

Informatik I: Einführung in die Programmierung

Prof. Dr. Peter Thiemann
Dr. Daniel Büscher, Hannes Saffrich
Wintersemester 2019

Universität Freiburg
Institut für Informatik

Übungsblatt 7

Abgabe: Montag, 9.12.2019, 9:00 Uhr morgens

Erinnerung. Nicht-ausführbare Programme werden mit 0 Punkten bewertet. Denken Sie also daran Ihr Programm vor der Abgabe nochmals auszuführen.

Aufgabe 7.1 (Filme; Datei: `movies.py`; Punkte: 3+3+4+4+4)

- (a) Implementieren Sie eine Klasse `Movie`, welche als Attribute den Titel `title` (str), das Erscheinungsjahr `year` (int), sowie die durchschnittliche Nutzerbewertung `rating` (float) besitzt. Beim Erzeugen eines `Movie`-Objekts sollen Titel, Erscheinungsjahr und Bewertung als Argumente übergeben werden. Hierzu ist es notwendig, eine `__init__` Funktion innerhalb der `class`-Anweisung zu definieren (siehe Vorlesung). Beispiel:

```
>>> m = Movie("Groundhog Day", 1993, 8.0)
>>> m.title, m.year, m.rating
('Groundhog Day', 1993, 8.0)
```

- (b) Implementieren Sie eine Funktion `movie_str(movie: Movie) -> str`, welche ein `Movie`-Objekt als Argument erhält und es, analog zur Vorlesung, in einen informativen String umwandelt. Der String soll dabei die folgende Form haben:

```
>>> movie_str(m)
"Movie('Groundhog Day', 1993, 8.0)"
```

- (c) Implementieren Sie eine Funktion `avg_score(movies: list) -> float`, welche eine Liste von `Movie`-Objekten als Argument erhält und die durchschnittliche Bewertung der enthaltenen `Movie`-Objekte zurückgibt. Wir nehmen hierzu an, dass die übergebene Liste mindestens ein Element enthält.

```
>>> movies = [
    Movie("Groundhog Day", 1993, 8.0),
    Movie('The Life Aquatic with Steve Zissou', 2004, 7.3),
    Movie('Tootsie', 1982, 7.4)]
>>> abs(avg_score(movies) - 7.566666) < 1e-4
True
```

- (d) Implementieren Sie eine Klasse `Actor` zur Repräsentation eines Schauspielers. `Actor`-Objekte besitzen die Attribute: `firstname` (str), `lastname` (str) und `movies` (Liste von `Movie`-Objekten). Beim Erzeugen einer Instanz von `Actor` sollen die Attribute als Argumente übergeben werden.

```
>>> bill = Actor('Bill', 'Murray', [Movie('Tootsie', 1982, 7.4)])
```

```
>>> bill.firstname, bill.lastname, len(bill.movies)
('Bill', 'Murray', 1)
```

- (e) Implementieren Sie eine Funktion `worst_actor(actors: list) -> Actor`, welche eine Liste von `Actor`-Objekten als Argument erhält und denjenigen Schauspieler zurückgibt, dessen Filme die niedrigste durchschnittliche Bewertung haben. Wir nehmen an, dass die Schauspielerliste, sowie die Film listen der einzelnen Schauspieler jeweils mindestens ein Element enthalten. Beispiel:

```
>>> actors = [
    Actor('Bill', 'Murray', [
        Movie('Groundhog Day', 1993, 8.0),
        Movie('The Life Aquatic with Steve Zissou', 2004, 7.3),
        Movie('Tootsie', 1982, 7.4)]),
    Actor('Steven', 'Seagal', [
        Movie('Black Dawn', 2005, 3.9),
        Movie('Exit Wounds', 2001, 5.5),
        Movie('The Patriot', 1998, 4.1)]),
    Actor('Ben', 'Kingsley', [
        Movie('Sexy Beast', 2000, 7.3),
        Movie('Lucky Number Slevin', 2006, 7.8)]]
>>> worst_actor(actors).firstname
'Steven'
```

Hinweis: Verwenden sie Ihre `avg_score` Funktion aus Aufgabenteil (c) wieder.

Aufgabe 7.2 (Erfahrungen; Datei: `erfahrungen.txt`; Punkte: 2)

Legen Sie im Unterverzeichnis `sheet07` eine Textdatei `erfahrungen.txt` an. Notieren Sie in dieser Datei kurz Ihre Erfahrungen beim Bearbeiten der Übungsaufgaben (Probleme, Bezug zur Vorlesung, Interessantes, benötigter Zeitaufwand, etc.).