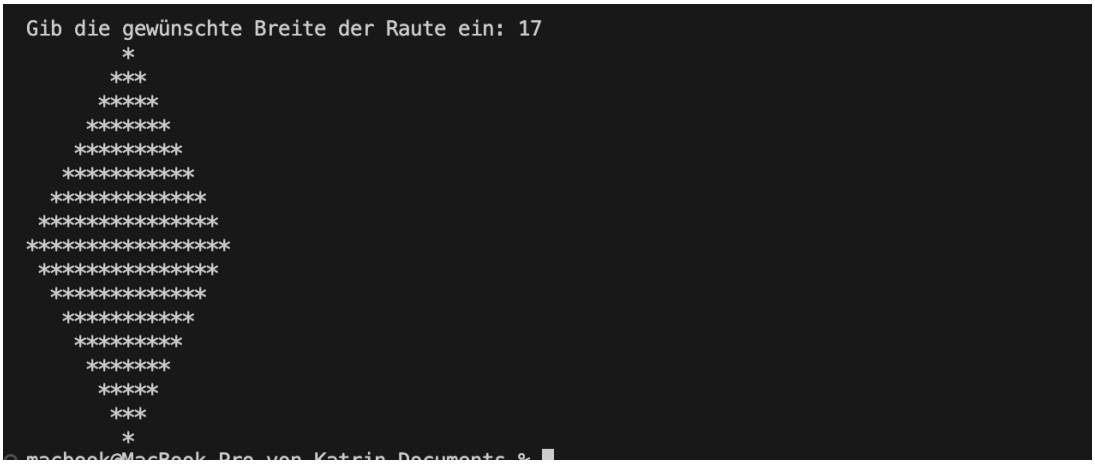
Input:

width = 9

Testfall 4

Input:

width = 17



Testfall 5

Input:

width = 25

4. Todo-Liste

Lösung

a) 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.

b) Implementierung

```python
myTasks = []
tasksDone = []

def menu():
    print("Meine To Do Liste")
    print("1. Aufgabe hinzufügen")
    print("2. Zu erledigende Aufgaben ansehen")
    print("3. Aufgabe erledigen")
    print("4. Erledigte Aufgaben ansehen")
    print("5. Text Aufgabe ändern")

def addTask():
    task = input("Gib eine neue Aufgabe ein. ")
    myTasks.append(task)
    print("Deine Aufgabe wurde erfolgreich hinzugefügt.")

def viewTask():
    print("Meine To Do's")
    if len(myTasks) == 0:
        print("Noch keine Aufgaben vorhanden. Füge zuerst welche hinzu.")
    else:
        for index, task in enumerate(myTasks, 1):   # Tasks werden beginnend mit 1 nummeriert
            print(f"{index}. {task}")

def markAsCompleted():
    taskNumber = int(input("Gib die Nummer der Aufgabe ein, um sie als erledigt zu markieren. "))-1     # -1, da Python bei 0 zu zählen beginnt
    if 0 <= taskNumber <= len(myTasks):
        completed_tasks = myTasks.pop(taskNumber)   # pop anstatt remove, da Task gespeichert und zu anderer Liste hinzugefügt
        tasksDone.append(completed_tasks)
        print("Die Aufgabe ist als erledigt gekennzeichnet.")
    else:
        print("Gib eine gültige Aufgabennummer ein.")

```

```python
def showCompletedTasks():
    print("Erledigte Aufgaben")
    if len(tasksDone) == 0:
        print("Noch keine Aufgaben erledigt.")
    else:
        for index, task in enumerate(tasksDone, 1):
            print(f"{index}. {task}")

def changeTaskText():
    taskNumber = int(input("Gib die Nummer der Aufgabe ein, um den Text zu ändern. ")) - 1
    if 0 <= taskNumber < len(myTasks):
        new_text = input("Gib den neuen Text für die Aufgabe ein. ")
        myTasks[taskNumber] = new_text
        print("Der Text der Aufgabe wurde erfolgreich geändert.")
    else:
        print("Gib eine gültige Aufgabennummer ein. ")


def main():
    while True:
        menu()
        choice = input("Gib deine Auswahl ein (1-5). ")
        if choice == "1":
            addTask()
        elif choice == "2":
            viewTask()
        elif choice == "3":
            markAsCompleted()
        elif choice == "4":
            showCompletedTasks()
        elif choice == "5":
            changeTaskText()

if __name__ == "__main__":
    main()
```

c) Testfälle

Testfall 1

Input:

choice = 1

task = Wäsche waschen, Hausübungen erledigen, Staubsaugen

```
Meine To Do Liste
1. Aufgabe hinzufügen
2. Zu erledigende Aufgaben ansehen
3. Aufgabe erledigen
4. Erledigte Aufgaben ansehen
5. Text Aufgabe ändern
Gib deine Auswahl ein (1-5). 1
Gib eine neue Aufgabe ein. Wäsche waschen
Deine Aufgabe wurde erfolgreich hinzugefügt.
Meine To Do Liste
1. Aufgabe hinzufügen
2. Zu erledigende Aufgaben ansehen
3. Aufgabe erledigen
4. Erledigte Aufgaben ansehen
5. Text Aufgabe ändern
Gib deine Auswahl ein (1-5). 1
Gib eine neue Aufgabe ein. Hausübungen erledigen
Deine Aufgabe wurde erfolgreich hinzugefügt.
Meine To Do Liste
1. Aufgabe hinzufügen
2. Zu erledigende Aufgaben ansehen
3. Aufgabe erledigen
4. Erledigte Aufgaben ansehen
5. Text Aufgabe ändern
Gib deine Auswahl ein (1-5). 1
Gib eine neue Aufgabe ein. Staubsaugen
Deine Aufgabe wurde erfolgreich hinzugefügt
```

Testfall 2

Input:

choice = 2

```
5. Text Aufgabe ändern
Gib deine Auswahl ein (1–5). 2
Meine To Do's
1. Wäsche waschen
2. Hausübungen erledigen
3. Staubsaugen
Meine To Do Liste
1. Aufgabe hinzufügen
```

Testfall 3

Input:

choice = 3

taskNumber = 2

```
Meine To Do Liste
1. Aufgabe hinzufügen
2. Zu erledigende Aufgaben ansehen
3. Aufgabe erledigen
4. Erledigte Aufgaben ansehen
5. Text Aufgabe ändern
Gib deine Auswahl ein (1–5). 3
Gib die Nummer der Aufgabe ein, um sie als erledigt zu markieren. 2
Die Aufgabe ist als erledigt gekennzeichnet.
Meine To Do Liste
```

Testfall 4

Input:

choice = 2

```
Die Aufgabe ist als erledigt gekennzeichnet.
Meine To Do Liste
1. Aufgabe hinzufügen
2. Zu erledigende Aufgaben ansehen
3. Aufgabe erledigen
4. Erledigte Aufgaben ansehen
5. Text Aufgabe ändern
Gib deine Auswahl ein (1–5). 2
Meine To Do's
1. Wäsche waschen
2. Staubsaugen
Meine To Do Liste
```

Testfall 5

Input:

choice = 4

```
Meine To Do Liste
1. Aufgabe hinzufügen
2. Zu erledigende Aufgaben ansehen
3. Aufgabe erledigen
4. Erledigte Aufgaben ansehen
5. Text Aufgabe ändern
Gib deine Auswahl ein (1–5). 4
Erledigte Aufgaben
1. Hausübungen erledigen
Meine To Do Liste
```

Testfall 6

Input:

choice = 5

taskNumber = 1

newText = Wäsche aufhängen

```
Meine To Do Liste
1. Aufgabe hinzufügen
2. Zu erledigende Aufgaben ansehen
3. Aufgabe erledigen
4. Erledigte Aufgaben ansehen
5. Text Aufgabe ändern
Gib deine Auswahl ein (1-5). 5
Gib die Nummer der Aufgabe ein, um den Text zu ändern. 1
Gib den neuen Text für die Aufgabe ein. Wäsche aufhängen
Der Text der Aufgabe wurde erfolgreich geändert.
Meine To Do Liste
```

Testfall 7

Input:

choice = 2

```
Meine To Do Liste
1. Aufgabe hinzufügen
2. Zu erledigende Aufgaben ansehen
3. Aufgabe erledigen
4. Erledigte Aufgaben ansehen
5. Text Aufgabe ändern
Gib deine Auswahl ein (1-5). 2
Meine To Do's
1. Wäsche aufhängen
2. Staubsaugen
Meine To Do Liste
```