

Tipps und Hinweise



TECHNISCHE
UNIVERSITÄT
DARMSTADT

Softwarepraktikum

Tipps

Hier ein paar inoffizielle Hinweise und Tipps zum Praktikum. Diese müssen nicht eingehalten werden.

- Jedes Arbeitsblatt hat mindestens zwei Wochen Bearbeitungszeit: Man sollte in der ersten Woche versuchen, alles zu lösen, damit man beim zweiten freiwilligen Termin Fragen an den Tutor stellen kann, die die Knackpunkte der Aufgaben betreffen.
- Weiterhin werden freiwillige Repetitorien angeboten, die unterstützend besucht werden können.
- Für (fast) jede Klasse, die geschrieben werden soll, findet sich eine Dokumentation in der API des Frameworks, dort sollte von Anfang an auf die geforderten Anforderungen geachtet werden!
- Jede Frontalveranstaltung ist für euch. Die Anwesenheit und aktive Teilnahme wird daher dringend empfohlen, besonders wenn die Musterlösung vorgestellt wird!
- Abgabe am besten am Donnerstagabend. Nicht erst eine Minute vor 0:00 Uhr.
- Man sollte sich frühzeitig Rat bei einem Tutor oder über das Moodle-Forum holen, nicht erst am Donnerstagabend.
- Alte Musterlösungen bringen nichts-> sind oft nicht aktuell und alles abgeschriebene wird erkannt (da bringt es auch nichts, Variablennamen zu ändern).
- Programmieren in der Gruppe ist nicht unbedingt einfacher als alleine: Daher sollte am besten ein erfahrener Programmierer die etwas weniger erfahrenen am Computer sitzen und tippen, sodass diese in ihrem Tempo schreiben, wirklich verinnerlichen, was da geschrieben wird und nachfragen. Man sollte auch immer mal wechseln.
- Jeder in der Gruppe sollte in der Lage sein, die Aufgaben zu lösen. Sich „mitschleifen“ zu lassen rächt sich schnell...
- Zu Beginn des Praktikums sollte man sich schnellstens auf ein grundlegendes Java- Niveau begeben. Folgende Begriffe sollten bis zum AB2 keine Schwierigkeiten machen:
 - Collections, Vector, Iterator, Hashset, Hashtable und Generics
 - Was sind die Unterschiede dieser Speicherklassen?
 - Javadoc und API
 - Interfaces, Klassen, Objekte, Klassen-/Objektmethoden und Variablen, abstract, public, private, protected
 - Wie durchlaufe ich einen Array, einen Vector, etc

- Das ganze Praktikum steht im Kontext des Energie-Versorgungs-Szenarios, also das nicht aus den Augen verlieren. Wozu mache ich das ganze? Was können die Klassen des Rahmenwerkes schon?
- Man sollte sich schnell mit dem Programm Eclipse anfreunden. Was kann das Programm? Was brauche ich davon? Wie kann mir Eclipse helfen? Debugger, Highlighting, Autovervollständigen...

Die wichtigsten Eclipse Tastenkombinationen

Mit den richtigen Tasten kommt man bei Eclipse schneller ans Ziel. Hier der Versuch die wichtigsten herauszusuchen mit denen man sich umständliches Herum klicken ersparen kann.

Grundlagen und Navigation

Shift+F2	Die Kontexthilfe öffnet die API Dokumentation zur Methode oder Klasse die sich gerade unter dem Cursor befindet
Strg+Leertaste	Aktiviert die Autovervollständigung
Strg+1	Kontextsensitive Vorschläge zur Vervollständigung oder Korrektur. Sehr praktisch ist dieses Feature beispielsweise um noch fehlende Deklarationen von lokalen oder Instanz-Variablen zu ergänzen. Das spart viel Tipparbeit.
F3	Navigiert zur Klasse oder Methode unter dem Cursor
F4	Zeigt die Klassenhierarchie zur Klasse unter dem Cursor an
Strg+Alt+H	Zeigt die Aufrufhierarchie für die Methode unter dem Cursor an. So erhält man schnell den Überblick wo eine Methode aufgerufen wird.
Strg+Shift+T	Öffnet eine Klasse: Inkrementelle Suche über den Klassennamen mit Wildcards.
Strg+S	Speichern.
Strg+Shift+O	Organisiert die import statements: Entfernt überflüssige und fügt fehlende imports hinzu. Lässt sich auch rekursiv auf packages oder Projekte anwenden.
Strg+Shift+F	Formatiert den Quelltext.

Refactoring

Alt+Shift+R	Umbenennen der Methode oder Klasse unter dem Cursor
Alt+Shift+M	Extrahiert die markierten Zeilen in eine neue Methode. Um Methodenparameter und Rückgabewert kümmert sich Eclipse automatisch.
Alt+Shift+L	Extrahiert den markierten Bereich in eine neue lokale Variable. Ein gutes Gegenmittel für lange Zeilen mit verschachtelten Aufrufen.
Alt+Shift+S	Kontextmenu für Quellcode Aktionen. Drückt man dann die Taste R, gelangt man zur Generierung von Gettern und Settern.
Alt+Shift+T	Kontextmenu für Refactorings
Crt+K	Zum nächsten Suchtreffer springen (nicht F3 ; diese Funktion ist zwar bei den meisten Programmen über F3 erreichbar, aber bei Eclipse diese Taste anders belegt)

Eine Übersicht über weitere Tastenkombinationen kann man sich mit Strg+Shift+L oder das Menu „Help|Key Assist“ anzeigen lassen (Eclipse 3.2). Dazu passend sei noch eine Windows Tastenkombination erwähnt: Drückt man Alt+Shift, so ändert man das Tastaturlayout von Deutsch auf Englisch beziehungsweise umgekehrt. Das passiert einem manchmal aus Versehen in Eclipse, besonders wenn man die Tastenkombinationen für Refactorings einsetzt.