

Dokumentation der Abschlussprüfung im Ausbildungsberuf
„Mathematisch-Technische*r Softwareentwickler*in“,
Prüfungsbereich: „Entwicklung eines Softwaresystems“

Thema Pending

Patrick Gustav Blaneck (20000)

IT Center RWTH Aachen University

Programmiersprache: Python



27. November 2022

Eidesstattliche Erklärung

Hiermit versichere ich, dass das vorliegende Prüfprodukt zum Thema

Thema Pending

von mir selbstständig erstellt wurde. Die als Arbeitshilfe genutzten Unterlagen sind in der Arbeit vollständig aufgeführt. Ich versichere, dass der vorgelegte Ausdruck mit dem Inhalt der von mir erstellten digitalen Version identisch ist. Weder ganz noch in Teilen wurde die Arbeit bereits als Prüfungsleistung vorgelegt. Mir ist bewusst, dass jedes Zuwiederhandeln als Täuschungsversuch zu gelten hat, der die Anerkennung des Prüfprodukts als Prüfungsleistung ausschließt.

Name: Patrick Gustav Blaneck

Aachen, den 27. November 2022

Unterschrift der/des Auszubildenden

Inhaltsverzeichnis

1	Aufgabenanalyse	4
1.1	Eingabe	4
1.2	Ausgabe	4
2	Verfahrensbeschreibung	6
2.1	Datenstrukturen	6
2.2	Algorithmen	6
3	Programmbeschreibung	6
3.1	Klassen und Schnittstellen	6
3.2	Algorithmen	7
3.3	Aufrufhierarchie	7
4	Testing und Laufzeitanalyse	8
4.1	Testing	8
4.2	Laufzeitanalyse	9
5	Abweichungen	10
6	Zusammenfassung und Ausblick	10
6.1	Zusammenfassung	10
6.2	Ausblick	10
Anhang	11
A	Benutzeranleitung	11
A.1	Installation	11
A.2	Benutzung	11
B	Entwicklerdokumentation	11
C	Hilfsmittel	12

1 Aufgabenanalyse

Im Rahmen dieses Softwareprojekts soll ein Programm erstellt werden, welches aus ... möglichst effizient ... erstellt.

Dabei soll das Programm insbesondere

Die ... werden aus einer Datei eingelesen und die ... werden in einer anderen Datei ausgegeben.

Die besondere Schwierigkeit bei diesem Projekt liegt darin, dass

Als generelles Vorgehen kommt ... in Frage.

1.1 Eingabe

Format

Für die Eingabe gilt, dass Ein Beispiel für eine Eingabedatei ist in Abbildung 1 zu sehen.

```
1 0 10 15 20
2 5 0 9 10
3 6 13 0 12
4 8 8 9 0
```

Abbildung 1: Beispiel für eine Eingabedatei

Dabei gilt

Fehlerquellen und -behebung

Die offensichtlichsten Fehlerquellen sind:

- ...
- ...
- ...

Um diese Fehlerquellen zu vermeiden, kann

1.2 Ausgabe

Format

Für die Ausgabe gilt, dass Ein Beispiel für eine Ausgabedatei ist in Abbildung 2 zu sehen.

```

1 Startmatrix:
2 0 10 15 20
3 5 0 9 10
4 6 13 0 12
5 8 8 9 0
6 mit Bewertung: max( 29, 32) = 32
7 *****
8 naechster Ort: B
9 0 10 0 0
10 0 0 9 10
11 6 0 0 12
12 8 0 9 0
13 mit Bewertung: max( 33, 35) = 35
14 *****
15 [...]
16 *****
17 naechster Ort: D
18 0 10 0 0
19 0 0 0 10
20 6 0 0 0
21 0 0 9 0
22 mit Bewertung: max( 35, 35) = 35
23 *****
24 also von B nach D
25 *****
26 Rundreise:
27 von A
28 ueber B
29 ueber D
30 ueber C
31 nach A
32 Entfernung: 35

```

Abbildung 2: Beispiel für eine Ausgabedatei

In der Ausgabedatei sollen also

Fehlerquellen und -behebung

Die offensichtlichsten Fehlerquellen sind:

- ...
- ...
- ...

Um diese Fehlerquellen zu vermeiden, kann

2 Verfahrenbeschreibung

2.1 Datenstrukturen

Aus der Aufgabenanalyse ergibt sich, dass folgende Datenstrukturen benötigt werden:

- ...
- ...
- ...

Besonders nennenswert ist dabei ...

2.2 Algorithmen

Die genutzten Algorithmen unterteilen sich in zwei Kategorien:

- Verarbeitung der Eingabedatei
- Erstellung der Ausgabedatei

Verarbeitung der Eingabedatei

Die Eingabe wird wie folgt verarbeitet:

1. ...
2. ...
3. ...

Erstellung der Ausgabedatei

Die Ausgabe wird wie folgt erstellt:

1. ...
2. ...
3. ...

Besonders nennenswert ist hier ...

3 Programmbeschreibung

3.1 Klassen und Schnittstellen

Wie bereits in der Verfahrenbeschreibung beschrieben, werden folgende Datenstrukturen benötigt:

- ...
- ...
- ...

Die konkrete Implementierung der Datenstrukturen erfolgt wie in Abbildung 3 zu sehen.

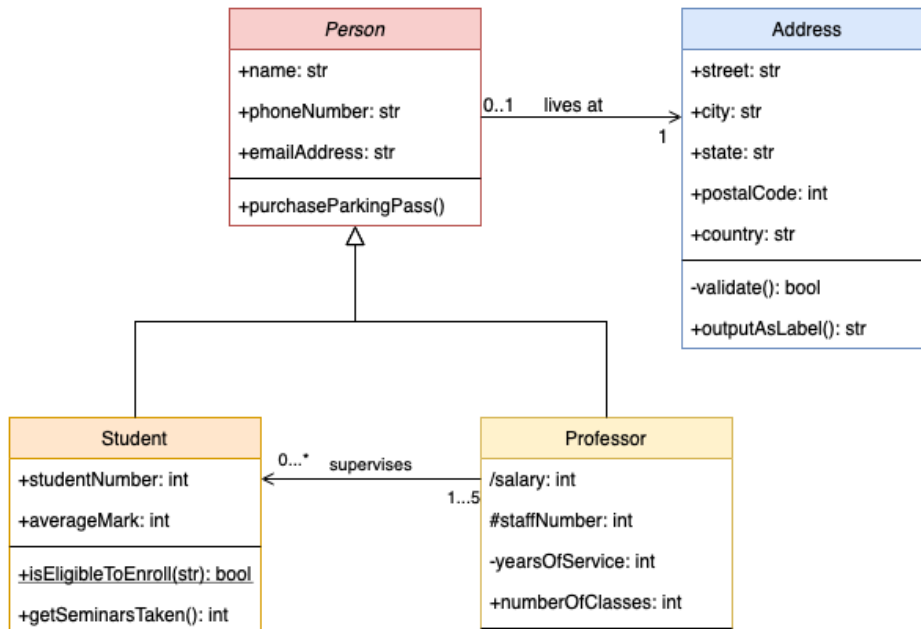


Abbildung 3: Beispiel für eine Datenstruktur

Besonders nennenswert ist dabei ...

3.2 Algorithmen

Mithilfe der Funktion ... wird die Eingabedatei verarbeitet und die Datenstrukturen gefüllt.

Die Ausgabedatei wird mit Hilfe der Funktion ... erstellt.

3.3 Aufrufhierarchie

Die Aufrufhierarchie ist beispielhaft in Abbildung 4 zu sehen.

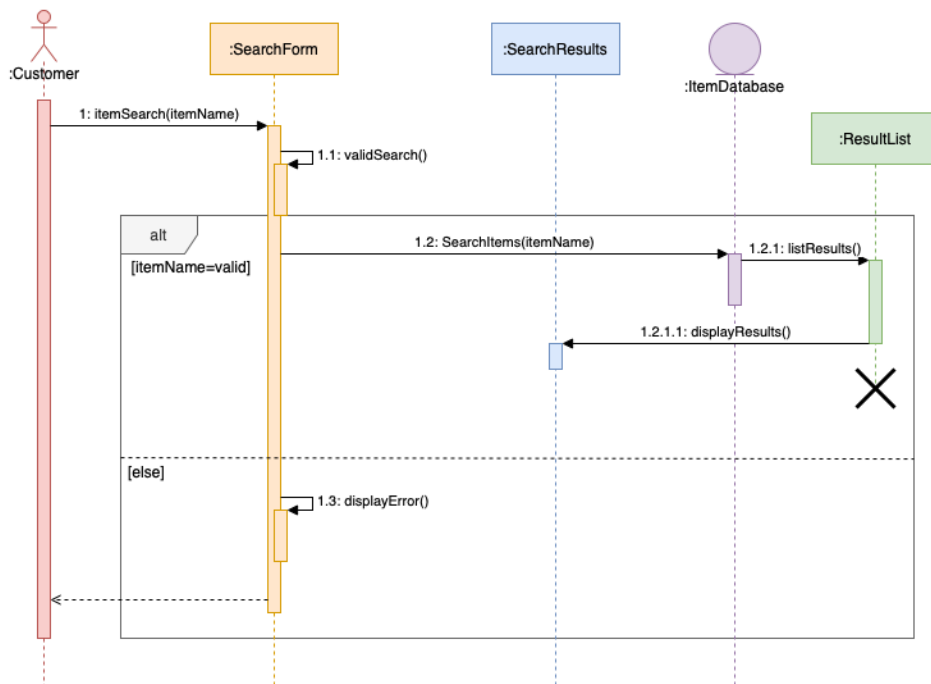


Abbildung 4: Beispiel für eine Aufrufhierarchie

4 Testing und Laufzeitanalyse

4.1 Testing

Teststrategie

Die Testfälle wurden nach der Methode ... ausgewählt. Die Testfälle sind in Tabelle 5 aufgelistet.¹

Testfall	Eingabe	Erwartete Ausgabe	Bemerkung
Normalfälle			
Grenzfälle			
Fehlerfälle			

Abbildung 5: Beispiel für eine Tabelle mit Testfällen

Beobachtungen

Es wurden folgende Beobachtungen gemacht:

- ...

¹Die Testfälle sind im Anhang ?? zu finden.

- ...
- ...

4.2 Laufzeitanalyse

Die Laufzeitanalyse wurde mit ... durchgeführt. Jeder Testfall wurde ... mal ausgeführt und im Anschluss die Laufzeiten gemittelt.

Für die einzelnen Testfälle aus Tabelle 5 ergeben sich die Laufzeiten in Tabelle 6.

Testfall	Median	Mittelwert	Standardabweichung
Normalfälle			
Grenzfälle			
Fehlerfälle			

Abbildung 6: Beispiel für eine Tabelle mit Laufzeiten

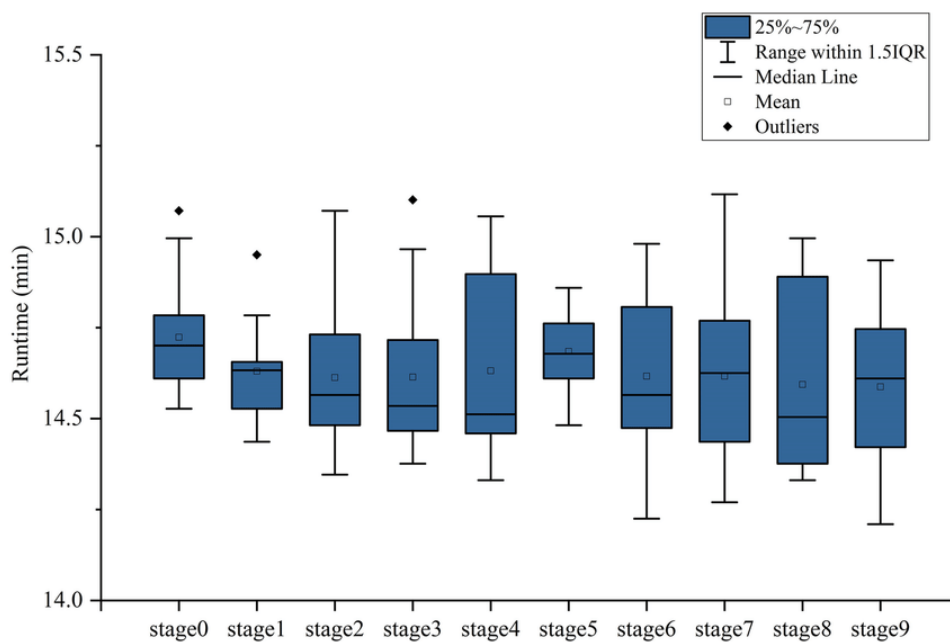


Abbildung 7: Beispiel für eine graphische Darstellung der Laufzeiten

Besonders auffällig ist ..., da der Algorithmus in $\mathcal{O}(n^m)$ läuft.

Für den praktischen Einsatz muss die Laufzeit nochmals verbessert werden. Siehe dazu Abschnitt 6.2.

5 Abweichungen

Im Laufe der Entwicklung des Programms sind einige Abweichungen vom ursprünglichen Konzept aufgetreten. Diese sind:

- ...
- ...
- ...

Besonders nennenswert ist dabei ..., da

6 Zusammenfassung und Ausblick

6.1 Zusammenfassung

In dieser Arbeit wurde ein Programm zur ... entwickelt.

Dabei wurde

Die ... wurden

6.2 Ausblick

In Zukunft könnte man zur Optimierung

Anhang

A Benutzeranleitung

A.1 Installation

Voraussetzungen

Vorgehen

A.2 Benutzung

Eingabe

Ausgabe

Kommandozeilenparameter

Beispiele

Fehlermeldungen

B Entwicklerdokumentation

Die Entwicklerdokumentation ist im Ordner `python/docs` zu finden.

Zum Generieren der Dokumentation wird das Tool `pdoc` verwendet. Dieses Tool generiert aus den Python-Dateien automatisch eine Dokumentation in HTML-Format mithilfe der Docstrings. Die Dokumentation kann mit dem Befehl in [8](#) generiert werden.²

```
1 pdoc project_name -o docs --docformat 'google'
```

Abbildung 8: Generieren der Dokumentation

²Vorausgesetzt, dass das Tool `pdoc` installiert ist und im Ordner `python` ausgeführt wird.

C Hilfsmittel

Die genutzten Hilfsmittel sind in 9 aufgeführt.

Typ	Tool	Hinweise
CPU	AMD Ryzen 5 3600	6x 3.60 GHz
RAM	32 GB	DDR4-3200
Betriebssystem	Ubuntu	20.04.1 LTS
IDE	Visual Studio Code	1.49.0
Compiler	Python	3.8.5
Debugger	Visual Studio Code	1.49.0
Testframework	unittest	3.8.5
Dokumentation	pdoc	0.9.2

Abbildung 9: Hilfsmittel