

Programmieren I

01. Übungsblatt

*Melden Sie sich bitte für die Veranstaltung bei Stud.IP (<http://studip.tu-bs.de>) an.
Wir bitten Sie **dringend**, sich in der Rechnerübung mit ihrem Übungspartner Ihrem
Tutor vorzustellen.*

*Sie können sich einen Partner/eine Partnerin suchen und sich gemeinsam vorstellen.
Verspätete oder nicht hochgeladene Abgaben, sowie Plagiate führen zur Bewertung der
Aufgabe mit 0 Punkten.*

Empfehlung: Machen Sie sich mit einer Shell vertraut. Wir empfehlen die Verwendung
der *bash*-Shell.

Sollten Sie auf einem eigenen Rechner entwickeln wollen, so installieren Sie das für ihr
Betriebssystem passende **Java SE Development Kit**¹.
Zum Versionieren Ihrer Quelltexte setzen wir **git** (<https://git-scm.com/>) voraus.
Wir raten Ihnen ausschließlich einen normalen Texteditor (e.g. gEdit, Kate, nano, Note-
pad, emacs, vim) zum Programmieren zu verwenden.

Aufgabe 1: Melden Sie sich an einem der Rechner an. Sollten Sie es noch nicht getan
haben, ändern Sie zuerst Ihr Passwort. Gehen Sie dabei z. B. wie folgt vor:

Nehmen Sie von einem Satz mit mindestens sechs Wörtern die Anfangsbuch-
staben jedes Worts. Fügen Sie die Anzahl der Wörter des Satzes sowie zwei
feste Sonderzeichen hinzu.

Melden Sie sich anschließend ab und wieder an. Zur Änderung Ihres Passworts verwenden
Sie bitte das Web-Interface:

<https://www.tu-braunschweig.de/it/service-interaktiv/passwortaendern>

¹OpenJDK oder das Oracle JDK, siehe Slides der ersten Großen Übung.

Aufgabe 2: Bearbeiten Sie diese Aufgabe, falls Sie im Umgang mit Linux noch nicht sattelfest sind.

- a) Am Ende dieses Aufgabenblatts finden Sie einige Unix-Kommandos. Lesen Sie die Manual-Seiten zu den Kommandos (e.g. `man ls`) und informieren Sie sich über ihre Optionen und Parameter.
- b) Wechseln Sie in Ihr Heimatverzeichnis (Befehl `cd`). Wie lautet der Pfad Ihres Heimatverzeichnisses (Befehl `pwd`)? Geben Sie zuerst mittels `ls -l` den Inhalt Ihres Heimatverzeichnisses aus und danach mittels `ls -al`. Was sind die Unterschiede? Lesen Sie die Manual-Seite (Befehl `man`) des Befehls `ls` und versuchen Sie diese zu verstehen. Testen Sie einige der angegebenen Optionen.
- c) Bewegen Sie sich mit dem Befehl `cd` durch den Dateibaum. Verwenden Sie dabei absolute und relative Pfadnamen. Benutzen Sie zur Navigation auch die speziellen Namen `.`, `..` und `/`.
- d) Wiederholen Sie den Umgang mit den Befehlen `cd`, `ls`, `cat`, `less`, `touch`, `mkdir`, `rmdir`, `cp`, `rm`, `mv` und `ln`, indem Sie sich selbstständig weitere Beispiele ausdenken und ggf. die Manual-Seiten konsultieren.

Aufgabe 3: Geben Sie mit einem Texteditor Ihrer Wahl (z. B. gEdit, Kate) die Datei `HelloWorld.java` mit folgendem Inhalt ein.

```
1 class HelloWorld {  
2     public static void main(String[] args) {  
3         System.out.println("Hello World");  
4     }  
5 }
```

listings/HelloWorld.java

Übersetzen Sie das Programm mit dem Java-Compiler (`javac`) und führen Sie es anschließend mit dem Java-Interpreter (`java`) in der Shell aus:

```
javac HelloWorld.java  
java HelloWorld
```

Beobachten Sie welche Dateien während des Übersetzungsvorgangs angelegt werden. Überlegen Sie welche Dateien für ein anderes Gruppenmitglied wichtig sind um das Programm selbstständig zu übersetzen und auszuführen.

Aufgabe 4: Machen sie sich mit Git vertraut.

Eine grundlegende Einführung in den Umgang mit Git gibt es in der großen Übung am 8. Mai. Zahlreiche Git Tutorials sind auch online verfügbar²³⁴.

Ähnlich wie beim Programmieren lernen sie Git allerdings nur durch aktive Nutzung! Wir stellen für die Abgabe der Pflichtaufgaben zwar einen Git Server bereit, dieser sollte allerdings nicht für Experimente verwendet werden⁵. Um gefahrlos testen zu können, empfehlen wir daher ein gemeinsames Repository bei einem der Anbieter aus dem Hinweis zu verwenden.

Hinweis: Mögliche Anbieter für Git Repositories sind beispielsweise:

- *Github*^a (Achtung: Github erlaubt das Anlegen von privaten Repositories nur für zahlende Nutzer oder Mitglieder des Github Education Programs^b)
- *Gitlab*^c Bietet unlimitierte private repositories.
- *Bitbucket*^d Bietet unlimitierte private repositories.

^a<https://github.com/>

^b<https://education.github.com/>

^c<https://about.gitlab.com/>

^d<https://bitbucket.org/>

²<https://git-scm.com/book/en/v2>

³<http://rogerdudler.github.io/git-guide/>

⁴<https://guides.github.com/introduction/git-handbook/>

⁵Das Löschen und neu anlegen von Repositories ist von unserer Seite aus beispielsweise nicht vorgesehen.

Hinweis: Sie sollten inzwischen eine Gruppe zum lösen der Aufgaben gefunden haben.

Hinweis: Die Pflichtaufgabe muss auf unserem Git Server bearbeitet werden. Dieser ist unter <https://dev0.ias.cs.tu-bs.de/> erreichbar. Dort können Sie sich mit Ihrer y-Nummer und dem zugehörigen Passwort anmelden.

Zum gemeinsamen Arbeiten an einem Repository melden Sie sich bitte bei dem Tutor/der Tutorin Ihrer Übungsgruppe. Dort werden die entsprechenden Zugangsberechtigungen gesetzt.

Pflichtaufgabe 5: Diese Aufgabe dient dazu sich mit der Nutzung von Git im Kontext eines Gruppen Projekts vertraut zu machen.

- Erstellen Sie eine lokale Kopie des gemeinsamen Projektes via *git clone*.
- Machen Sie sich mit der bereits vorhandenen Ordnerstruktur vertraut. Diese muss für die Abgaben so weiter verwendet werden (e.g. Pflichtaufgabe 1 in Order Blatt1)
- Sie sollen immer nur die Quelldateien in Git *committen*. Erstellen sie daher eine passende *.gitignore*⁶ Datei. Diese soll z.B. die vom Java Compiler erzeugten Bytecode Dateien ausschließen.
- Legen sie in dem für diese Aufgabe vorgesehenen Ordner eine oder mehrere Textdateien mit beliebigem Inhalt an.
- Bearbeiten Sie diese parallel so das beim *pushen* und *pullen* Merge Konflikte auftreten. Die Aufgabe ist es, dass jedes Gruppenmitglied mindestens einen Merge Konflikt aufgelöst hat. Das Resultat könnte beispielsweise aussehen wie in Abbildung 1!

Diese Aufgabe gibt 5 Punkte, für jeden der Zwischenschritte einen Teilpunkt.

Berücksichtigt werden ausschließlich Ergebnisse, die bis zum DO 21.11.2019 um 23:59 in das Git Repository Ihrer Gruppe gepusht wurden.

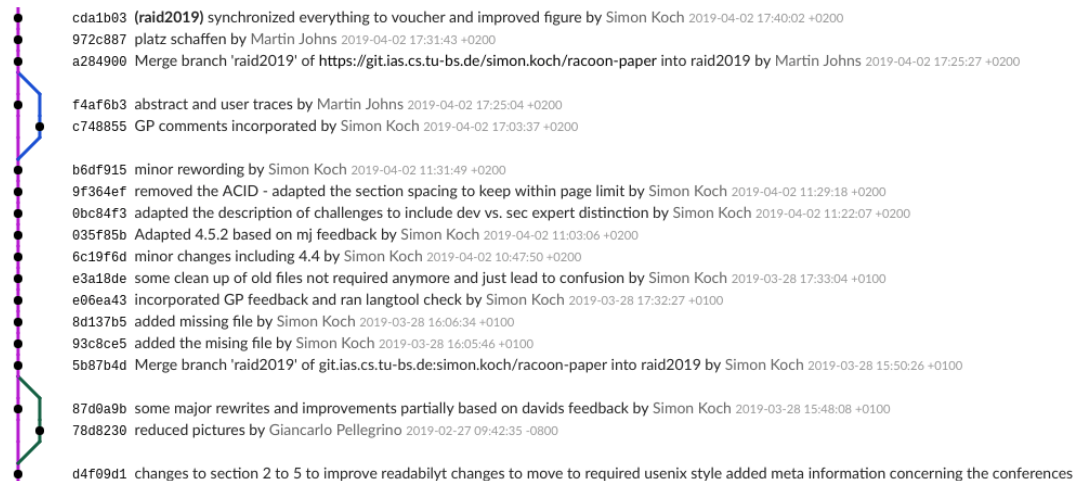
Um Ihre Lösung bewertet zu bekommen, müssen Sie diese Aufgabe in der Woche vom 25.11. - 29.11. dem Tutor/der Tutorin in Ihrer Übungsgruppe vorstellen. Dies ist nicht optional! Nicht oder mangelhaftes Vorstellen führt zu 0 Punkten

Hinweis: Die Bearbeitungszeit der Pflichtaufgabe dieses Aufgabenblatts ist gering. Nutzen Sie die Ihnen zur Verfügung stehende Zeit daher unbedingt um sich weiterhin mit der Nutzung von Linux/Git vertraut zu machen. Probleme mit Git sind generell **keine Entschuldigung für Probleme mit der Bearbeitung von Aufgaben.**

⁶Dokumentation zu *.gitignore*: <https://git-scm.com/docs/gitignore>

Abbildung 1: Beispielhaftes Ergebnis für die Pflichtaufgabe.

Commit Graph



Einige Unix-Kommandos

Informationen / Hilfe

`man` Manual-Seiten
`info` Info-Seiten

Shells

`sh`, `ksh`, `bash`, `zsh`, `csh`, `tcsh`, ...

bash (als Beispiel)

`bash` Aufruf der Shell
`exit` Beenden der Shell
`set` Variable
`alias` Namen für Kommandos
`history` Befehlsliste

Verzeichnisse

`ls` Anzeigen des Inhalts
`cd` Verzeichniswechsel
`mkdir` Anlegen eines Verzeichnisses
`rmdir` Löschen eines Verzeichnisses

Pfade

`pwd` Anzeige des aktuellen Pfads
`.` aktuelles Verzeichnis
`..` übergeordnetes Verzeichnis
`/` Wurzelverzeichnis

Dateiverarbeitung

Editor Dateibearbeitung
`less` Ansehen einer Datei
`cp` Kopieren einer Datei
`mv` Verschieben einer Datei
`rm` Löschen einer Datei
`diff` Vergleich von Dateien
`cat` Konkatenieren von Dateien

Editoren

`gedit`, `kate`, `vim`, ...

Java

`javac` Compiler
`java` Interpreter