



## Aufgabe A0 Email

Jedes Team sendet eine Email an

[WP.S16S@Hamburg-UAS.eu](mailto:WP.S16S@Hamburg-UAS.eu)

und als CC an jedes einzelne Teammitglied.<sup>1</sup>

Dies ist gemäß den in der Vorlesung durchgesprochenen Richtlinien zu tun.

Also mit einem Subject, das Ihren Teamnamen (der Ihnen während des ersten Laborterminals mitgeteilt wird) gefolgt von " WPP\_CPJ\_A0\_V#" (nur Großbuchstaben) enthält.

"#" ist durch eine Dezimalziffer(>0) zu ersetzen, die die Versionsnummer der Abgabe identifiziert.

← Eigentlich hätte ich gehofft, dass dieser Satz nicht nötig wäre ;)

Die Versionsnummer entspricht der Anzahl der Versionen einer Abgabe – die vierte Email zur Abgabe einer spezifischen Aufgabe hat also die Versionsnummer 4.

In der Email schreiben Sie die vollständigen Namen aller Teammitglieder, deren Matrikelnummer und deren (Haupt-)Email-Adresse.

Bsp:

Ihr Team sei

S2T3

Teammitglieder seien

Beate Beispielfrau, Matr.Nr.:1234567, [Beate.Beispielfrau@haw-hamburg.de](mailto:Beate.Beispielfrau@haw-hamburg.de)

Max Mustermann, Matr.Nr.:1234568, [Max.Mustermann@haw-hamburg.de](mailto:Max.Mustermann@haw-hamburg.de)

Die Email wird das erste Mal geschickt.

Die Email hat also folgenden Aufbau

Subject: S2T3\_WPP\_CPJ\_A0\_V1

Mail: Name: Beispielfrau, Beate,  
Matr.Nr.: 1234567,  
email: [Beate.Beispielfrau@haw-hamburg.de](mailto:Beate.Beispielfrau@haw-hamburg.de)

Name: Mustermann, Max,  
Matr.Nr.: 1234568,  
email: [Max.Mustermann@haw-hamburg.de](mailto:Max.Mustermann@haw-hamburg.de)

Bemerkungen:

- Subject ist das englische Wort für Betreff. Es ist also die Betreffzeile gemeint!
- Ihr Subject muss nicht farbig sein.
- Ein Team besteht aus 2 Mitgliedern.

<sup>1</sup> Sie können im CC-Feld auch mehr als eine Email-Adresse für jedes einzelne Teammitglied angeben.

Beispielsweise: [beispl\\_b@informatik.haw-hamburg.de](mailto:beispl_b@informatik.haw-hamburg.de), [beate.beispielfrau@yahoo.de](mailto:beate.beispielfrau@yahoo.de), [muster\\_m@informatik.haw-hamburg.de](mailto:muster_m@informatik.haw-hamburg.de), [maexchen@maxi.org](mailto:maexchen@maxi.org)  
Wenn ich Sie erreichen will, mache ich ein Reply-To-All auf die A0-Email. Sie haben es in der Hand, wo diese Email dann hinget.

Sofern Sie noch nicht mit dem Version Control System "GIT" vertraut sind, machen Sie sich damit vertraut. Im Netz finden Sie das freie Buch "Pro Git" von Scot Chacon - lesen die ersten Kapitel.

Für GIT (auf der Client-Seite - also auf "Ihrer Seite") empfehle ich Portable GIT - dies ist kein Zwang - nur eine Empfehlung. Es folgen ein paar Bemerkungen zu Portable GIT auf dem USB Stick für DOSen.

Unter:

[https://users.informatik.haw-hamburg.de/~schafers/PUB/GITonUSB\\_v\\*/](https://users.informatik.haw-hamburg.de/~schafers/PUB/GITonUSB_v*/)

finden Sie eine Zusammenstellung, die den Umgang mit Portable GIT auf einem USB-Stick erleichtert. Nach dem Entpacken von GITonUSB\_v\*.zip finden Sie eine Datei \_GITonUSB\_HowTo.txt. Folgen Sie den darin enthaltenen Anweisungen. Lassen Sie Ihr Home-Directory auf dem USB-Stick - dies ist so voreingestellt. Sofern Sie es nicht schon getan haben, erzeugen Sie sich einen Identifizierungs-SSH-Key (.ssh/id\*) und legen diesen Key in dem Home-Directory auf dem USB-Stick ab und **nicht!** auf dem Laborrechner. Andernfalls könnten sich Fremde als Sie identifizieren und das wäre fatal.

## Für alle (weiteren) Aufgaben gilt:

- Alle Source-Code-Dateien weisen einen Header auf in dem min. alle beteiligten Code-Autoren mit Ihrer Email-Adresse stehen.
- Sofern es nicht ausdrücklich in einer Aufgabenstellung anders eingefordert wird gilt:  
**Ein Team hat pro Aufgabe immer genau ein Repository auf BitBucket.org.** Dieses Repository dient als zentrales Repository und hat **immer** den aktuell gültigen Code zu enthalten. D.h. alle Source-Code-Dateien, die Sie selber erzeugt haben oder die zum Projekt gehören sind in dem zentralen Repository abzulegen.
- Konsequenz (vom Punkt zuvor): Das Repository wird genutzt! In sinnvollen Abständen wird Code eingchecked. Alle Teammitglieder nutzen GIT. D.h. es sind Check-Ins von allen Teammitgliedern zu sehen.
- **Als 1. Aktion bei der Bearbeitung einer Aufgabe ist immer jeweils dieses eine Repository auf BitBucket anzulegen.** Hiermit belegen Sie den Start Ihrer Vorbereitungen.
- Das Repository auf BitBucket ist **min. 1 Woche vor** dem zugehörigen Laborterminal anzulegen. Unmittelbar danach ist "die Email" (s.u.) zu verschicken. "Die Email" wird also min. 1 Woche vor dem zugehörigen Laborterminal abgesendet.
- Das Repository auf BitBucket bleibt min. bis zum Beginn des nächsten Semesters bestehen.
- **Das Repository auf BitBucket ist private.**
- **Jedes Teammitglied hat mindestens Schreibrechte auf das Repository auf BitBucket.**
  - **ElCritico (Prof) sind Leserechte auf das Repository auf BitBucket zu geben.** (Die Leserechte erhalten sie ebenfalls min. 1 Woche vor dem jeweiligen Laborterminal).
- Das Repository (auf BitBucket) enthält u.a.
  - eine Datei readme.md, die in einer Zeile den Sinn des Repository bzw. Projekts erklärt.
  - eine Datei ReadMe.txt mit ausführlicheren Informationen u.a. beteiligte Autoren&Quellen, Kurz-Beschreibung - was macht das Projekt (AufgabenNr reicht), TeamNr, Semester, History der Code-Reviews/Abnahmen/Abnahmeversuche
  - Ihre Lösung (nur Source-Code z.B.: \*.java, \*.vhd, \*.do, \*.ucf, \*.h, \*.c)
- Nach dem Anlegen des Repository ist eine **Email an WP.S16S@Hamburg-UAS.eu** (und alle Teammitglieder als CC) mit folgendem Aufbau zu schicken:

Subject:    **S#T#\_WPP\_CPJ\_A#\_V#**

Mail:        **URL des GitBucket-Repository**

"#" ist durch eine Dezimalziffer(>0) zu ersetzen, die die Versionsnummer der Abgabe identifiziert. Wie zuvor entspricht die Versionsnummer der Anzahl der Versionen einer Abgabe zu der jeweiligen Aufgabe.

### Beispiel:

Das Beispiel Team **S9T8** schickt im Falle der Aufgabe **A1** als **1.** Email die folgende Email:

Subject:    **S9T8\_WPP\_CPJ\_A1\_V1**

Mail:        **[git@bitbucket.org/BE/A1.git](mailto:git@bitbucket.org/BE/A1.git)**

BE sei der Account-Name von Elke Beispielfrau bei BitBucket.org. In diesem Beispiel habe Elke Beispielfrau das Repository für A1 auf BitBucket.org angelegt. Durch das Senden der Email an alle Teammitglieder (als CC) können Sie die erfolgreiche Versendung überprüfen und haben einen Beleg, dass Sie die Teamkollegen und den Prof. über das Repository informiert haben. Die Teammitglieder können durch clonen des Repository viele grundsätzliche Dinge testen.

## Aufgabe A1 Verschiedene Teilaufgaben

Vorweg:

Bzgl. Zeitlicher Auflagen zu GIT – das ist entspannt – ich bin ja selbst zu spät.  
Aber trotzdem bitte als Probelauf machen.

Teilaufgabe 2 ist die schwierigste – falls keine Idee, mich Fr. nach CE abpassen und nach Tipp fragen

Teilaufgabe 3 bitte versuchen – falls zu schwer, dann zum nächsten Termin abgeben – sollte aber nicht zu schwer sein.

Die Aufgabe A1 ist im Package "a1" zu lösen.

### Teilaufgabe1

Die Teilaufgabe1 ist im Subpackage "p1" bzw. "a1.p1" zu lösen.

Ein Thread (der "main()-Thread" ist ok) soll jeweils nach Eigenschaften für seinen Kind- und Enkel-Thread fragen. Konkret jeweils nach dem Thread-Namen und der Thread-Priorität und diese sollen dann entsprechend für Kind und Enkel gesetzt werden. Jeder Thread - also "Start-Thread" selbst (vermutlich der main()-Thread) sowie Kind-Thread und Enkel-Thread sollen jeweils für sich Thread-ID, Thread-Name und Thread-Priorität ausgeben.

Ferner sollen die Eltern jeweils auf den Tod ihrer Kinder warten bevor sie selbst sterben.

Für mögliche Ausgaben sollten Sie folgendes Schema einsetzen.

```
StringBuffer sb = new StringBuffer();
...
// Ausgabe in sb berechnen
...
// "Kurz" ausgeben
System.out.flush();
System.out.print( sb.toString() );
System.out.flush();
```

### Teilaufgabe2

Die Teilaufgabe2 ist im Subpackage "p2" bzw. "a1.p2" zu lösen.

Bestimmen Sie mit einem Java-Programm die Länge einer (JVM-)Zeitscheibe.

Achten Sie dabei darauf, dass im "Hintergrund" keine Programme laufen, die die "Messung" belasten.

### Teilaufgabe3

Die Teilaufgabe3 ist im Subpackage "p3" bzw. "a1.p3" zu lösen.

In einem Friseur-Salon gibt es einen Friseur, einen Frisier-Stuhl und einen Warteraum mit  $n$  weiteren Stühlen. Wenn keine Kunden da sind sitzt der Friseur im Frisier-Stuhl und schläft. Kommt ein Kunde, dann

- weckt er den Friseur (falls dieser schläft), setzt sich auf den Frisier-Stuhl und lässt sich frisieren, oder
- setzt sich ins Wartezimmer (falls der Friseur gerade frisiert und ein Stuhl im Wartezimmer frei ist) und wartet dort bis er dran ist (FIFO-Prinzip), oder
- geht wieder (falls der Friseur gerade frisiert und kein Stuhl im Wartezimmer mehr frei ist)