Heinrich Heine Universität Düsseldorf

Programmierung – WS 2023/24
Dr. Markus Brenneis
Gesamtpunktzahl: 8 Punkte
Abgabefrist: 16.10.23, 10 Uhr

1. Übungsblatt

Beachten Sie die organisatorischen Informationen auf der Kursseite. Mit den ersten beiden Blättern können Sie noch **keine Punkte für die Klausurzulassung** sammeln, aber testen, wie das Abgabesystem funktioniert. Die Aufgaben werden also „ganz normal“ automatisch bewertet und Sie erhalten Feedback von unseren Korrektor:innen, die Punkte der Blätter 1 und 2 zählen aber nicht für die Bestimmung der Klausurzulassung.

Bevor Sie das Blatt bearbeiten, müssen Sie ein JDK 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

[4 Punkte]

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

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

Auf diesem ersten Blatt erklären wir Ihnen, wie Sie eine Programmier-Aufgabe bearbeiten und über GitHub Classroom abgeben. Eine Video-Anleitung dazu finden Sie im Kapitel *Organisatorisches* 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 (= Klassenname = das, was hinter `class` 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. 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.
3. 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.

4. Compilieren Sie das Programm mit `javac HelloWorld.java`. Wenn nichts ausgegeben wird, hat das Übersetzen richtig funktioniert.
   - Falls es Fehlermeldungen gibt, beheben Sie die Fehler von oben nach unten. Compilieren Sie nach jeder Fehlerbehebung neu.
5. Testen Sie Ihr Programm mit `java HelloWorld`. Falls es nicht das Richtige tut, korrigieren Sie Ihren Quellcode (speichern nicht vergessen!), compilieren Sie ihn erneut und testen Sie Ihr Programm noch einmal.
6. Sobald Ihr Programm funktioniert, geben Sie es über GitHub Classrom 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.
   - Denken Sie daran, uns im Ilias mitzuteilen, welches GitHub-Konto zu Ihnen gehört, damit wir die Punkte für die Klausurzulassung richtig zuordnen können. (vgl. Lernmodul *Organisatorisches*)
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.
    - Die vom Compiler erstellen class-Dateien müssen Sie nicht hochladen.
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 müsse sie neu laden.)
    - Bei Compilerfehlern erhalten Sie keine Punkte für Ihre Abgabe. Ihre voraussichtliche Punktzahl wird dann als 0 angezeigt. Da Sie aber Ihr Programm vorher selbst mit `javac` kompiliert haben, sollte es nur dann Compilerfehler geben, wenn Sie einen Fehler beim Hochladen gemacht haben.
    - Falls nicht alle Tests erfolgreich durchlaufen, erhalten Sie nur Teilpunkte. 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 Lernmodul zu Woche 1 sind die häufigsten Meldungen erklärt. Fragen Sie im Zweifel im Forum, wenn Sie nicht verstehen, was die Testmelden Ihnen sagt; wenn Sie

Ihren GitHub-Namen mit angeben, können auch unsere Tutor:innen (aber keine anderen Student:innen) direkt nachsehen, was das Problem bei der Abgabe ist.

- Sie dürfen Ihren Code bis zur Deadline (siehe oben auf dem Blatt) beliebig oft über *Add file* nachbessern; bewertet wird Ihre neueste Abgabe.

13. Ihre Abgaben werden in der Woche nach der Einreichfrist von unseren Korrektor:innen bewertet und kommentiert. Die Kommentare und Ihre finale Punktzahl finden Sie dann spätestens eine Woche nach der Abgabefrist im Reiter „Issues“.

Aufgabe 2: Textaufgabe: Compilieren & Ausführen nach VL 1 [4 Punkte]

Angenommen, Sie wollen die java-Quellcode-Datei `Hello.java` über die Konsole compilieren und ausführen. Die Datei liegt im Ordner `/home/jinora/progra/Blatt01`.

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

Schreiben Sie Ihre Antwort in eine Plaintext-Datei[1] *und geben Sie diese ab.*

Bei dieser Aufgabe handelt es sich um eine Textaufgabe, bei der Sie keinen Java-Code schreiben sollen.

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 z. B. als `loesung.txt`.
   - Achtung: Wenn Sie Text aus anderen Quellen (dazu zählen auch die Vorlesungsfolien) wörtlich oder sinngemäß übernehmen, *müssen* Sie die Quelle(n) angeben und wörtliche Übernahmen in Anführungszeichen setzen. (Ähnliches gilt auch für Code.)
2. Klicken Sie oben neben dem Aufgabennamen von **Aufgabe 2** auf das 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 keine Punkte, 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.)
3. Die weiteren Schritte sind jetzt wie bei der Abgabe der Programmieraufgabe. (Schritte 6–12)
4. Sie erhalten bei Textaufgaben *keine* automatische Bewertung, sondern unsere Korrektor:innen werden die Aufgabe gemäß der Bewertungsrichtlinien der Musterlösung bewerten. Stellen Sie sicher, dass sich nach *Commit changes* Ihre Textdatei in der online angezeigten Dateiliste befindet. Wie bei Code-Aufgaben dürfen Sie Ihre Lösung bis zur Deadline nachbessern.
   - 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.

[1] z. B. txt, adoc (AsciiDoc), md (Markdown), . . . ; insbesondere folgende Formate akzeptieren wir **nicht**: docx (Office Open XML), odt (OpenDocument-Text)