

## 1. Übungsblatt

Beachten Sie die organisatorischen Informationen auf der Kursseite.

**Bevor** Sie das Blatt bearbeiten, müssen Sie ein JDK (mind. Version 25) installiert haben (siehe Kursseite zu Woche 1). Falls Sie Probleme bei der Rechnereinrichtung oder der Bearbeitung von Blatt 1 haben, nutzen Sie die Hilfstermine für die Rechnereinrichtung (siehe organisatorischen Informationen).

### Aufgabe 1: Hello World nach VL 1

Schreiben Sie ein Programm `HelloWorld`, welches **ausschließlich** `Hello GitHubName!` (mit genau einem Leerzeichen nach „Hello“) auf der Standardausgabe ausgibt, wobei *GitHub-Name* Ihr eigener GitHub-Benutzername<sup>1</sup> ist.

*Für alle Übungsaufgaben gilt: Achten Sie auf Groß- und Kleinschreibung und geben Sie keine weiteren Zeichen als die geforderten aus.*

Wir empfehlen Ihnen, mithilfe unseres Abgabesystems zu prüfen, ob Sie Programmieraufgabe größtenteils korrekt gelöst haben, und Feedback von unserem Korrektur-Team einholen. Auf diesem ersten Blatt erklären wir Ihnen einmal, wie Sie eine Programmieraufgabe bearbeiten und über GitHub Classroom abgeben. Der Classroom führt sofort eine Reihe automatischer Tests durch, sodass Sie nach wenigen Minuten sehen, ob Sie die Aufgaben größtenteils richtig gelöst haben.

Eine Video-Anleitung dazu finden Sie im Kapitel zu Woche 1 auf der Kursseite im Abschnitt zum Abgabesystem.

Sollten Sie Vorgaben der Aufgabenstellung nicht verstehen oder andere Fragen haben, fragen Sie gerne im Ilias-Forum nach.

1. Öffnen Sie Ihren Texteditor und schreiben Sie den Quellcode für ein Programm, das den geforderten Text auf der Konsole ausgibt. Achten Sie dabei darauf, dass die Aufgabenstellung den Programmnamen (= das, was im Dateinamen vor `.java` steht) vorgibt; halten Sie immer die geforderte Groß-/Kleinschreibung ein, da diese beim Programmieren in Java relevant ist.
  - Wenn Sie nicht auswendig wissen, wie ein Java-Programm aufgebaut ist, schauen Sie in den Vorlesungsfolien, -aufzeichnungen oder der Literatur nach.
2. Wenn Sie zu einer Programmier-Aufgabe Feedback von unseren Hilfskräften bekommen wollen, z. B. zu Ihrer Lösungsstrategie, Ihrem Programmierstil oder warum die Abgabe nicht vollständig richtig ist, schreiben Sie `// Feedback gewünscht` ganz oben als erste Zeile in die java-Datei.

<sup>1</sup>oder irgendetwas anderes, falls Sie kein Konto für das Abgabesystem erstellen wollen

3. Speichern Sie die Datei auf Ihrem Computer unter dem Namen `Programmname.java`, also in diesem Fall `HelloWorld.java`. Merken Sie sich, in welchem Ordner Sie die Datei gespeichert haben.
4. Öffnen Sie ein Terminal und wechseln Sie mit dem Befehl `cd Ordner-Pfad` in den Ordner, wo Sie `HelloWorld.java` gespeichert haben.
  - Prüfen Sie mit dem Befehl `ls` (bzw. unter Windows (cmd): `dir`), ob Sie im richtigen Verzeichnis sind. Unter den aufgelisteten Dateien sollte sich Ihre `HelloWorld.java` befinden.
5. Testen Sie Ihr Programm mit `java HelloWorld.java`. Falls es nicht das Richtige tut, korrigieren Sie Ihren Quellcode (speichern nicht vergessen!) und testen Sie Ihr Programm noch einmal.
6. Sobald Ihr Programm funktioniert, geben Sie es über GitHub Classroom ab. Stellen Sie sicher, dass Sie unter <https://github.com/> mit Ihrem GitHub-Konto angemeldet sind.
  - Falls Sie kein GitHub-Konto haben, müssen Sie eines erstellen.
7. Klicken Sie auf das Symbol  oben neben dem Titel dieser Aufgabe. Eine Webseite sollte sich jetzt öffnen.
8. Klicken Sie auf *Accept this assignment*.
9. Warten Sie einen kurzen Augenblick und laden Sie die Seite neu. Klicken Sie dann auf den Link unter *Your assignment repository has been created*.
10. Klicken Sie neben dem grünen Button oben rechts auf *Add file* und dann auf *Upload files*.
11. Ziehen Sie alle für die Abgabe relevanten Dateien (in diesem Fall die `HelloWorld.java`) in das Upload-Feld.
12. Sobald die Datei hochgeladen ist, klicken Sie unten auf den grünen Button *Commit changes*, um die Datei zu speichern.
  - Die automatischen Tests prüfen nun Ihre Abgabe. Nach wenigen Minuten sollte das Ergebnis auf der Seite, wo Sie den Button *Add file* haben, sichtbar sein. (Die Seite wird nicht automatisch aktualisiert, sondern Sie müsste sie neu laden.)
  - Falls nicht alle Tests erfolgreich durchlaufen, sehen Sie nur eine Teilpunktzahl; Ihr Programm funktioniert dann nicht in allen Fällen so, wie es soll. Um zu sehen, welche Testfälle nicht funktionieren, klicken Sie unterhalb des Buttons *Add file* auf das rote X, dann auf *Details*, und schauen Sie sich die Ausgabe von *Prüfe Abgabe mit automatischen Tests* an. Das Lesen der Ausgabe der automatischen Tests erfordert etwas Übung; im Material zu Woche 1 sind die [häufigsten Meldungen erklärt](#). Fragen Sie im Zweifel im Forum, wenn Sie nicht verstehen, was die Testmelden Ihnen sagt; geben Sie dann am besten Ihren GitHub-Namen an (nur das Progra-Team kann Ihre Abgabe sehen, sonst niemand).
  - Sie können Ihren Code beliebig oft über *Add file* nachbessern. Die letzte Abgabe, die Sie vor der oben auf dem Übungsblatt angegebenen Deadline abgeben, wird sicher von unserem Feedback-Team angesehen. Verspätete Abgaben erhalten kein Feedback mehr.

- Wie in der Klausur gilt: Falls Sie verschiedene Lösungen zu einer Aufgabe abgeben (ohne genau anzugeben, welche bewertet werden soll), wird die schlechteste Lösung bewertet. Auf [der Kursseite ist erklärt](#), wie Sie eine falsch hochgeladene Datei löschen können. Wenn Sie eine Datei überschreiben wollen, muss die alte Datei nicht gelöscht werden; Sie können direkt die neue Datei hochladen.
13. Sofern Sie menschliches Feedback gewünscht haben, wird Ihre Abgabe in der Woche nach der Einreichfrist von einer Hilfskraft bewertet und kommentiert. Die Kommentare finden Sie dann spätestens eine Woche nach der Abgabefrist im Reiter „Issues“.

## Aufgabe 2: Textaufgabe: Ausführen [nach VL 1](#)

Angenommen, Sie wollen das java-Programm in der Datei `Hello.java` über die Konsole ausführen. Die Datei liegt im Ordner `/home/jinora/progra/Blatt01` und Ihr Terminal startet im Ordner `/home/jinora`.

Welche Befehle müssen Sie dafür auf der Konsole ausführen? Was ist der Zweck der einzelnen Befehle? (Ein Satz pro Befehl ist als Beschreibung ausreichend.)

Falls Sie die Antworten zur Korrektur abgeben: Schreiben Sie Ihre Antwort in eine Plaintext-Datei `antwort.txt` und fordern Sie Feedback mit `// Feedback gewünscht` an.<sup>2</sup>

Bei dieser Aufgabe handelt es sich um eine Textaufgabe, bei der Sie keinen Java-Code schreiben sollen. Wir empfehlen Ihnen, zu Freitextabgaben Korrektur-Feedback anzufordern; schreiben Sie dazu `// Feedback gewünscht` als erste Zeile in die Datei und geben Sie sie dann über das Abgabesystem ab.

Wir erklären Ihnen jetzt, wie das Abgeben von Textaufgaben erfolgt:

1. Formulieren Sie Ihre Antwort selbstständig, geben Sie sie in Ihrem Texteditor ein und speichern Sie sie als `antwort.txt`.
  - Achtung: Wenn Sie Text oder Gedanken aus anderen Quellen (dazu zählen nicht nur Buch- und Internetquellen, sondern auch die Vorlesungsfolien oder KI-Systeme) wörtlich oder sinngemäß übernehmen, *müssen* Sie gemäß der Regeln guter wissenschaftlicher Praxis die **Quelle(n) angeben** und wörtliche Übernahmen in Anführungszeichen setzen. Die Pflicht zur Quellenangabe gilt nicht nur für Freitext, sondern auch für Code.
  - Wenn Sie Korrektur-Feedback erhalten wollen, muss Ihre Abgabe zusätzlich den Text `// Feedback erwünscht` enthalten.
2. Klicken Sie oben neben dem Aufgabennamen von **Aufgabe 2** auf das Pfeil-Symbol, um zur Abgabeseite im Classroom zu gelangen.
  - Jede Aufgabe hat einen eigenen Abgabe-Link. Da das Verschieben falsch hochgeladener Abgaben ein hoher Aufwand für uns wäre, gibt es kein Feedback, wenn eine Aufgabe nicht an der richtigen Stelle hochgeladen worden ist. (Zumindest bei Programmieraufgaben merken Sie aber anhand der Tests, ob Sie an der richtigen Stelle hochgeladen haben.)

<sup>2</sup>Plaintext-Dateien können Sie direkt mit Ihrem Texteditor, den Sie auch zum Programmieren benutzen, erstellen. Schreiben Sie den Text in Ihrem Texteditor und speichern Sie ihn unter dem Namen `antwort.txt` ab.

3. Die weiteren Schritte sind jetzt wie bei der Abgabe der Programmieraufgabe. (Schritte [6–12](#))
4. Es wird automatisch geprüft, ob die korrekt benannte Datei vorhanden ist. Sie erhalten bei Textaufgaben aber keine automatische, inhaltliche Bewertung. Wenn Sie Feedback angefordert haben, werden innerhalb von 7 Tagen nach der oben auf dem Blatt angegebene Deadline unsere Korrektor:innen Ihnen eine Rückmeldung schreiben. Wie bei Code-Aufgaben können Sie Ihre Lösung jederzeit nachbessern.