

# Assignment 1

## GRUNDLAGEN

Abgabe: 17.04.2018, 23:59 Uhr

*So viel haben sie schon gelernt, daß sie nur den Statistiken glauben, die sie selbst gefälscht haben.*

— Hanns-Erich Haack, *Ameisenstaat oder Sintflut*. 1946, S. 137–147

### I. ORGANISATORISCHES

- Machen Sie einen Commit mit einem *Aussagekräftigen Text* und pushen Sie all ihre lokalen Commits zum Server. Es können nur Abgaben gewertet werden, die auch zu unserem Server gepusht wurden.
- Es zählt der letzte Commit vor der Abgabezeit. In unserem Projektmanagementsystem (<https://pm.stud.uni-hannover.de>) können Sie den aktuellen Stand ihres Repositories kontrollieren.
- Beachten Sie die Hinweise im StudIP.
- Gruppenabgaben sind nicht zulässig. Plagiate gelten als Täuschungsversuch.
- Es wird voraussichtlich 5 Übungen geben.
- Für die Studienleistung müssen alle Assignments und die Gruppenaufgabe bestanden werden.
- Alle Abgaben müssen einem Tutor erklärt werden können. Ist dies nicht der Fall, wird von einem Plagiat ausgegangen.

## II. EINRICHTUNG

### 1. JDK

1. Besuchen Sie <http://www.oracle.com/>
2. Installieren Sie das JDK entsprechend Ihres Betriebssystems.
3. Erstellen Sie eine neue Datei HelloWorld.java
4. Füllen Sie die Datei mit folgendem Inhalt:

```
public class HelloWorld
{
    //prints hello world
    public static void main(String[] args)
    {
        System.out.println("Hello , World!");
    }
}
```

5. Öffnen Sie das Terminal (Linux, OS X) bzw. die Eingabeaufforderung (Windows) und navigieren Sie in den Ordner, in dem sich die Datei befindet.
6. Kompilieren Sie die Datei mit:

```
$ javac HelloWorld.java
```

7. Führen Sie die Datei aus:

```
$ java HelloWorld
Hello , World!
$
```

### 2. Git

1. Git installieren:

- Windows: git for windows (<http://msysgit.github.io>)
- Mac OS: git-osx-installer (<http://sourceforge.net/projects/git-osx-installer/>)
- Linux: Installation über den Paketmanager. Z.B. für Ubuntu: `sudo apt install git`

2. Git konfigurieren:

```
$ git config --global user.name "Max Musteruser"
$ git config --global user.email user@domain.de
```

3. Im Projektmanagementsystem (PMS) anmelden:

- (a) Adresse: <https://pm.stud.sim.uni-hannover.de>
- (b) Nachdem Sie sich in im StudIP in eine Gruppe eingetragen haben, erhalten Sie am nächsten Tag (ca. 22:00 Uhr) Ihre Zugangsdaten an die im StudIP hinterlegte E-Mail-Adresse
- (c) Ändern Sie das Standardpasswort nach dem ersten Login
- (d) Unter *Projekte* finden Sie ihr persönliches Projekt

(e) Unter *Projektarchiv* ist ihre Kennung für das Git-Repository zu finden.



4. Persönliches Repository klonen:

```
$ cd /path/to/some/folder
$ git clone https://git.pm.stud.sim.uni-hannover.de/<Kennung>
```

- Anschließend muss der Benutzername und das Passwort des PMS eingegeben werden

5. Grundgerüst aus dem StudIP herunterladen und in das Git-Repository kopieren.

6. Übungsblatt bearbeiten

7. Dateien tracken und stagen (aus dem Hauptverzeichnis):

```
$ git add .
```

8. Den Status des Git anzeigen, um zu kontrollieren, ob die gewünschten Dateien gestaged sind.

```
$ git status
```

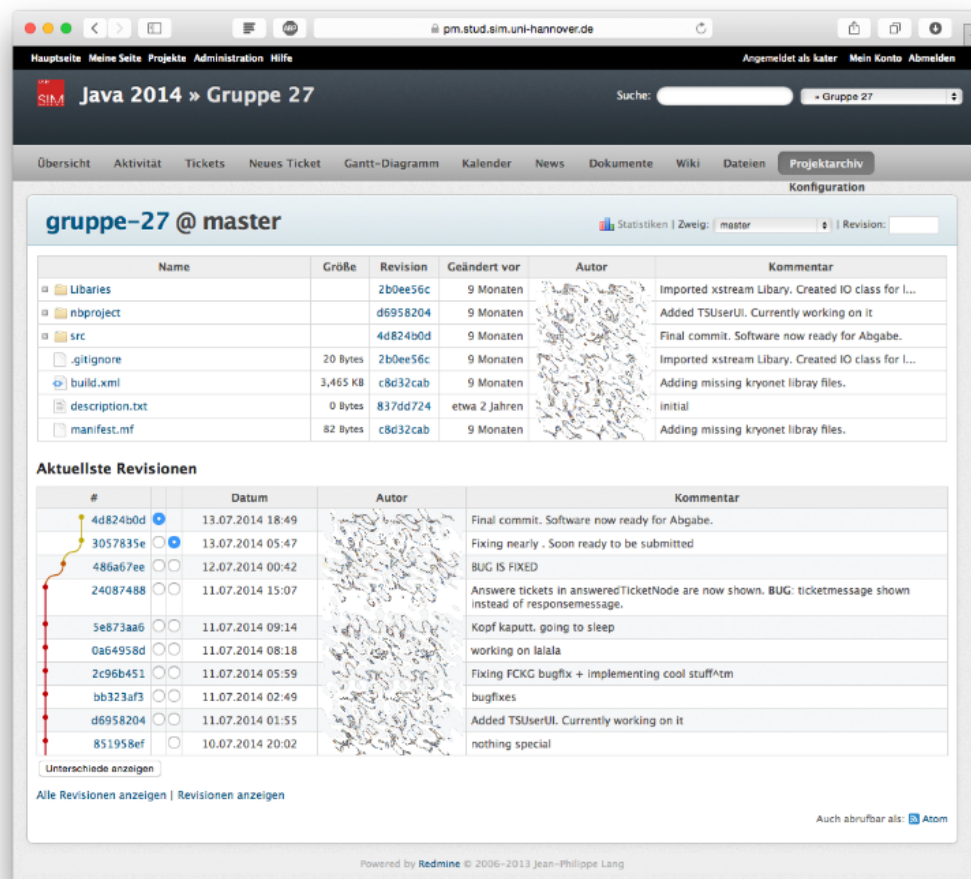
9. Einen Commit anlegen.

```
$ git commit -m "Abgabe Uebungsblatt 1"
```

10. Den Commit an den Server pushen

```
$ git push
```

11. Im PMS kontrollieren, ob der Commit an den Server gepusht wurde.



**gruppe-27 @ master**

Name	Größe	Revision	Geändert vor	Autor	Kommentar
Libraries		2b0ee56c	9 Monaten		Imported xstream Library. Created IO class for I...
nbproject		d6958204	9 Monaten		Added TSUserUI. Currently working on it
src		4d824b0d	9 Monaten		Final commit. Software now ready for Abgabe.
.gitignore	20 Bytes	2b0ee56c	9 Monaten		Imported xstream Library. Created IO class for I...
build.xml	3,465 KB	c8d32cab	9 Monaten		Adding missing kryonet library files.
description.txt	0 Bytes	837dd724	etwa 2 Jahren		Initial
manifest.mf	82 Bytes	c8d32cab	9 Monaten		Adding missing kryonet library files.

**Aktuellste Revisionen**

#	Datum	Autor	Kommentar
4d824b0d	13.07.2014 18:49		Final commit. Software now ready for Abgabe.
3057835e	13.07.2014 05:47		Fixing nearly . Soon ready to be submitted
486a67ee	12.07.2014 00:42		BUG IS FIXED
24087488	11.07.2014 15:07		Answers tickets in answeredTicketNode are now shown. BUG: ticketmessage shown instead of responsemessage.
5e873aa6	11.07.2014 09:14		Kopf kaputt. going to sleep
0a64958d	11.07.2014 08:18		working on lalala
2c96b451	11.07.2014 05:59		Fixing FCKG bugfix + implementing cool stuff!tm
bb323af3	11.07.2014 02:49		bugfixes
d6958204	11.07.2014 01:55		Added TSUserUI. Currently working on it
851958ef	10.07.2014 20:02		nothing special

**Hinweis:** Zur Übersicht wurde in dieser Anleitung nur ein Commit erstellt. Sinnvoller ist es mehrere kleine Commits zu machen. Pro Commit sollte ein Feature implementiert sein.

```
$ git add Matrix.java
$ git commit -m "Modelklasse Matrix implementiert"
$ git add MatrixOperations.java
$ git commit -m "Methode matMult implementiert"
$ git push
```

### III. STATISTIK

Um einen Datensatz statistisch zu analysieren und zu interpretieren, kann es hilfreich sein, einige Kennwerte auszurechnen:

- Gegeben: ein Array mit Messdaten *data*
- Anzahl der Daten:  $n := |data|$
- Maximum, das größte Element des Arrays:

$$\max(data) := x \in data : \forall y \in data : y \leq x$$

- Minimum, das kleinste Element des Arrays:

$$\min(data) := x \in data : \forall y \in data : x \leq y$$

- Arithmetisches Mittel:

$$\mu_{data} := \frac{\sum_{i=0}^{n-1} data[i]}{n}$$

Das arithmetische Mittel ist die gebräuchlichste Methode, kann aber stark durch „Ausreißer“ (extremes Maximum oder Minimum) beeinflusst werden.

- Untermedian: Mittleres Element des *sortierten* Arrays:

$$\text{median}(data) := data[\lfloor \frac{n-1}{2} \rfloor]$$

Der Median wird im Gegensatz zum arithmetischen Mittel weniger stark von Ausreißern beeinflusst.

- Standardabweichung:

$$\sigma := \sqrt{\frac{1}{n} \sum_{i=0}^{n-1} (data[i] - \mu_{data})^2}$$

Die Standardabweichung ist ein Maß dafür, wie weit die Stichprobe im Schnitt um das arithmetische Mittel streut.

- Spannweite

$$\text{range}(data) := \max(data) - \min(data)$$

- Quartile: Wie *median*, nur das *data* in vier Teile geteilt wird.

#### IV. AUFGABEN

1. Führen Sie die Einrichtung wie in II angegeben aus.
2. In der Datei `Statistics.java` finden Sie ein Array mit Messdaten und die Funktion `calculateMax()`, welche das Maximum der Daten im Datensatz ausgibt.  
Implementieren Sie folgende Funktionen nach dem Prinzip von `calculateMax()`:  
`calculateMin()`, `calculateMean()`, `calculateMedian()`,  
`calculateStandardDeviation()`, `calculateRange()`, `calculateLowerQuartile()`,  
`calculateUpperQuartile()`
3. Die Ausgabe sollte wie folgt aussehen:

```
$ javac Statistics.java
$ java Statistics
maximum: 1.088398684
minimum: -4.486836197
mean: -2.0369579577624
median: -2.246221753
standard deviation: 1.0515574835721706
range: 5.575234881
lower quartile: -2.760476106
upper quartile: -1.515287465
```

#### V. BEWERTUNGSKRITERIEN

1. Die Einrichtung wird nicht bewertet, ist aber **essentiell** für die Bearbeitung der weiteren Aufgaben. In folgenden Übungen wird davon ausgegangen, dass Ihr System korrekt eingerichtet ist. Bei Problemen sprechen Sie uns bitte **diese** Woche an.
2. Zum Bestehen der Aufgabe III...
  - (a) ...muss das Programm fehlerfrei kompilierbar und ausführbar sein.
  - (b) ...müssen alle Werte richtig berechnet werden.
  - (c) ...muss der Code einem Tutor erklärt werden können.