

Algorithmen und Datenstrukturen II

Leistungsnachweis

Prof. Dr. Oliver Braun

Fakultät für Informatik und Mathematik
Hochschule München

Letzte Änderung: 16.03.2020 17:49

Lesen Sie sich unbedingt vor Bearbeitung des Blattes die [Informationen zu den Praktikumsaufgaben](#) durch und beachten Sie diese.

Termine

- **06.04.2020, 11:30 Uhr** — Themenfindung abgeschlossen
- **28.05.2020, 08:00 Uhr** — Vorstellung und Besprechung Zwischenstand und Release 0.9
- **22.06.2020, 08:00 Uhr** — Abschlußpräsentation und Release 1.0

Leistungsnachweis

Sie sind nur dann zur Teilnahme am Leistungsnachweis und der mündlichen Prüfung zugelassen, wenn Sie bei der ZPA-Vergabe einen Platz in dieser Lehrveranstaltung bekommen haben, d.h. wenn Sie im ZPA eingetragen sind. **Es gibt keine Ausnahme von dieser Regelung!**

Für den Leistungsnachweis, der Sie zur Teilnahme an der mündlichen Prüfung berechtigt, müssen Sie dieses Blatt in einer Gruppe erfolgreich bearbeiten. Bewertet wird jedoch die Leistung jeder/jedes Einzelnen.

1. Wenn Sie **keinen** Schein benötigen, greifen Sie einfach direkt auf das [Repository](#) zu.
2. Wenn Sie **einen Platz im ZPA zugewiesen bekommen haben** und einen Schein bekommen wollen, tun Sie folgendes:

- a) Treten Sie über den GitHub-Classroom-Link https://classroom.github.com/g/YPX_r0TQ dem **einzigen** Team 20ss bei.
- b) Füllen Sie im Moodle-Kurs zur Veranstaltung das Feedback *Zuordnung GitHub-Account* aus, damit ich weiß wer hinter welchem GitHub-Account steckt.

Über den folgenden Link können Sie sich in Gruppen zu maximal 4 Studierenden¹ zusammen finden:

<https://classroom.github.com/g/B0uDE0xp>

Der erste Studierende erzeugt eine neue Gruppe und bekommt ein Repository. Die weiteren bis zu drei Gruppenmitglieder können über den selben Link der Gruppe beitreten und damit auch auf das Repository zugreifen. Sie haben Adminrechte in Ihrem Repository, können also alles selbst konfigurieren, aber das Repository auch selbst löschen oder Teile zerstören!

Tools

Auf der [DevBox-VM](#) ist bereits [Go](#) sowie [Visual Studio Code](#) mit der [Go-Extension](#) installiert. Es lohnt sich aber auf jeden Fall immer auf die aktuellen Versionen upzugraden. Von JetBrains gibt es [GoLand](#), wenn Sie eine IDE à la IntelliJ IDEA o.ä. bevorzugen. Das müssten Sie sich, auch auf der DevBox, noch selbst installieren.

Themenfindung

Erzeugen Sie in **Ihrem** Repository ein [Issue](#) mit dem Titel **Vorschlag für Thema** in dem Sie Ihren Vorschlag beschreiben. Erst wenn das Thema von mir **approved** ist, können Sie es tatsächlich bearbeiten.

Damit ich es bearbeite, muss es außerdem mir (@obcode) zugewiesen werden **und** das Label **proposal**² haben. Außerdem müssen Sie Ihr Issue mit dem Thema als Text auf der Wiki-Seite [Proposals](#) verlinken. Dort sehen Sie was die anderen Gruppen machen wollen. Sollten zwei Themen identisch oder zu nah zusammen sein, können Sie sich versuchen entsprechend abgrenzen. Nachdem die Repositories **private** sind, können Sie z.B. über ein Issue in unserem gemeinsamen Repository kommunizieren. Wenn Sie wollen, können Sie Ihr Repository natürlich auch **public** machen.

Im Rahmen der Praktika und des Issues können wir dann über das Thema diskutieren. Sobald ich einverstanden bin, setze ich das Label **approved** und entferne mich selbst aus den **Assignees**. Ab dem Moment können Sie das Issue schließen und mit der Arbeit beginnen.

Spätester Termin für ein genehmigtes Thema: **06.04.2020, 16:45**

¹Es gibt keine Ausnahme. Wenn Sie mehr als 4 sind, müssen Sie sich aufteilen!

²Das Label **proposal** müssen Sie selbst erst erzeugen.

Bearbeitung

Bearbeiten Sie Ihr Projekt gemeinsam. Nutzen Sie zur Planung und Koordination [Project Boards](#) und [Issues](#). Nutzen Sie die Möglichkeiten unterstützend, weisen Sie sich Issues zu und **schließen** Sie sie, wenn Sie abgearbeitet sind, wieder.

Arbeiten Sie in der Gruppe nach dem [Git-Flow](#).

Beachten Sie <https://ob.cs.hm.edu/exercises>. Weitere nützliche Infos, wie z.B. das automatische Schließen von Issues über eine Commit-Message, finden Sie auf <https://help.github.com/>.

Sollten Sie Unterstützung von mir benötigen, erzeugen Sie ein Issue und weisen Sie es mir zu. Versehen Sie es mit dem passenden Label, z.B. `help wanted`.

Vorstellung und Besprechung Zwischenstand und Release 0.9

Während der Vorlesung am **18.05.2020** zeigen Sie **uns allen** auf dem Beamer einen Zwischenstand und einen Plan bis zur Fertigstellung. Erzeugen Sie von Ihrem Zwischenstand dazu **vorher** bis zum **18.05.2020, 08:00** ein [Release](#) mit dem Tag 0.9.

Abgaben und Abschlußpräsentation

Im Rahmen des Vorlesungstermins (und Praktikumstermine) am **22.06.2020** muss Ihre Gruppe Ihr Ergebnis präsentieren. Für jede Gruppe steht ein 15-Minuten-Slot zur Verfügung, der, inkl. Anschluß Notebook an Beamer, eingehalten und vollständig genutzt werden muss.

Das gesamte Projekt (siehe [Umfang](#)) muss **bis zur Abschlußpräsentation** fertig bearbeitet und als [Release](#) mit dem Tag 1.0 bis spätestens **22.06.2020, 08:00** (mit entsprechender Beschreibung) veröffentlicht sein.

Beispiele für bereits bearbeitete Themen sind:

- [Graph-Tool](#)
- Implementierung und Visualisierung von Ameisen
- Labyrinthherstellung
- Suchen in statischen Texten
- Suchen in dynamischen Texten
- Rucksack Problem
- Traveling Salesman - Vergleich verschiedener Algorithmen
- Backtracking (Labyrinthsuche)
- Zuordnungsprobleme

Diese Themen sind in C++ bearbeitet worden und können mit einer eigenen Umsetzung in Go selbstverständlich noch einmal bearbeitet werden.

Umfang

- Go³-Code im Gruppen-GitHub-Repository
- README mit Installations- und Nutzungshinweisen
- Projekteintrag im [Wiki](#) mit Links zu Repo, Website, ...
- Unittests mit Code-Abdeckung $\geq 90\%$, wenn sinnvoll auch Benchmarks
- [Website für das Projekt](#) inkl. Sourcecode-Dokumentation
- Docker⁴-Image Ihres Programms
- Anbindung an meinen [Drone CI](#) und an [GitHub Actions](#)
 - Projekt übersetzen
 - Unittests ausführen
 - Docker-Image bauen und in die GitHub Packages-Registry pushen
 - wenn der Inhalt der GitHub-Page (Doku etc.) automatisiert gebaut werden soll, kann dies mit einer GitHub Action gemacht werden

³Für die Einarbeitung in Go können Sie z.B. das Aufgabenblatt 0 von [Verteilte Softwaresysteme](#) nutzen.

⁴Wenn Sie Docker auf Ihrem eigenen Rechner oder auf dem MWN-PC nutzen wollen, beachten Sie bitte, dass Docker nicht direkt auf dem MWN-PC bzw. nicht unter vielen Windows-Versionen läuft. Sie können aber sowohl auf dem MWN-PC als auch auf Ihrem eigenen Rechner meine [DevBox-VM](#) nutzen.