



Softwarepraktikum im Grundstudium WS 2004/05

Webtechnologien

Aufgabenblatt 1

Testat am 29.10.2004

21.10.2004

1. Aufgabe: Warmwerden

Führen sie zunächst `/home/spgs04/lib/bin/i3init` in ihrem Homeverzeichnis aus und starten sie die Konsole neu.

Erstellen Sie ein Java-Programm `Hello.java`, das den Text „Hello World!“ ausgibt. Übersetzen Sie das Programm mit `javac` und führen Sie es mit `java` aus. Legen Sie Ihren Quellcode ins Verzeichnis `/home/$USERNAME/spgs0405/Uebung01` (alternativ für diesen Pfad wird `~/spgs0405` in der Aufgabenstellung verwendet).

2. Aufgabe: Warmwerden

Erstellen Sie ein weiteres Programm `Pyramide.java`, welches folgende Ausgabe erzeugt:

```
1
1 2 1
1 2 3 2 1
1 2 3 4 3 2 1
```

Übersetzen Sie das Programm mit `javac` und führen Sie diese mit `java` aus.

3. Aufgabe: Repository anlegen

Erstellen Sie mit `cvs -d ~/spgs0405/Repository.Local init` ein lokales CVS-Repository. Erstellen sie ein neues Modul und fügen Sie gleichzeitig ihre Quellcode-Dateien in das Repository ein.

```
cvs -d ~/spgs0405/Repository.Local
import -I "*.class" Uebung01 NAME start
```

Der Befehl muss **im** Verzeichnis `~/spgs0405/Uebung01` ausgeführt werden.

Was macht der Befehl? Wieso ist er sinnvoll?

Jetzt stehen die Dateien im Repository, aber noch steht das Verzeichnis `Uebung01` nicht unter Versionskontrolle. Woran erkennt man das? Erzeugen Sie eine Lokale Arbeitskopie des Moduls `Uebung01` im Verzeichnis `~/spgs0405` .

```
cvs -d ~/spgs0405/Repository.Local checkout Uebung01
```

4. Aufgabe: Verzeichnisse und Binärdateien

Erstellen Sie ein Unterverzeichnis `res` im Übungsverzeichnis und verschieben Sie die Dateien `readme.txt` und `readme.pdf` aus `~/res_spgs0405` dorthin. Fügen Sie zuerst mit `cv`s **add** `res` das Verzeichnis zum Repository hinzu. Anschließend sollen die beiden Dateien zum Repository hinzugefügt werden, die PDF-Datei allerdings als Binärdatei.

```
cv
```

s **add** -kb `readme.pdf`

```
cv
```

s **add** `readme.txt`

Diese Befehle merken die Datei nur als hinzugefügt vor. Führen Sie deshalb ein `cv`s `update` und ein `cv`s `commit` aus.

Wieso sollte man die PDF-Datei als Binär-Datei einchecken?

5. Aufgabe: Update / Commit

Erweitern Sie das in Aufgabe 2 erstellte Programm um eine ausführliche Kommentierung. Geben Sie insbesondere einen Datei-Header an, der Ihren Namen, das Änderungsdatum, den verwendeten Compiler und eine Beschreibung der Funktionalität enthält.

Prüfen Sie mit `cv`s `update`, ob eine neuere Version ihres Programms im Repository liegt, falls dies nicht der Fall ist, spielen Sie `cv`s **commit** -m „Commit Message“ die Änderungen ins Repository ein. Geben Sie in der commit message eine nachvollziehbare Beschreibung der Änderungen an.

Führen Sie erneut `cv`s `update` aus. Geben Sie an, was die Zeichen (U, ?, A, R, C) vor den Dateinamen bedeuten.

6. Aufgabe: CVS-Schlüsselwörter

Fügen sie in Ihrem Quellcode in der ersten Kommentarzeile `$Id$` ein. Machen sie ein `update` und `commit`, öffnen Sie die Datei erneut, um die Ersetzung zu erkennen.

7. Aufgabe: Informationen aus CVS anzeigen

Verwenden Sie den Befehl `cv`s **log** `Pyramide.java`, um sich die gespeicherten Informationen zu der Datei `Pyramide.java` anzeigen. Vergleichen Sie die Ausgabe mit der des Befehls `cv`s **annotate** `Pyramide.java`

8. Aufgabe: Hilfe zu CVS

Verwenden Sie das Programm `info cv`s, um herauszufinden, wie man eine Datei aus dem Repository löscht. Erläutern sie den Unterschied zwischen dem Löschen in einer lokalen Kopie und im Repository.

9. Aufgabe: Grenzen von CVS

Prüfen Sie, welche Möglichkeiten es gibt, eine Datei umzubenennen.

10. Aufgabe: Gemeinsames Repository

Legen Sie sich im Verzeichnis ~/spgs0405 ein Unterverzeichnis Work.Common an und erzeugen Sie in diesem Verzeichnis eine private Arbeitskopie des Moduls Aufgaben aus dem Repository.Common. Dazu müssen sie sich zuerst beim pserver identifizieren:

```
cvs -d :pserver:GruppeXY@cvs-i3.informatik.rwth-  
aachen.de:/home/projects/spgs0405/Repository.Common login
```

Anschließend kann die lokale Arbeitskopie erzeugt werden:

```
cvs -d :pserver:GruppeXY@cvs-i3.informatik.rwth-  
aachen.de:/home/projects/spgs0405/Repository.Common checkout Aufgaben
```

Jedes Repository enthält ein Modul CVSR00T, in dem Administrationsdateien abgelegt sind. Die Administration des Repository erfolgt durch Änderungen der Arbeitskopien und anschließendes einchecken.

Das Verzeichnis ~/spgs0405/src/ wird Ihr privates Arbeitsverzeichnis. Checken Sie das Modul CVSR00T Ihres Gruppen-Repository in diesem Verzeichnis aus.

```
cvs -d :pserver:GruppeXY@cvs-i3.informatik.rwth-  
aachen.de:/home/projects/spgs0405/Repository.GruppeXY login
```

```
cvs -d :pserver:GruppeXY@cvs-i3.informatik.rwth-  
aachen.de:/home/projects/spgs0405/Repository.GruppeXY checkout CVSR00T
```

Ändern Sie die Konfiguration (in der Datei logininfo) so, dass die Commit-Messages an alle Mitglieder Ihrer Arbeitsgruppe versendet werden. Anschließend müssen Sie wieder ein update und commit machen.

Ihre Lösung der ersten Übung soll jetzt von ihrem lokalen Repository ins globale verschoben werden. Leider sind da einige Klimmzüge zu machen, da CVS es so nicht unterstützt.

```
cvs -d ~/spgs0405/Repository.Local release -d Uebung01
```

```
cvs -d ~/spgs0405/Repository.Local export -r HEAD -d nonVersionControlled  
Uebung01
```

```
cd nonVersionControlled
```

Auch hier müssen Sie sich wieder einloggen.

```
cvs -d :pserver:GruppeXY@cvs-i3.informatik.rwth-  
aachen.de:/home/projects/spgs0405/Repository.GruppeXY login
```

```
cvs -d :pserver:GruppeXY@cvs-i3.informatik.rwth-  
aachen.de:/home/projects/spgs0405/Repository.GruppeXY import Uebung01  
GRUPPE NAME
```

```
cd ..
```

```
rm -rf nonVersionControlled
```

```
cd ~/spgs0405/src
```

```
cvs -d :pserver:GruppeXY@cvs-i3.informatik.rwth-  
aachen.de:/home/projects/spgs0405/Repository.GruppeXY checkout Uebung01
```

Erklären sie die einzelnen Befehle.

Nutzen Sie Ihr Repository zum gemeinsamen Erstellen der Lösungen. Die Version im Repository ist maßgeblich für das Testat.