

# ATESTEO TestTool

## Zusatzinformationen

### 1 Workspace einrichten

Das gesamte Projekt befindet sich auf dem Bitbucket Server im Projekt namens „TestTool“ (Kürzel TTJ). Darin liegt der Ordner „AtesteoTestTool“, welcher ein Eclipse Workspace mit mehreren Java Projekten ist.

Die Projekte im Workspace (Stand August 2016) werden in Tabelle 1 kurz erläutert.

Projekt	Beschreibung	Anmerkung
TestTool	Kernfunktionalität	
TestDesignerGUI	GUI für Testerstellung (JavaFX)	verwendet Projekt TestTool
TestExecuterGUI	GUI für Testdurchführung (JavaFX)	verwendet Projekt TestTool
KeywordAnnotations	Enthält die Annotations @KeywordLibrary und @Keyword	wird von Projekten für Bibliotheken referenziert
TCULibrary	Bibliothek mit Schlüsselwörtern für die TCU IV	Verwendet Projekt HttpLibrary
HttpLibrary	Enthält Methoden für HTTP Requests	
CompareKeywordLibrary	Bibliothek für Vergleiche	
DelayKeywordLibrary	Bibliothek für Verzögerungen	
DialogKeywordLibrary	Bibliothek für Dialoge	

*Tabelle 1: Projektbeschreibungen*

Um den Workspace einzurichten wird Eclipse gestartet und der Ordner „AtesteoTestTool“ (vorher mit Git klonen) als Workspace angegeben. Anschließend müssen die Projekte importiert werden.

Kontextmenü: *Import... > General > Existing Projects into Workspace*

Anschließend können Projekte bearbeitet oder neue hinzugefügt werden. Dabei ist zu beachten, dass manche Projekte Abhängigkeiten zu anderen Projekten oder JARs haben.

Kontextmenü: *Build Path > Configure Build Path > (Tab) Projects oder (Tab) Libraries*

## 2 Bibliothek erstellen

Um eine Bibliothek zu erstellen wird im Workspace ein neues Java Projekt erstellt. Anschließend wird eine Abhängigkeit zum „KeywordAnnotations“ Projekt hinzugefügt, da die Annotations `@KeywordLibrary` und `@Keyword` für die Bibliothek benötigt werden werden.

Kontextmenü: *Build Path > Configure Build Path > (Tab) Projects > Add...*

*Alternativ: Projekt „KeywordAnnotations“ als JAR exportieren und dann in neuem Projekt referenzieren*

Anschließend wird die Bibliothek wie ein normales Java Projekt entwickelt. Die Bibliotheks-Klasse muss mit der Annotation `@KeywordLibrary` versehen. Sie muss ebenfalls alle Methoden für Schlüsselwörter enthalten, welche mit der Annotation `@Keyword` versehen werden müssen.

## 3 Bibliothek exportieren

Eine Bibliothek wird immer als JAR Datei exportiert.

Kontextmenü: *Export... > Java > JAR file*

Dabei ist zu beachten, dass vom Projekt referenzierte JAR Dateien in die zu exportieren JAR integriert werden. Dies sollte am besten kontrolliert werden, da dieser Fehler erst bei der Testdurchführung auffällt. Außerdem muss der Dateiname der JAR die Namenskonvention beachten.

*Konvention: Pakete durch Punkte getrennt vor den Klassennamen der Bibliotheks-Klasse setzen. (z.B. „com.example.FooLibrary“ wobei FooLibrary die Klasse mit der @KeywordLibrary Annotation ist)*

Die JAR Datei wird im Ordner für Bibliotheken abgelegt, welcher sich standardmäßig „Eigene Dokumente / AtesteoKeywordTesttool / libs“ ist. Wenn die Bibliothek häufig/fast immer benutzt wird, dann kann sie im Unterverzeichnis „std“ abgelegt werden. Dadurch können die Schlüsselwörter ohne explizite Referenzierung der Bibliothek in einer Testdatei verwendet werden.

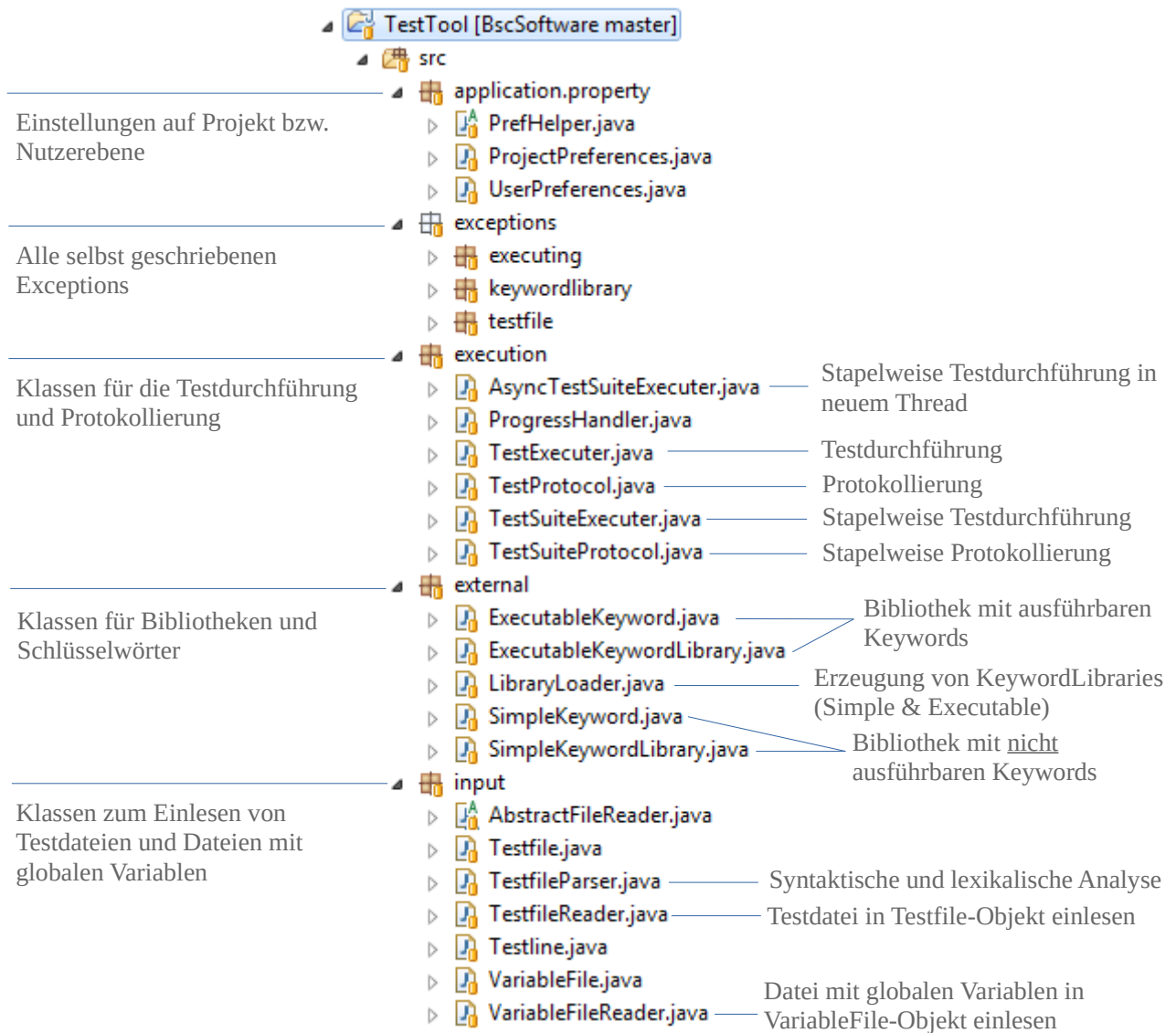
*Anmerkung: Damit neue Bibliotheken erkannt werden muss die GUI der Testerstellung neu gestartet werden. Die GUI der Testdurchführung muss nicht neu gestartet werden.*

## 4 Werkzeuge

Werkzeug	Version
Java Development Kit	1.8.0_91
Eclipse IDE	Neon Release 4.6.0
Git	2.9.0 Windows
Gluon Scene Builder	8.2.0

## 5 Kernprojekt: TestTool

Dieses Projekt enthält den Kern des gesamten Softwareprojekts. Also alles vom Einlesen über die Testdurchführung bis hin zur Testprotokollierung sowie Einstellungen. Die Projekte TestDesignerGUI und TestExecuterGUI enthalten lediglich Programmcode für die graphischen Oberfläche. Alles andere wird aus der Kernfunktionalität, also dem Projekt TestTool, bezogen. Nachfolgend werden die Packages/Dateien des Hauptprojekts kommentiert.



Anmerkung: Es gibt ausführbare (ExecutableKeyword) und nicht-ausführbare Keywords (SimpleKeyword). Die erste Art wird bei der Testdurchführung benötigt, da die zugehörige Programmlogik ausgeführt werden soll. Die zweite Art wird bei der Testerstellung verwendet. Die beiden unterscheiden sich dadurch, dass in einem ausführbaren Keyword eine Instanz der Bibliothek gespeichert ist. Das bedeutet, dass der Konstruktor der Bibliotheks-Klasse aufgerufen werden muss. Da dies unter Umständen erst zum Zeitpunkt der Testausführung funktioniert gibt es nicht-ausführbare Keywords, die lediglich Informationen speichern und keinen Konstruktoraufruf erfordern.