

Führen Sie die folgenden Übungen an einem der Laborrechner oder Ihrem eigenen Rechner aus. Nutzen Sie die Eclipse-IDE für die Entwicklung mit der Programmiersprache JAVA.

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

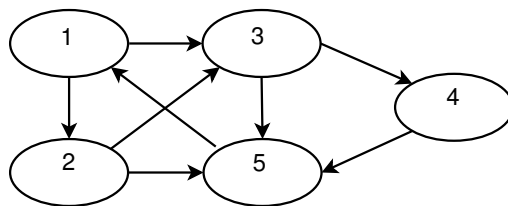
$$G = \{V, \vec{X}\}$$

$$V = \{1, 2, 3, 4, 5\}$$

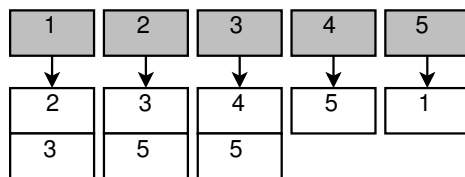
$$\vec{X} = \{(1, 2), (2, 3), (3, 4), (5, 1), (1, 3), (2, 5), (3, 5), (4, 5)\}$$

**Beispiellösung:**

**Grafische Darstellungsform**



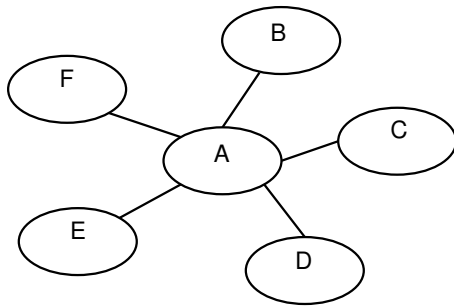
**Adjazenzliste**



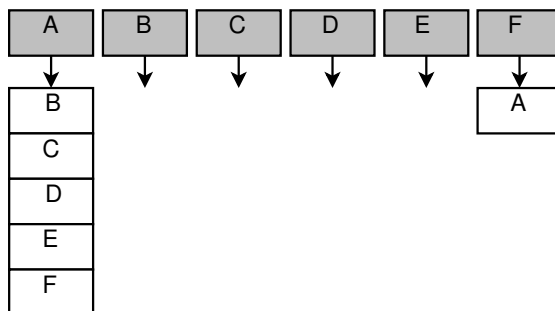
**Adjazenzmatrix**

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



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).

	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 aus Übung 2 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):
- Implementieren Sie die für einen Stack üblichen Funktionen **push** und **pop**.
  - Implementieren Sie Funktionalität, um ihrem Stack ein Element hinzuzufügen und zu löschen.
  - Implementieren Sie Funktionen zur Ausgabe der Elemente im Stack.
  - Implementieren Sie Funktionalität zur Ausgabe der Anzahl von Elementen im Stack.

**Hinweise:** Zum Lösen der Aufgabe verwenden Sie bitte die Standarddatentypen und -definition von JAVA. Stack, Queue aus der `java.util` dürfen nicht verwendet werden. Sollten Sie diese Übung abgeben wollen, geschieht das bitte in Form eines Archives mit einer Lösungsbeschreibung im PDF-Format (sowie Lösungen der Aufgaben, die nicht die Implementierung betreffen) und einem Ordner mit dokumentiertem Quelltext (Javadoc). Zusätzlich soll der Anwender die Möglichkeit haben die Anwendung per Console zu bedienen. Sollten Sie mit noch keinem Textverarbeitungsprogramm vertraut genug sein, sei Ihnen  $\LaTeX$  wärmstens ans Herz gelegt.

## Deadline und Punkte

Abgabetermin: 23.1.2017 - 23:55 Uhr.

Maximale Punktzahl: 2

## Wichtige Informationen zur Abgabe

**Makefile:** Es muss bei jeder Abgabe eine **Makefile** im Rootverzeichnis vorhanden sein. Ohne die Makefile wird **keine Bewertung der Aufgabe** vorgenommen. Es dürfen in dieser nur **relative Pfade** verwendet werden. Zusätzlich müssen in dieser die folgenden Targets definiert und ausführbar sein:

- **default:** Kompilierung der Java-Konsolenanwendung,
- **run-app:** Ausführung der Java-Konsolenanwendung,
- **run-test:** Für die Durchführung der JUnit-Tests (Jede Funktionalität soll gegen einen JUnit-Test getestet werden),
- **doc:** Für die Generierung der Quelltext-Dokumentationen,
- **clean:** Säubert das Arbeitsverzeichnis von den automatisch generierten Dateien (Binärdatei z.B. \*.class),

**Zusätzliche Literatur:** Im Folgenden sind Literaturverweise für das Lösen der Aufgabenblätter angegeben:

- Make<sup>1</sup>
- JUnit<sup>2</sup>
- Java OpenJDK<sup>3</sup>
- Openbook: Shell-Programmierung<sup>4</sup>
- Openbook: Java ist auch eine Insel<sup>5</sup>
- Pro Git Book - Englisch (PDF etc.)<sup>6</sup>
- Pro Git Book - Deutsch<sup>7</sup>
- HTW Berlin - Git-Server<sup>8</sup>

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

**Abschließende Informationen** Vergessen Sie bei der Entwicklung der Java-Konsolenanwendung die Benutzerfreundlichkeit nicht. Über produktives und sachliches Feedback zu den Aufgabenblättern und Übungen von Ihnen würde ich mich sehr freuen. Viel Spass und Erfolg bei der Entwicklung der Java-Konsolenanwendungen.

---

<sup>1</sup><https://www.gnu.org/software/make/>

<sup>2</sup><http://junit.org/>

<sup>3</sup><http://www.oracle.com/technetwork/java/javase/downloads/index.html>

<sup>4</sup>[http://openbook.rheinwerk-verlag.de/shell\\_programmierung/](http://openbook.rheinwerk-verlag.de/shell_programmierung/)

<sup>5</sup><http://openbook.rheinwerk-verlag.de/javainsel/>

<sup>6</sup><https://git-scm.com/book/en/v2>

<sup>7</sup><https://git-scm.com/book/de/v1>

<sup>8</sup><https://studi.f4.htw-berlin.de/www/>