



## 1. Übungsblatt zur Vorlesung Einführung in die Informatik

### Aufgabe 1: Daphne

Melden Sie sich bei Daphne für die Vorlesung an:

- Loggen Sie sich mit ihrem Rechenzentrumsaccount (der Benutzername hat etwa die Form ab123) bei Daphne ein.
- Belegen Sie dort diese Vorlesung.
- Sie erhalten dort unter anderem zwei URLs: einen für Subversion, einen für Jenkins. Diese werden Sie später brauchen.

### Aufgabe 2: Subversion

Richten Sie das *Subversion* (kurz: SVN) Versionskontrollsystem ein, mit dessen Hilfe Sie ihre Abgaben hochladen sollten.

- Installieren Sie einen SVN Client. (Eine genauere Anleitung finden Sie im Forum.)

Der Befehl

```
svn checkout --username <ihr RZ-Benutzername> <ihr daphne svn pfad>
```

erstellt eine „working copy“ im aktuellen Verzeichnis auf ihrem PC, das heißt eine Kopie des Serververzeichnisses, die dann via SVN abgeglichen werden kann.

- Legen Sie darin ein Verzeichnis uebung01 (Achtung: Kleinschreibung beachten, kein Leerzeichen) an, mittels

(in das Verzeichnis ab123 wechseln)

```
mkdir uebung01
```

```
svn add uebung01
```

```
svn commit -m "Verzeichnis für Übung 1 hinzugefügt"
```

### Aufgabe 3: Java Development Kit

- Installieren Sie bei Ihnen JDK 1.7. Im Forum finden Sie diesbezüglich Links und eine Beschreibung und können gegebenenfalls Fragen stellen.
- Schreiben Sie mit einem Texteditor ein Java-Programm, das „Hallo Welt!“ ausgibt. Speichern Sie es in das Verzeichnis `uebung01`. Kompilieren Sie es mit dem Befehl

```
javac <Dateiname mit .java>
```

und führen Sie es mit dem Befehl

```
java <Name der Hauptklasse im Programm, d.h. Dateiname ohne .java>
```

aus, um seine Funktion zu überprüfen.

### Aufgabe 4: Jenkins

Laden Sie das Programm (nur die `.java` Datei) mittels ihres SVN-Clients in Daphne hoch.

```
svn add <Dateiname mit .java>
```

```
svn commit -m "<ihre commitmessage>"
```

Es ist nicht sinnvoll die `.class`-Dateien hochzuladen, da diese automatisch erzeugt werden. Damit das nicht aus Versehen passiert, können Sie das „global ignore“-Pattern um `*.class` erweitern. Unter Linux setzen Sie dazu die Option `global-ignores` in der Datei `~/.subversion/config`; unter Windows können Sie in TortoiseSVN die Option unter Settings/General setzen.

Schauen Sie in Jenkins nach (den Link finden Sie auch in Daphne), ob Fehler beim Kompilieren Ihres Uploads aufgetreten sind. Falls ja, beseitigen Sie diese und laden Sie ihre Abgabe nochmal hoch (`svn add` ist dann nicht mehr nötig, da es pro Datei nur einmal durchgeführt werden muss).

### Aufgabe 5: IO-Klassen, Java Syntax

- Laden Sie die Datei `Addition.java` von der Vorlesungswebsite in das Verzeichnis `uebung01` herunter.
- Führen Sie `svn add` und `svn commit` darauf aus und öffnen Sie das Webinterface von Jenkins (in Daphne). Sehen Sie sich die Fehlermeldungen an, die Jenkins zu dieser Datei liefert.
- Versuchen Sie die Datei zu kompilieren. Da die Datei `Prog1Tools.IOTools` importiert, müssen Sie die entsprechende Bibliothek herunterladen und den Java-Classpath entsprechend setzen. (Jenkins ist bereits entsprechend konfiguriert.) Den Link zum Download und einer entsprechenden Anleitung finden Sie auf der Vorlesungswebsite.
- Auch jetzt sollte die Datei noch nicht kompilierbar sein. Finden und beheben Sie die Fehler, so dass ein Programm entsteht, das zwei Ganzzahlen von der Kommandozeile einliest und deren Summe ausgibt.