

SWT1 - Kennenlerntutorium

Tutorium Nr. 17

Kay Schmitteckert | 24.04.2015

INSTITUT FÜR PROGRAMMSTRUKTUREN UND DATENORGANISATION (IPD)



- 1 Organisatorisches
 - Vorstellungsrunde
 - Übungsbetrieb & Klausur

- 2 Tools
 - Git
 - Maven
 - Checkstyle

- 3 Übungsblatt
 - JUnit Modultests
 - Codebeispiel

Über mich

- Kay Schmitteckert
- Informatik
- 4. Semester
- E-Mail: uheat@student.kit.edu
- Folien: github.com/kay-es/swt1

- Name
- Studiengang und Semester
- Hobbys
- Erfahrungen mit Softwareentwicklung
- Erwartungen und Wünsche ans Tutorium

- lockere Atmosphäre
- Rücksichtnahme
- aktive Teilnahme
 - **Mitdenken**
 - **Fragen stellen**

- Tutorium alle 14 Tage (**nächstes Tut am 07.05.2015**)
- 6 Übungsblätter
- Abgabe in der Regel Mittwoch 12 Uhr (auch alle 14 Tage)
- insgesamt 150 Punkte
- Übungsschein bestanden mit 50% der Punkte
 - *Anmeldung im Studienportal notwendig!*
- Abgabe:
 - Theorieteil: Holzkiste 3. Stock vor Raum 368 im Infobau
 - Praxisteil: Lösungseinzugszentrale: *<http://lez.ipd.kit.edu>*

- i.d.R 60 Minute Klausur
 - Hauptklausur: 10.08.2015, 14:00 Uhr
 - Nachklausur: 06.10.2015, 11:00 Uhr
- *Anmeldung auch übers Studienportal*

Wichtige Befehle

<code>git init</code>	initialisiert ein Git-Depot im aktuellen Ordner
<code>git add <i>Dateiname</i></code>	fügt eine Datei zur Versionsverwaltung hinzu
<code>git log</code>	gibt alle Änderungen in chronologischer Reihenfolge aus
<code>git rm <i>Dateiname</i></code>	löscht eine Datei im Verzeichnis
<code>git commit -m <i>Beschreibung</i></code>	bucht Änderungen in das Depot ein

Wichtig!

- kurze aber aussagekräftige Kommentare
- **So kurz wie möglich, so lang wie nötig**
- Einbuchen in sinnvollen Teilschritten - *"neue Klasse hinzugefügt"*
kein guter Eintrag!
- Entscheidet euch für eine Sprache und behaltet sie bei!

Wichtige Befehle

<code>mvn compile</code>	kompiliert die verwendeten Quelltexte zu <code>.class</code> -Dateien
<code>mvn test</code>	kompiliert die Tests und führt sie aus
<code>mvn package</code>	erstellt ein ausführbares <code>.jar</code> -Paket des Projekts, das mit <code>java -jar projekt.jar</code> gestartet werden kann
<code>mvn clean</code>	löscht den Ordner mit den Kompilaten

Kennt ihr aus Programmieren

- Testframework für Java
- bereits in Eclipse integriert
- Anlegen einer Testklasse
 - mindestens ein eigenes Package für Tests
`src/test/java/"PackageName"`
 - mindestens eine Testklasse pro Komponente

Kennzeichnung der Methoden durch Annotationen

@Before	wird vor jedem Testfall ausgeführt
@After	wird nach jedem Testfall ausgeführt
@Test	kennzeichnet einen Testfall
@BeforeClass	Ausführung vor Instanziierung der Testklasse
@AfterClass	Ausführung nach allen anderen Methoden
@Ignore	noch kein Inhalt, Test wird ignoriert

Codebeispiel