

## Vorkurs

### 04. Übungsblatt

**Hinweis:** Aufgabe 1, 2 und 3 sind aus dem Themenbereich Bedingungen, Aufgabe 4 und 5 Git. Ich empfehle Dir auf jeden Fall, Dich an den Git-Aufgaben zu versuchen, da Git auch später zur Abgabe der Hausaufgaben verwendet wird.

**Aufgabe 1:** Gebe alle geraden Zahlen, sowie die, die durch 7 teilbar sind, von 1 bis 1000 in der Konsole aus. Beispiel:

```
2 4 6 7 8 10 12 14 ...
```

**Aufgabe 2:** Erstelle ein Programm in der jshell, das den Stadtnamen und ihre Einwohnerzahl als von Dir bestimmte Variablen besitzt, und anhand der Einwohnerzahl die Stadtart nach deutschem Stadtgesetz zurückgibt. Beispiel mit Coburg:

```
jshell> String name = "Coburg";  
jshell> int einwohnerZahl = 41316;
```

```
...
```

```
Coburg ist eine Mittelstadt!
```

**Aufgabe 3:** Schreibe einen Taschenrechner in der jshell, welcher zwei ganze Zahlen und als drittes einen Operator als von Dir bestimmte Variablen entgegennimmt, und dann das Ergebnis der Rechnung ausgibt. Er soll Plus, Minus, Mal und Geteilt beherrschen. Beispiel:

```
jshell> int linkeZahl = 1;  
jshell> char operator = "+";  
jshell> int rechteZahl = 5;
```

```
...
```

```
1 + 5 = 6
```

**Aufgabe 4:** Viele Programme werden kontinuierlich in einem Team weiterentwickelt. Um die gemeinsame Arbeit zu erleichtern, gibt es Versioning Tools wie Git. In dieser Aufgabe wird ein erstes Repository erstellt und eine Datei committed.

a) Erstelle einen neuen Ordner **MeinRepository**, navigiere in ihn hinein und erzeuge in ihm ein neues **Repository** mit `git init`.

b) Der neue Ordner beinhaltet noch keine Dateien außer den `.git`-Ordner, der die Repositorydaten enthält. Erstelle die Textdatei `readme.txt` und schreibe einen beliebigen Begrüßungstext in sie. Sie ist nun Teil der **Working Copy**.

c) Füge jetzt die Textdatei zum **Index** mit `git add readme.txt` hinzu.

d) Wenn nun eine andere Person dein Repository klonen würde, hätte er noch keinen Inhalt. Deine Datei befindet sich nämlich noch im Index. Du musst die Datei mit einem **Commit** zum Teil des Repositories machen. Führe nun zuletzt ein `git commit -m "Mein erster Commit!"` aus.

**Aufgabe 5:** Meist liegt das Repository für ein Projekt an einem entfernten Ort, zum Beispiel Github. In dieser Aufgabe geht es darum, sich eine lokale Kopie eines solchen Repositories zu erzeugen, auf diesem zu arbeiten und die Änderungen zum entfernten Repository zu übertragen.

a) Wenn du dich in dem Überordner von **MeinRepository** aus Aufgabe 3 befindest, kclone es in **MeinLokalesRepository** mittels

`git clone MeinRepository MeinLokalesRepository`. **MeinLokalesRepository** ist allerdings ein Repository mit Working Copy. Damit du problemlos deine Commits auf dieses Repository pushen kannst musst du hineinnavigieren und es mittels `git config core.bare true` zu einem *Bare-Repository* machen. Dadurch kann auf **MeinRepository** nicht mehr lokal committed werden, es verliert also seine Fähigkeit seine eigene Working Copy zu besitzen.

b) Erstelle eine Textdatei `HalloWelt.txt` in **MeinLokalesRepository**, in der *Hallo, Welt!* steht. Füge nun `HalloWelt.txt` zum Index hinzu und committe es.

c) Übertrage deine Änderungen zu **MeinRepository** mit `git push`.

d) Navigiere in den Überordner der Repositories und kclone **MeinRepository** in **MeinZweitesLokalesRepository** mittels

`git clone MeinRepository MeinLokalesRepository`.

Deine `readme.txt` und das Java-Programm sollten nun auch in dem neuen Ordner verfügbar sein.