

Table of Contents

Laboreinheit: Algorithmen und Datenstrukturen – Aufgabenblatt 3	1
Aufgaben	1
Deadline und Punkte	3
Feedback	3
Literaturverzeichnis	4
Appendix A: Leitfaden für das Konsolenmenü	5
Appendix B: Abgabebedingungen	5
Appendix C: Bewertung und Verteilung der Klausurpunkte (KP)	7

Laboreinheit: Algorithmen und Datenstrukturen – Aufgabenblatt 3

Tom Kamberg-Buhrtz <buhrtz@htw-berlin.de>



Zum Lösen des Aufgabenblattes verwenden Sie bitte die **Standarddatentypen und -definition von JAVA**. `LinkedList<T>`, `ArrayList` usw. **dürfen nicht verwendet werden**. Es darf nur für die Verwaltung der `ICommands` eine `LinkedList<ICommand>` oder `HashMap<Integer, ICommand>` verwendet werden (siehe für weitere Informationen erstes Aufgabenblatt), dies soll Ihnen die Erstellung des Command-line Menüs erleichtern.

Aufgaben

1. Gegeben ist ein gerichteter Graph in mathematisch/algebraischer Darstellung. Stellen Sie diesen in allen Ihnen bekannten Darstellungsformen dar.

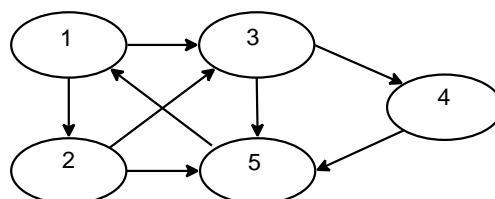
(3 Pkt.)

Gegeben:

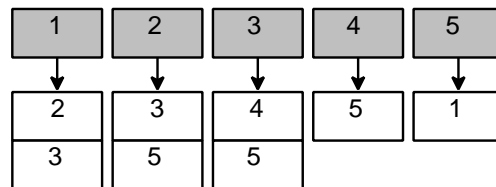
$$\begin{aligned} G &= \{V, \vec{X}\} \\ V &= \{1, 2, 3, 4, 5\} \\ \vec{X} &= \{(1, 2), (2, 3), (3, 4), (5, 1), (1, 3), (2, 5), (3, 5), (4, 5)\} \end{aligned}$$

Lösung:

- a. Grafische Darstellungsform



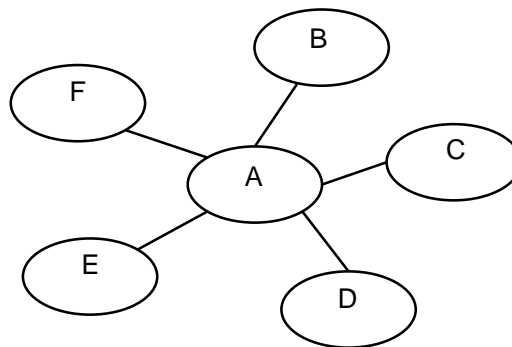
b. Adjazenzliste



c. Adjazenzmatrix

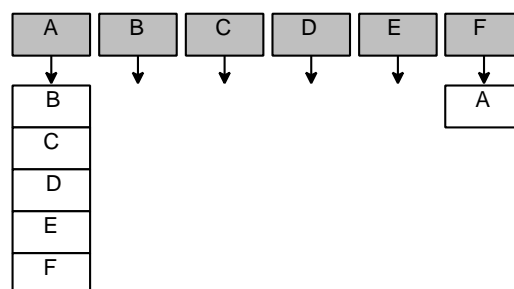
	1	2	3	4	5
1	0	1	1	0	0
2	0	0	1	0	1
3	0	0	0	1	1
4	0	0	0	0	1
5	1	0	0	0	0

2. Gegeben ist ein Graph in grafischer Darstellung. Stellen Sie diesen in allen Ihnen bekannten Darstellungsformen dar.



(3 Pkt.)

3. Gegeben ist die Adjazenzliste eines Graphen. Stellen Sie diesen in allen Ihnen bekannten Darstellungsformen dar.



(3 Pkt.)

4. Gegeben ist die Adjazenzmatrix eines Graphen. Stellen Sie diesen in allen Ihnen bekannten Darstellungsformen dar.



Die Zahlen sind hier die Gewichtungen der einzelnen Verbindungen!

(3 Pkt.)

	1	2	3	4	5	6
1	0	5	3	11	1	9
2	0	0	0	0	0	0
3	0	0	0	0	0	0
4	0	0	0	0	0	0
5	0	0	0	0	0	0
6	0	0	0	0	0	0

5. Implementieren Sie basierend auf der Datenstruktur Student einen Stack und ein ausführbares Programm um diesen Stack, bei dem man folgende Funktionen interaktiv in der Kommandozeile ausführen kann (Beispielsweise über ein Menü mit Zahleneingabe):

(8 Pkt.)

- Implementieren Sie die für einen Stack üblichen Funktionen **push** und **pop**, um ihrem Stack einen Studenten hinzuzufügen oder zu entnehmen.
- Implementieren Sie zusätzlich die Funktionen **peek**, **empty** und **clear** zum Stack.
- Implementieren Sie eine Funktion zur Ausgabe aller Elemente im Stack.
- Implementieren Sie eine Funktion zur Ausgabe der Anzahl von Elementen im Stack.

Deadline und Punkte

Table 1. Maximale Punkte und Deadline

Deadline:	05.12.2021 - 23:55
Maximale KP:	15 KP (Klausurpunkt)
Maximale Punktzahl:	20 Punkte == 100 % == 15 KP



Wird die volle Punktzahl beim Aufgabenblatt erreicht, dann haben Sie **100 %** erfüllt und somit **15 KP * 1.0 = 15 KP**. Haben Sie nur **50 %** erfüllt, dann berechnen sich die Klausurpunkte, wie folgt **15 KP * 0.5 = 7.5 KP** usw.

Feedback

Senden Sie mir gerne konstruktives Feedback zu den Aufgabenblättern und Dokumenten zu.



Verbesserungsvorschläge für dieses Aufgabenblatt können Sie gerne per Git-Patch oder schriftlich per E-Mail an mich senden. Git-Patches geben Ihnen die Möglichkeit, zusätzliche Punkte für ein Aufgabenblatt zu erhalten. Dafür müssen die Git-Patches atomar sein und im ZIP-Archive abgelegt, oder per E-Mail an mich gesendet werden.

Viel Spaß und Erfolg bei der Bearbeitung der Aufgabenblätter!

Erstellungsdatum und -zeit: 2021-10-10 - 23:49:30 +0200

Literaturverzeichnis

- [1] Thomas Ottmann and Peter Widmayer, Algorithmen und Datenstrukturen (German Edition), Spektrum Akademischer Verlag}, 2002, ISBN: 3827410290.
- [2] Sebastian Dworatschek, Grundlagen der Datenverarbeitung (de Gruyter Lehrbuch) (German Edition), de Gruyter, 1989, ISBN: 3110120259.
- [3] Robert Sedgewick, Algorithmen in C++ (German Edition), Pearson Studium, 2002, ISBN: 3827370264.
- [4] Ullenboom, Christian (Autor), Java ist auch eine Insel – Das Standardwerk für Programmierer. Über 1.000 Seiten Java-Wissen. Mit vielen Beispielen und Übungen, aktuell zu Java 14, 2020, ISBN: 9783836277372. <https://www.buchhandel.de/buch/Java-ist-auch-eine-Insel-9783836277372>
- [Java10] [Openbook: Java ist auch eine Insel](#)
- [Java11] [Java ist auch eine Insel - Kapitel Vererbung](#)
- [Java12] [Java ist auch eine Insel - Kapitel Vererbung](#)
- [Java13] [JAVA Interfaces](#)
- [Java14] [JUnit-Testframework](#)
- [Java15] [Java ist auch eine Insel - Generics<T>](#)
- [Gradle20] [Gradle](#)
- [Gradle21] [Gradle Installation](#)
- [Gradle22] [Gradle Tutorials and Guides](#)
- [Git-SCM] [Git](#)
- [Git-EN-v2-Book] [Pro Git Book - Englisch \(PDF etc.\)](#)
- [Git-DE-v2-Book] [Pro Git Book - Deutsch](#)
- [HTW70] [HTW Berlin - Projekteserver](#)
- [PlantUML-ClassDiagram] [Plant-UML-Klassendiagramme](#)
- [Asciidoctor] [Asciidoctor](#)
- [QuickAsciidoctor] [Asciidoctor - Quick Reference](#)
- [HTWMoodle] [HTW Berlin - Moodle](#)
- [DeepLTranslator] [AI Assistance for Language](#)
- [Eclipse Plugin for Gradle] [Buildship Plugin for Gradle](#)
- [GraphViz Pocket Reference] [GraphViz Pocket Reference](#)

Appendix A: Leitfaden für das Konsolenmenü

Die Reihenfolge und Zuordnung der Nummern (siehe unten) zu den verschiedenen Optionen bitte einhalten, um auf Grund der hohen Anzahl von Abgaben ein automatisiertes Skript zur Kontrolle einsetzen zu können.

Konsolenhauptmenü

Dieses Menü unterstützt die relevanten Hauptfunktionalitäten vom Stack für das Testen per Command Line Interface (CLI).

```
Console-Application: Exercise-3          <Vorname> <Name> <Matrikelnummer>
```

1. Push student to stack.
2. Pop student from stack.
3. Peek student.
4. Is stack empty.
5. Clear stack.
6. Print all students from stack.
7. Returns size of stack.
0. Exit.

Please enter a number for an option:

Eigenschaften des Studenten einlesen:

Bei der Auswahl der Option 1 werden die folgenden Informationen abgefragt und eingelesen:

```
Please enter prename: Sheldon
Please enter surname: Cooper
Please enter course number: 1
Please enter matriculation number: 5
```

Vielen Dank!

Appendix B: Abgabebedingungen

Voraussetzung, damit eine Bewertung vorgenommen wird!

Bei jeder Abgabe muss der Gradle-Wrapper `./gradlew` im Projektrootverzeichnis vorhanden sein und funktionieren, sonst wird keine Bewertung der Aufgabe vorgenommen. Es dürfen nur relative Pfade verwendet werden. Die folgenden Tasks müssen per Command-line im

Projektrootverzeichnis ausführbar sein:

```
./gradlew clean           # Löscht automatisch generierte Dateien.  
./gradlew build           # Programm kompilieren.  
./gradlew test            # Alle Unittests ausführen.  
./gradlew javadoc         # API Dokumentation erstellen.  
./gradlew run -q --console=plain # Programm per Console ausführen.  
./gradlew run             # Programm per Console ausführen.
```



Bitte in der `build.gradle` den Parameter `System.in` setzen, damit die Eingabe von Parametern `./gradlew run -q --console=plain` auf der Konsole per Gradle funktioniert.



Halten Sie bitte die Reihenfolge der Optionen und die Zuordnung der Nummern ein, damit automatische Skripte für die Kontrolle der Abgaben eingesetzt werden können.



Verwalten Sie den Source Code über die Versionsverwaltung Git als Übung für zukünftige Projekte in der Angewandten Informatik (AI).

Appendix C: Bewertung und Verteilung der Klausurpunkte (KP)

Wie wird die Bewertung in diesem Semester vorgenommen?

Es müssen folgende Aufgabenblätter in diesem Semester abgegeben werden. Sie können insgesamt 50 Klausurpunkte (KP) bei mir in der Laboreinheit erhalten. Sie müssen mindestens 25 Klausurpunkte (KP) bei mir erreichen, damit Sie an der Prüfung bei Prof. Sieck teilnehmen können.

Aufgabenblatt 1

15 Klausurpunkte - Themen: Einführung (Gradle, allgemeine Algorithmen, Anforderung an die Abgaben)

Aufgabenblatt 2

25 Klausurpunkte - Themen: Einfach und doppelt verkettete Liste, Sortierverfahren und Komplexitätsberechnung

Aufgabenblatt 3

15 Klausurpunkte - Themen: Adjazenzliste, Adjazenzmatrix, Graphen und ADTs

Aufgabenblatt 4

Entfällt

Aufgabenblatt 5

10 Klausurpunkte - Themen: Lineares Sondieren, Quadratisches Sondieren und Double Hashing

Aufgabenblatt 6

30 Klausurpunkte - Themen: Verlustfreie Kompressionsalgorithmen

Die Aufgabenblätter haben ein eigenes Punktesystem. Es sind keine Klausurpunkte (KP). Wenn Sie 100 % des Aufgabenblatts erfüllt haben, dann bekommen Sie die maximal möglichen Klausurpunkte pro Aufgabenblatt. Die maximal möglichen Klausurpunkte stehen auf dem jeweiligen Aufgabenblatt am Ende. Hier ein kurzes Rechenbeispiel der maximal möglichen Klausurpunkte pro Aufgabenblatt:

- Wenn Sie z.B. beim Aufgabenblatt 1 die volle Punktzahl (100%) erreichen, dann bekommen Sie die vollen 15 Klausurpunkte.
- Wenn Sie z.B. beim Aufgabenblatt 2 die Hälfte der angegebene Punktzahl (50%) erreichen, dann bekommen Sie 12,5 Klausurpunkte von 25 maximal möglichen Klausurpunkten.
- Wenn Sie z.B. beim Aufgabenblatt 3 die volle Punktzahl erreichen, dann bekommen Sie die 15 Klausurpunkte.

In Summe können Sie aus der Laboreinheit und dem seminaristischen Unterricht 100

Klausurpunkte erreichen. Bei mir können Sie genau die Hälfte der Klausurpunkte (also 50 KP) erreichen. Die restlichen 50 Klausurpunkte bekommen Sie von Prof. Sieck in einer mündlichen oder schriftlichen Prüfung. Wenn Sie alle Aufgabenblätter in der Laboreinheit mit 100 % absolvieren, dann haben Sie die maximal möglichen 50 Klausurpunkte (KP) bei mir aus der Laboreinheit erreicht.