

SWT1: Vorbereitung und Aufwärmen

(Programmierungsumgebung einrichten)

Ausgabe: Montag, 20. April 2020

Abgabe: Sonntag, 26. April 2020, 23:59 Uhr

Dies ist kein Pflichtübungsblatt. Sie brauchen aber für die kommenden Übungsblätter eine funktionierende Entwicklungsumgebung, die Sie mit Hilfe dieser Anleitung einrichten und ausprobieren. Dafür erhalten Sie bis zu 3 Bonuspunkte.

Der Einsatz fortschrittlichster Recherchewerkzeuge, wie z. B. Google oder Bing, ist ausdrücklich erlaubt. 😊

Bitte melden Sie sich an der LEZ mindestens einmal an: <https://lez.ipd.kit.edu>. Nur dann kann Ihr Tutor Ihre Punkte erfassen.

Entwicklungsumgebung und Quelltextqualität: Eclipse, CheckStyle, JUnit

- a) Installieren Sie Eclipse (*for Java Developers*): <https://www.eclipse.org/downloads/packages>
(Verwenden Sie für alle Projekte Java 14)
- b) Ergänzen Sie diese Installation um CheckStyle: <https://checkstyle.org/eclipse-cs>
- c) Ergänzen Sie diese Installation um Eclemma: <https://www.eclemma.org>
- d) Installieren Sie Git: <https://git-scm.com/downloads>

Maven in 5 Minuten

Machen Sie sich mit Maven vertraut; lesen Sie hierzu <https://maven.apache.org/guides/getting-started/maven-in-five-minutes.html>. Mindestens folgenden Begriffe und die zugehörigen Konzepte sollten Ihnen geläufig sein: `pom.xml`, `mvn package`, `mvn clean`, `mvn test` und `mvn install`. Sie werden Maven als Eclipse-Plugin verwenden, daher müssen Sie es nicht separat herunterladen (Sie dürfen natürlich 😊: <https://maven.apache.org/install.html>)

Versionskontrolle: Git

Machen Sie sich mit dem Versionskontrollsystem Git vertraut; lesen Sie dazu <https://git-scm.com/book>. Mindestens folgende Befehle und die zugehörigen Konzepte sollten Ihnen geläufig sein: `git init`, `git add`, `git commit`, `git log`, `git checkout`, `git ignore`.

Aufgabe 1: Teilschritte mit Maven automatisieren (3 Bonuspunkte)

Lernen Sie Maven in einem ersten praktischen Einsatz kennen und verstehen Sie, wie Maven mit Ihrem Quelltext interagiert.

- a) Laden Sie sich das Maven-Testprojekt `Lez-test-dummy` aus Ilias herunter und entpacken Sie es.
- b) Importieren Sie es als existierendes Maven-Projekt in Eclipse.
- c) Konfigurieren Sie CheckStyle in Eclipse so, dass es ausschließlich die im Projekt hinterlegte Konfigurationsdatei `src/test/resources/checkstyle_swt1.xml` verwendet.
- d) Analysieren Sie das Projekt mit CheckStyle. Beheben Sie die Fehler.
- e) Bearbeiten Sie die `pom.xml` und passen Sie `ArtifactId` und `GroupId` an. **Beachten Sie hierzu die Hinweise am Ende des Vorbereitungsblattes.**
- f) Implementieren Sie die Methoden der Klasse `EditMe` entsprechend ihrer Kommentare sowie die zugehörigen Testfälle in `EditMeTest`.
- g) Erstellen Sie in Eclipse eine neue „Run Configuration“. Es soll Maven mit dem Goal „package“ ausgeführt werden. Dies entspricht dem Kommandozeilenaufwurf `mvn package`. Maven erstellt daraufhin eine ZIP-Datei mit den Quelltextdateien. Übertragen Sie diese Datei in die LEZ. Übertragen Sie *nicht* die ausführbare JAR-Datei, sondern die ZIP-Datei, die die Quellen des Programms enthält.

Hinweis zu den Werkzeugen

Git ist das in den Übungsaufgaben und in den Tutorien verwendete Werkzeug zur Versionskontrolle. Eclipse ist die in der Vorlesung und in den Tutorien verwendete Programmierumgebung. Sie können auch eine andere Programmierumgebung einsetzen, wenn Sie damit ausreichend vertraut sind und die gestellten Aufgaben genauso bearbeiten können. Es werden keine Aufgaben ausgegeben, deren Lösungen Eclipse-spezifische spezielle Fähigkeiten benötigen. Sie müssen „nur“ in einer modernen Entwicklungsumgebung programmieren und Versionskontrolle einsetzen können.

Verwenden Sie bitte für alle Aufgaben Java 14.

Hinweis zu den Programmieraufgaben

Sofern nicht anders angegeben, sind die Aufgaben mit Java zu lösen. Die Einbuchungen und ihre Kommentare in Git werden bewertet (Existenz, sinnvolle Einbuchungshäufigkeit, -zeitpunkte und -dateien, sprechende Kommentare).

Konfigurieren Sie Maven mittels der Datei `pom.xml` **immer** wie folgt, um konsistente Projekte zu erhalten:

1. `ArtifactID` ist Ihre Matrikelnummer.
2. `GroupID` ist „swt1.ub<Übungsblatt-Nr.>.a<Aufgaben-Nr.>“, also z. B. „swt1.ub0.a1“ für Aufgabe 1 des Vorbereitungsblatts oder „swt1.ub3.a2“ für Aufgabe 2 des dritten Übungsblatts.
3. Ändern Sie keine Einstellungen, die im XML-Dokument mit Kommentaren von uns versehen wurden (bspw. den Bereich `DependencyManagement`). Änderungen an diesen Bereichen können die automatische Analyse Ihrer Programme verhindern – in so einem Fall müssen Sie mit Punktabzug rechnen.
4. Wenn Sie Quelltexte abgeben, erzeugen Sie die Abgabedatei immer mit dem Befehl `mvn package`; laden Sie nicht die Binärfassung des Projekts hoch (*.jar), sondern die ZIP-Datei mit den Programmquellen. **Lösungen ohne Quelltext können nicht bewertet werden!**

Hinweis zur Lösungseinzugszentrale (LEZ)

Die Abgabe erfolgt per Lösungseinzugszentrale (LEZ). Sie können sich bei der LEZ mit Ihrem SCC-Benutzerkonto anmelden (<https://lez.ipd.kit.edu>). Sie können pro Aufgabe eine einzelne Datei, ggf. auch ein gepacktes Archiv im ZIP-Format, abgeben. Soweit nicht anders angegeben, ist für die Programmieraufgaben hier immer `mvn package` auszuführen und die resultierende ZIP-Datei mit den Quelltextdateien Ihrer Lösung zu übertragen. Achten Sie darauf, dass Sie eine signierte E-Mail der LEZ erhalten, in welcher eine Prüfsumme Ihrer abgegebenen Lösung enthalten ist. Diese dient bei technischen Problemen als Nachweis der Abgabe.

Hinweis zur Einzelbewertung

Die Lösungen aller Übungsblätter sind Einzelleistungen. Plagieren von Lösungen führt zum Abzug von 25 Punkten bei allen am Plagiat Beteiligten.

Beispiel: Piggeldy lässt Frederick Aufgabe 2 abschreiben, ansonsten bearbeiten sie Übungsblatt 1 aber getrennt. Beide erhalten nun 25 Minuspunkte sowie keine Punkte für die übrigen Aufgaben von Blatt 1.

Hinweis zu Aktualisierungen des Übungsblatts

Verfolgen Sie Nachrichten in ILIAS und prüfen Sie es regelmäßig auf Aktualisierungen und Korrekturen des aktuellen Übungsblatts.