

Hochschule Osnabrück, Fakultät Ingenieurwissenschaften und Informatik

Prüfungsthemen im Modul *Informatik für Maschinenbau*, Sommersemester 2020

Prof. Dr.-Ing. Thomas Mechlinski

Thema SoSe_2020_10: Liftsteuerung

1 Allgemeines zur Prüfung

1.1 Prüfungsform und Bedeutung der Aufgabenstellung

Die Prüfung im Fach „Informatik für Maschinenbau“ erfolgt in Form einer Hausarbeit und einer Erläuterung, zu der Sie ggf. eingeladen werden.

Mit dieser Prüfungsaufgabe erhalten Sie eine Aufgabenstellung für Ihre Hausarbeit. Sie bezieht sich auf die Inhalte der Veranstaltung „Informatik für Maschinenbau“. Ihre Aufgabe ist die Lösung der Aufgabenstellung in Form eines Java-Programms (Implementierung) und einer schriftlichen Dokumentation Ihres Programmes.

Ihr Java-Programm muss die Anforderungen der Aufgabenstellung erfüllen sowie fehlerfrei funktionieren! Mit der eidesstattlichen Erklärung (s. u.) erklären Sie außerdem, dass Sie Ihre Arbeit **selbständig und ohne Hilfe von anderen Personen** angefertigt haben.

Falls sich herausstellt, dass Sie trotz der eidesstattlichen Erklärung fremde Hilfe in Anspruch genommen haben, wird die Arbeit als Täuschung mit „nicht bestanden“ bewertet. Weitergehende rechtliche Konsequenzen können folgen – Sie persönlich sind durch Ihre Unterschrift verantwortlich!

1.2 Schriftliche Ausarbeitung

Sie müssen Ihre Lösung in Form einer schriftlichen Ausarbeitung dokumentieren. Wissenschaftliche Ausarbeitungen werden nie in der „Ich-Form“ abgefasst. Schreiben Sie also nicht Sätze wie „Ich habe in dieser Arbeit folgende Aufgabe gelöst ...“ sondern etwa „In dieser Arbeit wird die Lösung einer Aufgabenstellung ...“.

Verwenden Sie diese Vorlage für Microsoft Word: Sie können die Datei [Vorlage_IfM_Hausarbeit.docx](#) in eines Ihrer Verzeichnisse speichern (Acrobat Reader: Rechtsklick auf den Link und „Eingebettete Datei auf Datenträger speichern ...“, Firefox: Doppelklick auf den Link, dann speichern, Microsoft Edge: Rechtsklick auf den Link und „Speichern unter“). Falls Sie mit LibreOffice/OpenOffice arbeiten wollen, verwenden Sie [diese Vorlage](#). Sie müssen Ihre Ausarbeitung mit einer dieser Vorlagen erstellen. Die Formatvorlagen in der Vorlage dürfen nicht verändert werden! Dies gilt insbesondere für die Schriftart und -größe, den Zeilenabstand, Absatzabstand und die sonstige Absatzformatierung sowie für die Seitenränder.

Die Ausarbeitung ist wesentlicher Bestandteil der Hausarbeit, eine mangelhafte Ausarbeitung führt daher zum Nichtbestehen der Prüfungsleistung. Ihre Ausarbeitung muss im durch die Vorlage vorgegebenen Format **mindestens 3, höchstens 6** von Ihnen **selbst verfasste** Seiten enthalten (Quelltextanteile, Titelblatt, Verzeichnisse, Aufgabenstellung und eidesstattliche Erklärung sowie übermäßiger Weißraum zählen nicht).

Die Ausarbeitung muss mindestens folgende Abschnitte enthalten:

- **Aufgabenstellung**

Kopie des Aufgabentexts. Den Aufgabentext für Ihre Aufgabe können Sie als [RTF-Datei](#) aus dieser Datei exportieren und dann in Ihr Dokument einbetten. Hinweis: Wegen der Konvertierung der Aufgabenstellung aus dem Textsatzsystem LaTeX nach RTF kann die RTF-Datei Fehler enthalten (insbesondere bei Bildreferenzen und bei Formeln). Sie müssen diese Fehler für Ihre Ausarbeitung ggf. manuell korrigieren.

- **Einleitung**

Kurzüberblick über die von Ihnen verstandenen Anforderungen und Ihre Lösung

■ **Ergebnisse**

In diesem Abschnitt sollen Sie Ihr Programm beschreiben. Dazu gehört eine Übersicht über den Programmablauf (ergänzt durch ein oder mehrere Nassi-Shneiderman-Diagramme) und eine detaillierte Beschreibung der einzelnen Schritte. Tipp: Die Nassi-Shneiderman-Diagramme lassen sich leicht mit dem Programm Structorizer (<http://structorizer.fisch.lu>) erstellen. Achten Sie in den Diagrammen darauf, die Algorithmen textuell zu beschreiben – es reicht nicht, Java-Code in die Diagramme zu schreiben.

Die Beschreibungen können sich auch auf Code-Ausschnitte (Java) beziehen. Es reicht jedoch nicht, nur den kommentierten Quelltext abzudrucken.

Ihre Arbeit wird besser lesbar und besser verständlich, wenn Sie den Text in Unterkapitel gliedern. Versuchen Sie, das Kapitel „Ergebnisse“ in etwa 3-5 Unterkapitel zu unterteilen. Beschreiben Sie KEINE Selbstverständlichkeiten (z. B. wie ein Java-Projekt in Eclipse angelegt wird, allgemeine Informationen über die Vorlesung usw.).

■ **Quellenverzeichnis**

Wie in wissenschaftlichen Arbeiten üblich, müssen Sie alle Quellen, die Sie zur Bearbeitung der Aufgabenstellung benutzt haben, in einem Quellenverzeichnis dokumentieren. Dazu zählen Bücher, aber auch Internetseiten und -videos, die Sie für die Lösung benötigt haben.

■ **Eidesstattliche Erklärung**

Die folgende Erklärung muss wortwörtlich enthalten sein (mit dem Ort, dem Datum und **Ihrer Unterschrift**). Fehlt die Erklärung oder ist sie nicht von Ihnen unterschrieben, ist die Arbeit „nicht bestanden“.

Name, Vorname, Matrikelnummer

Eidesstattliche Erklärung

Hiermit erkläre ich an Eides statt, dass ich die vorliegende Arbeit und das zugehörige Java-Programm selbständig und ohne fremde Hilfe angefertigt habe. Die aus fremden Quellen direkt oder indirekt übernommenen Gedanken sind als solche einzeln kenntlich gemacht. Es wurden keine anderen als die angegebenen Quellen und Hilfsmittel benutzt. Die Arbeit wurde bisher keiner anderen Prüfungsbehörde vorgelegt und auch nicht veröffentlicht.

Ort, Datum, Unterschrift

1.3 Abgabe Ihrer Hausarbeit

Ihre Hausarbeit geben Sie wie folgt ab:

- Die schriftliche Ausarbeitung als **ausgedrucktes** Dokument (möglichst gebunden, mindestens ordentlich geheftet) **und**
- Ihre Dateien im OSCA-Abgabebereich zu dieser Veranstaltung. Folgende Dateien müssen Sie im OSCA-Abgabebereich ablegen:
 1. Ihre **Java-Quelltextdatei(en)** (nur Dateien mit der Endung .java).
 2. Die **schriftliche Ausarbeitung als PDF-Datei**. Eine Abgabe als Word-Datei führt zu Notenverschlechterung.
 3. Eventuell erforderliche Input-Dateien (z. B. CSV- oder Textdateien)
 4. Geben Sie Ihre Dateien **NICHT** als ZIP- oder sonstiges Archiv ab!

Die ausgedruckte Dokumentation und die Dateien im OSCA-Abgabebereich gelten als Prüfungsunterlagen. Daher reicht eine E-Mail mit Ihrem Programm und Ihrer Ausarbeitung als Datei **nicht** aus, Sie würden den Leistungsnachweis nicht bestehen.

Im OSCA-Abgabebereich können Sie Ihre eigenen bereits hochgeladenen Dateien sehen, aber nicht mehr verändern oder löschen. Die Version der Dateien in OSCA muss mit der ausgedruckten Version übereinstimmen. Wenn die Datei nicht mit dem ausgedruckten Dokument übereinstimmt, kann das ein

Nichtbestehen zur Folge haben. Nachträgliche Korrekturen der hochgeladenen Java-Dateien sind nicht möglich – überlegen Sie sich also gut, was Sie in den Abgabebereich hochladen.

Sie müssen Ihre eigenen Java-Quelltextdateien hochladen, aber keine sonstigen Dateien aus Eclipse! Im Normalfall werden Sie eine Java-Datei und eine PDF-Datei hochladen. Laden Sie NICHT die Vorgabedateien aus der Aufgabenstellung (z. B. IO.java, FileInput.java) hoch! Input-Dateien (z. B. csv-Dateien) sollten Sie nur hochladen, wenn Sie selbst neue Dateien erzeugt haben, die also nicht aus der Aufgabenstellung stammen.

1.4 Bewertung der Hausarbeit

Ihre Hausarbeit wird durch den Prüfer bewertet. Dabei wird sowohl die schriftliche Ausarbeitung als auch das Java-Programm bewertet. Die Bewertungskriterien wurden in der Vorlesung erläutert und sind im Vorlesungsskript nachzulesen.

1.5 Täuschungsversuche

Ein Prüfungsthema kann von mehreren Prüfungsteilnehmerinnen oder Prüfungsteilnehmern bearbeitet werden. Dabei wird vorausgesetzt, dass Sie Ihre eigene Lösung erarbeiten. Wie in Klausurarbeiten erhalten Sie bei einem Täuschungsversuch die Note „Nicht bestanden“.

Falls festgestellt wird, dass sich eingereichte Lösungen in signifikanter Weise ähneln bzw. dass Lösungen aus anderen Quellen kopiert wurden, wird die Prüfungsleistung für **alle am Täuschungsversuch beteiligten Prüfungsteilnehmerinnen und Prüfungsteilnehmer als „Nicht bestanden“** bewertet. Die Prüfungsleistung ist für Sie auch „Nicht Bestanden“, wenn Sie Ihre Lösung von jemand anderem erstellen lassen.

1.6 Termine

Die Prüfungsunterlagen müssen bis zum **27. Juli 2020, 12 Uhr mittags**, dem Prüfer ausgehändigt werden (Postfach Mechlinski im SI-Gebäude (1. Etage, hinter dem Durchgang zum Altbau rechts), oder direkt bei Frau Hammer in ihrem Büro).

1.7 Erläuterung

Zur Bewertung Ihrer Hausarbeit können Sie zusätzlich zu einer Erläuterung eingeladen werden. Sie erhalten in dem Fall die Einladung per E-Mail an Ihre Hochschul-E-Mail-Adresse. Wenn Sie dann nicht zur Erläuterung erscheinen, ist die Prüfung „nicht bestanden“.

Die Erläuterung besteht aus einer Vorstellung Ihres Java-Programms, in der Sie die Aufgabenstellung und den von Ihnen gewählten Lösungsansatz erklären müssen. Dabei werden Sie auch aufgefordert werden, bestimmte Java-Konstrukte lauffähig in einer Java-Entwicklungsumgebung selbst zu erstellen (ohne Hilfsmittel).

Wenn Sie nicht in der Lage sind, Ihr eigenes Programm zufriedenstellend zu erläutern oder einfache Java-Konstrukte zu programmieren, ist die Prüfung „nicht bestanden“. Dabei ist es unerheblich, ob das Programm selbst korrekt ist.

WICHTIG: wenn Sie sich in OSCA für die Prüfung angemeldet haben, dann aber doch nicht zur Prüfung antreten wollen (Rücktritt), müssen Sie persönlich im Studierendensekretariat Ihren Rücktritt von der Prüfung erklären. Falls Sie das nicht machen, zählt die Prüfung als „Nicht bestanden“.

2 Aufgabenstellung SoSe_2020_10: Liftsteuerung

Ihr Programm soll die Steuerung eines vertikalen Personenfahrstuhls (Lift) simulieren. Die Simulation soll auf Eingaben reagieren, die ein Benutzer macht. Das Verhalten des Lifts soll durch folgende Annahmen simuliert werden:

- Der Lift soll keine Türsteuerung haben
- Der Lift wartet mindestens 3 Sekunden, wenn er in einem Ziel-Stockwerk angekommen ist
- Die Anzahl der Stockwerke muss beim Start des Programms abgefragt werden. Dabei wird jeweils eine Zahl für das unterste und eine für das oberste abgefragt. Es können auch negative Zahlen eingegeben werden. Die maximale Stockwerksnummer ist 12, die minimale -3.
- Die Fahrzeit zwischen zwei aufeinanderfolgenden Stockwerken beträgt immer 1 Sekunde
- Der Lift gibt jeweils aus, wenn er ein bestimmtes Stockwerk als Ziel erreicht („X. Etage: Aussteigen bitte“) oder an ihm vorbeifährt („X. Etage“)
- Wenn der Lift ein bestimmtes Stockwerk als Endposition erreicht hat, kann der Benutzer jeweils Anforderungen für die nächsten Ziele machen.
 - Dabei können auch mehrere, unterschiedliche Ziele eingegeben werden, maximal 4 Ziele. Dabei zählen keine Eingaben für das aktuelle Stockwerk, diese sollen nicht entgegengenommen werden.
 - Der Abschluss der Benutzereingaben soll durch Eingabe der Zahl 99 signalisiert werden.
 - Zur Vereinfachung soll Ihre Steuerung nur dafür sorgen, dass als erstes entweder das unterste oder das oberste, gewünschte Stockwerk angefahren wird. Danach sollen die darüber bzw. darunter liegenden Ziele angefahren werden, am Ende wird das höchste bzw. niedrigste eingegebene Ziel erreicht. Zur Bestimmung der Richtung soll ausgewertet werden, wie der Fahrstuhl zu den angeforderten Zielen steht: je nachdem ob zuerst das oberste oder das unterste Ziel erreicht werden kann soll er zuerst aufwärts oder abwärts fahren und nach Erreichen des obersten bzw. untersten Ziels die darunter bzw. darüber liegenden Ziele anfahren.
 - Erst wenn alle Ziele abgearbeitet sind, sollen wieder neue Anforderungen entgegen genommen werden.

Beispiel für eine Simulationsfahrt:

```
1._Etage
Ihre_Eingaben_bitte
Eingabe:_2
Eingabe:_-1
Eingabe:_1
Fehler:_Lift_befindet_sich_bereits_im_1._Stockwerk
Eingabe:_4
Eingabe:_99
0._Etage
-1._Etage:_Aussteigen_bitte
0._Etage
1._Etage
2._Etage:_Aussteigen_bitte
3._Etage
4._Etage:_Aussteigen_bitte
Ihre_Eingaben_bitte
Eingabe:_12
...
```

Hinweise zur Programmierung

- Um eine bestimmte Zeit abzuwarten, können Sie die Methode `Thread.sleep (zeit)` einsetzen. Die Methode sorgt dafür, dass der Programmablauf für die als (long)-Parameter `zeit` übergebene Anzahl von Millisekunden wartet. Um die Methode einsetzen zu können, müssen Sie sie mit einem try-catch-Block einfassen.

- Die Fahrtziele sollten Sie in einer `ArrayList<Integer>` speichern. Sie können sie leicht aufsteigend bzw. absteigend sortieren. Das absteigende Sortieren funktioniert z. B. so:

```
ArrayList<Integer> a = new ArrayList<Integer>;  
Collections.sort (a, Collections.reverseOrder ());
```

Weitere Hinweise

- Ihr Programm muss mögliche Fehler (z. B. Formatfehler bei der Eingabe, Dateizugriffsfehler) abfangen und aussagekräftige Fehlermeldungen ausgeben.
- **Überprüfen Sie das Ergebnis, das Ihr Programm ausgibt!** Ihr Programm muss korrekte Ergebnisse liefern, damit die Prüfungsleistung mindestens als „Bestanden“ gewertet werden kann. Überprüfen heißt, dass Sie das Verhalten und die Ausgaben Ihres Programmes mit allen vorgebenen Anforderungen aus der Aufgabenstellung vergleichen. Dieser Vergleich muss auch in brauchbarer Form in Ihrer Ausarbeitung dokumentiert werden (z. B. mit Hilfe von Bildschirmkopien). Werden Anforderungen nicht erfüllt, werden Sie nicht bestehen!
- Zur Eingabe von Ganzzahlen können Sie die in der Vorlesung vorgestellte Methode `IO.readInt()` nutzen.