

Benutzerhandbuch des Codegenerators für Lebenslaufakten

Inhaltsverzeichnis

1. Einleitung
2. Vorstellung der verwendeten Tools
3. Funktionalität des Programms
4. Hinweise zur Verwendung
 - 4.1. UML-Standard
 - 4.2. abschließende Hinweise
5. Anleitung zur Installation
 - 5.1. Vorbereitung der Installation
 - 5.2. Installation
 - 5.3. Verwendung der Website
 - 5.3.1 Grundlegendes
 - 5.3.2 Hintergrundwissen
 - 5.3.3 Beschreibung der Webseite
6. FAQ
7. Glossar
8. Entwicklerteam und Support

1. Einleitung

Da Sie dieses Benutzerhandbuch lesen, ist anzunehmen, dass sie Betreiber oder Mitarbeiter einer Gesellschaft sind, die ein Kraftwerk basierend auf erneuerbaren Energien verwaltet oder besitzt. Sie wissen also, welcher immenser Aufwand hinter der Wartung und Modernisierung solcher Anlagen steht. In diesem Sinne liegt es uns fern, Ihnen noch weiter aufzuzeigen, wie wichtig eine gute Koordinierung der Daten einer Lebenslaufakte ist. Uns ist vielmehr daran gelegen, Ihnen eine Möglichkeit zu bieten, diese Vorgänge zu vereinfachen. Der hier nun vorgestellte und erklärte Codegenerator wird Ihnen helfen, aus UML-Modellen eine entsprechende Datenbank und eine Website zu generieren und damit dann Ihre Dokumente leichter zu ordnen.

Im nun folgenden Benutzerhandbuch werden folgende Dinge behandelt:

- *Vorstellung der verwendeten Tools:*

In diesem Abschnitt werden die Programme und Bibliotheken vorgestellt, welche der Codegenerator benutzt. Es wird erklärt werden, welchen Nutzen diese besitzen und warum genau diese ausgewählt wurden. Dieser Teil des Benutzerhandbuch wird also für den Endbenutzer, welcher Sie evtl. sind, eher informativen Charakter haben, aber der das System zukünftig pflegende Mitarbeiter wird diesen Abschnitt mit Sicherheit nützlich finden, da er der Einarbeitung in das System dient.

- *Funktionalität des Programms*

Hier wird Ihnen erklärt welche Funktionen unser Programm erfüllt.

- *Hinweise zur Verwendung:*

Die in diesem Teil folgenden Informationen dienen der präventiven Klärung von Problemen und natürlich auch der Abgrenzung von gewissen Features. Es wird dementsprechend hier geklärt werden, welche Form das UML-Modell besitzen darf, damit es korrekt eingelesen und verarbeitet werden kann. Des Weiteren definiert dieser Abschnitt einige nicht unterstützte Funktionen und erklärt, warum diese Funktionen von uns nicht mit eingebracht worden sind. Für den das Programm pflegenden Mitarbeiter oder den Support wird hier geklärt warum verschiedene Dinge nicht funktionieren werden und bei einigen Punkten werden dafür Lösungsvorschläge eingebracht. Außerdem finden hier Entwickler, welche dieses Programm ausbauen wollen, indirekt Vorschläge, was noch implementiert werden könnte.

- *Anleitung zur Installation :*

Hier werden Sie nun durch eine Reihe von Schritten geführt, welche vom Vorbereiten der Installation bis hin zu Ihrem ersten Testbeispiel reichen. Dieser Bereich wird speziell auf nicht technisch-versierte Personen ausgerichtet sein, um auch diesen den Einstieg in das Programm zu erleichtern. Es wird allerdings auch den technischen Mitarbeitern empfohlen, diesen Bereich durchzuarbeiten, um möglichen Problemen vorzubeugen.

In der Hoffnung Ihnen hiermit genügend Überblick über das Handbuch gegeben zu haben, wünschen wir Ihnen nun gutes Gelingen.

2. Vorstellung der verwendeten Tools

- **Eclipse IDE Enterprise Edition von Juno, Version 1.5.2.20130211-1820**
Eclipse ist ein open source Programmierwerkzeug zur Entwicklung von Software verschiedenster Art. Für Eclipse gibt es viele verschiedene Erweiterungen. Es basiert auf JAVA.
Wurde zur Programmierung und zum starten des Generators inklusive aller Plugins verwendet.
- Plugins:
 - o **Xpand SDK, Version 1.2.1.v201206110941**
Xpand ist eine statisch typisierte Templatesprache. "Templates können polymorph aufgerufen werden und durch Aspekte erweitert bzw. verdeckt werden"(Quelle: Wikipedia). Xpand basiert auf EMF models.
Xpand erfordert das man ein EMF metamodel definiert und ein oder mehrere templates, die das Model in text umwandeln.
Genutzt um alle templates zu schreiben in denen definiert wird wie die Webseite aufgebaut ist und die aus dem UML JAVA-Klassen
 - o **UML2 Extender SDK, Version 4.0.2.v20130114-0902**
Genutzt um ein UML Metamodel für unsern Generator bereitzustellen und somit die UML2 Definitionen zu nutzen
 - o **Jetty, Version 8.1.3.v20120522**
Jetty ist ein in Java geschriebener Servlet/JSP-Container und Webserver. Es ist dank seiner geringen Größe leicht möglich es in eine Software zu integrieren. Jetty nutzt den JSP-Compiler Jasper des Apache Tomcat.
Genutzt für den Aufbau unseres Servers
- **Maven 3.0.5**
Maven ist ein auf Java basierendes Build-Management-Tool der Apache Software Foundation. Mit ihm kann man Programme erstellen und verwalten
In der pom.xml werden Softwareabhängigkeiten angegeben. Diese Abhängigkeiten werden von Maven aufgelöst aus einem lokalen Verzeichnis oder einem aus dem Internet, dem Maven- Repository, bereits vorhanden sind. Sind sie es, verwendet Maven z. B. beim Kompilieren die lokal vorhandene Datei von dort, also *ohne* sie in das Projektverzeichnis zu kopieren. Wird bei uns für Tapestry benötigt.

Maven Struktur:

- *src:* (Eingabedateien)
 - o *src/main:* Erstellung des eigentlichen Produkts
 - o *src/main/java:* Java-Quelltext
 - o *src/main/resources:* andere benötigte Dateien
 - o *src/test:* Eingabedateien für Tests
- *target:* erzeugte Dateien
 - o *target/classes:* kompilierte Java-Klassen

3. Funktionalität des Programms

Unser Programm ist eine CRUD Anwendung d.h. eine Anwendung, die die Funktionen Create, Read, Update und Delete erfüllt.

Auf Basis eines UML-Modells wird in unserem Programm die Datenbank und die Webanwendung erstellt. Dabei werden sämtliche Restriktionsbedingungen gewahrt. Der Benutzer benutzt nur die Webanwendung über die er die Daten verwalten kann. Die Navigation in der Webanwendung ermöglicht einen Zugang von allen Seiten zu allen anderen Seiten über möglichst wenig anderen Seiten. Es gibt eine Anzeige einer Gesamtübersicht zur Ansicht aller relevanten Daten

- Create

In der Webanwendung ist das Anlegen von Objekten gemäß seiner eingegebenen Spezifikationen möglich, dabei werden die Restriktionsbedingungen gewahrt.

○ Instanzen anlegen

Insbesondere werden Instanzen angelegt. Ebenso ist es möglich, Attribute entsprechend der vorgegebenen Datentypen zu belegen.

▪ Restriktionsbedingungen beim Anlegen von Instanzen

Beim Anlegen von Instanzen werden klassenspezifische Restriktionsbedingungen (Existenzabhängigkeit) entsprechend des UML-Diagramms beachtet (u.a. betrifft dies Aggregatklassen in Kompositionsbeziehungen). Diese sind entsprechend überprüft worden und bei Anlegen einer Instanz in unzulässiger Weise, wird die Operation zurückgewiesen werden.

▪ Unterscheidung zwischen zwingenden und optionalen Attributen

Wenn im UML Diagramm zwischen optionalen und zwingenden Attributen unterschieden wird, so sind die gültigen Belegungen zwingender Attribute immer gewährleistet.

○ Relationen anlegen

In der Webanwendung können Relationen entsprechend des UML-Modells zwischen Instanzen und die Attribute von Relationen definiert und angelegt werden.

▪ Relationsrestriktionen

Entsprechend der Spezifikationen des UML-Modells wird das Anlegen von Relationen entsprechend der Restriktionsbedingungen überprüft und ggf. zurückgewiesen. Dabei wird insbesondere auf Multiplizitäten geachtet.

- Read

Es ist innerhalb der Webanwendung möglich, Objekte, welche gespeichert wurden, anzuzeigen.

○ Instanzen anzeigen

Instanzen werden dabei insbesondere mit Ihren Attributen und Relationen dargestellt.

- Delete

○ Löschen von Objekten

Relationen und Instanzen können gelöscht werden, dadurch kann es zu Zurückweisung kommen hierauf hat der Benutzer zu achten.

- Update

Bei einer Änderungsoperation wird das Objekt neu gespeichert.

○ Ändern von Objekten

▪ Kaskadierung von Operationen

Bei den Bearbeitungsoperationen werden die Integritätsbedingungen gewahrt, dazu werden Operationen ggf. durch den Datenbestand kaskadiert oder zurückgewiesen.

▪ Instanzen bearbeiten

Es ist insbesondere möglich innerhalb der Webanwendung bereits angelegte Instanzen zu bearbeiten, u.a. in Hinsicht auf die Attribute und Relationen. Dabei ist insbesondere auf Restriktionsbedingungen durch Multiplizitäten (beim Löschen) und auf Datentypen der Attribute (Editieren) zu achten.

▪ Relationen bearbeiten

Es ist ebenfalls möglich innerhalb der Webanwendung bereits angelegte Relationen zu bearbeiten, u.a. in Hinsicht auf die beteiligten Instanzen, die ggf. aktualisiert werden müssen.

4. Hinweise zur Verwendung

4.1. UML-Standard

Das zu verwendende UML-Modell muss in Form einer XML/XMI-Datei nach OMG-Standard vorliegen. Der Code-Generator unterstützt UML-Modelle nach UML 2-Standard.

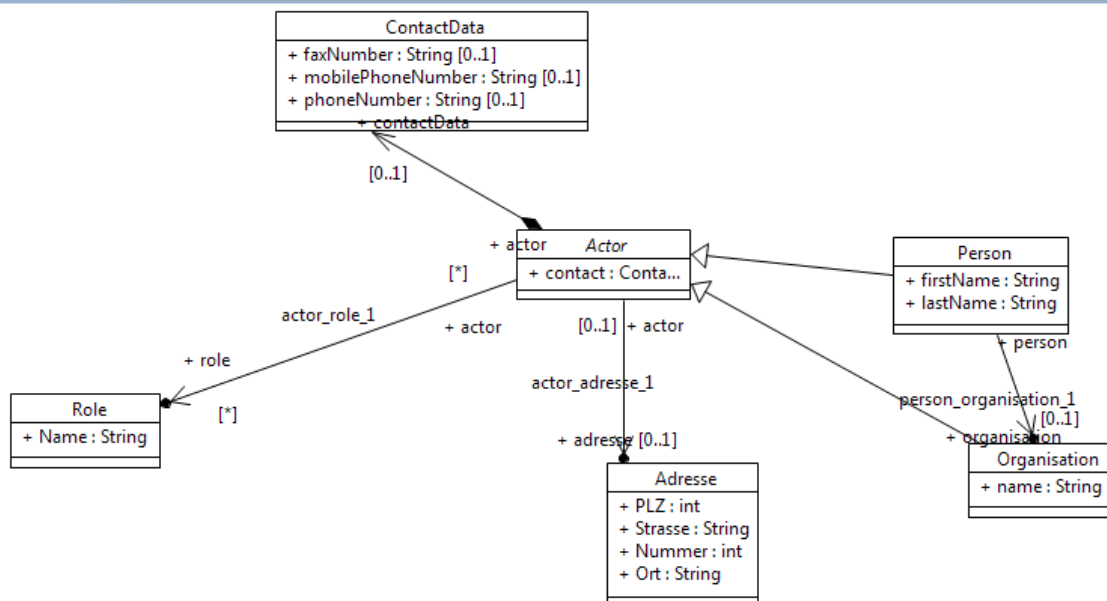
Dabei ist zu beachten, dass im einzulesenden Modell nur die primitiven Datentypen verwendet werden, die im „primitives.profile“ spezifiziert sind. Andere primitive Datentypen sind zu vermeiden.

Bsp.: String, Integer, boolean ...

Weiterhin darf das Modell keinerlei Mehrfachvererbung enthalten, da Java-Quelltext generiert werden soll, und in der Programmiersprache Java diese Art der Vererbung nicht formuliert werden kann.

Relationen auf sich selbst sind ebenfalls zu vermeiden. Bitte für jedes Attribut den Typ angeben.

Ein Beispiel für ein akzeptiertes Model:



4.2. abschließende Hinweise

Zur Versionierung: Dieses Programm ist die erste Version des Programmes. Ziel war es alle Grundfunktionalitäten umzusetzen und zu implementieren. Eine weitere Erweiterung des Programmes kann folgen (z.B. stylesheets für die Weboberfläche).

Erstellung der UMLs: Wir empfehlen die UMLs mit Eclipse zu erstellen. Mit der Eclipse Erweiterung Topcased die auf Papyrus basiert sollte dies keine Probleme darstellen, denn Topcased arbeitet mithilfe eines Baukastenprinzips.

Zur Installation: Bitte achten Sie darauf das sie Eclipse genug Platz auf ihrem Rechner bieten und genug Arbeitsspeicher sonst verlangsamen Sie das Programm unnötig.

Generierungsort: Der generierte Quelltext wird nach erfolgreicher Ausführung des Generators im Ordner "\swplawebsite\src" zu finden sein.

Anforderungen an die Hardware: Bitte beachten Sie weiterhin, dass die tatsächlichen Systemanforderungen bzgl. Prozessor und Arbeitsspeicher stark von der Größe des verwendeten UML-Modells abhängen. Es kann also sein, dass die Leistung Ihres Systems für bestimmte UML-Modelle ausreicht, bei der Abarbeitung größerer Modelle jedoch an seine Grenzen stößt. Daher ist es empfehlenswert, einen Testlauf mit einem UML-Modell der beabsichtigten Größe durchzuführen, bevor die Software produktiv eingesetzt wird.

5. Anleitung zur Installation

5.1. Vorbereitung der Installation

Vor der Installation des Generators ist wichtig, dass Sie sich über einige Punkte im Klaren sind:

Der Generator (bzw. einige der benutzten Tools) werden umso mehr Zeit brauchen, desto größer das Modell ist. Sollte Ihr Modell also extrem groß und/oder komplex sein, sollten Sie dem Generator etwas Zeit geben, um die Informationen entsprechend zu verarbeiten. Ein anderer Punkt ist der freie Speicherplatz. Dieser sollte, gerade wenn Sie noch einige der grundlegenden Programme (Java etc.) installieren müssen, entsprechend groß sein, da Sie Ihren PC sonst überlasten könnten. (In den meisten Fällen wird der vorhandene Speicherplatz zwar sicher ausreichend sein, aber dennoch wollten wir darauf hinweisen.) Der dritte und letzte Punkt ist, dass dieses Programm unter Windows 7 bzw Windows 8 entwickelt wurde. Wir zeigen in diesem Benutzerhandbuch also auch nur, wie dieses Programm in Windows zu installieren ist. Andere Betriebssysteme bzw. ältere Versionen von Windows könnten noch spezielle Vorarbeit benötigen.

Wenn Sie diese Punkte nun überprüft haben, können Sie mit der Vorbereitung der Installation beginnen. Der erste Schritt dazu ist die Überprüfung ob Ihr Rechner an das Internet angeschlossen ist.

Nun folgt der nächste Schritt. Überprüfen Sie ob Sie schon JAVA installiert haben, indem Sie unter Computer -> Programme (oder Programme(x86)) nachsehen ob Sie dort einen Ordner "Java" finden und ob sich in diesem Ordner ein Ordner Java JDK befindet. Wenn dies nicht der Fall ist, dann sollten Sie es hier herunterladen <http://www.oracle.com/technetwork/java/javase/downloads/index.html> und sich das für Ihr System passende Java JDK zu installieren.

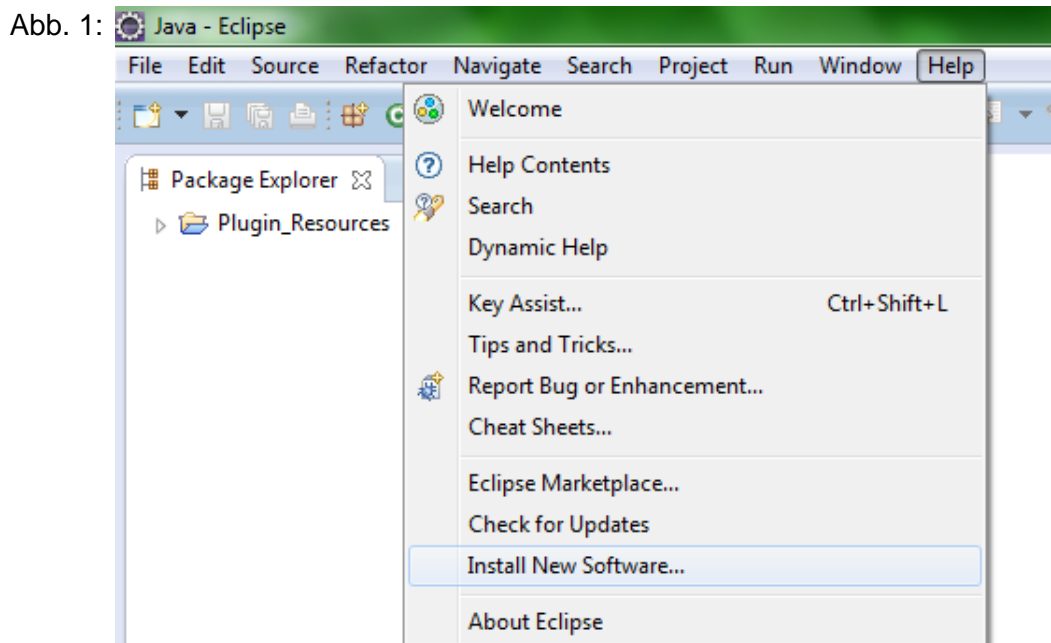
Da Sie nun Java besitzen, folgt der nächste Schritt.

5.2 Installation

Laden Sie sich Eclipse IDE Enterprise Edition von Juno, Version 1.5.2.20130211-1820 herunter.

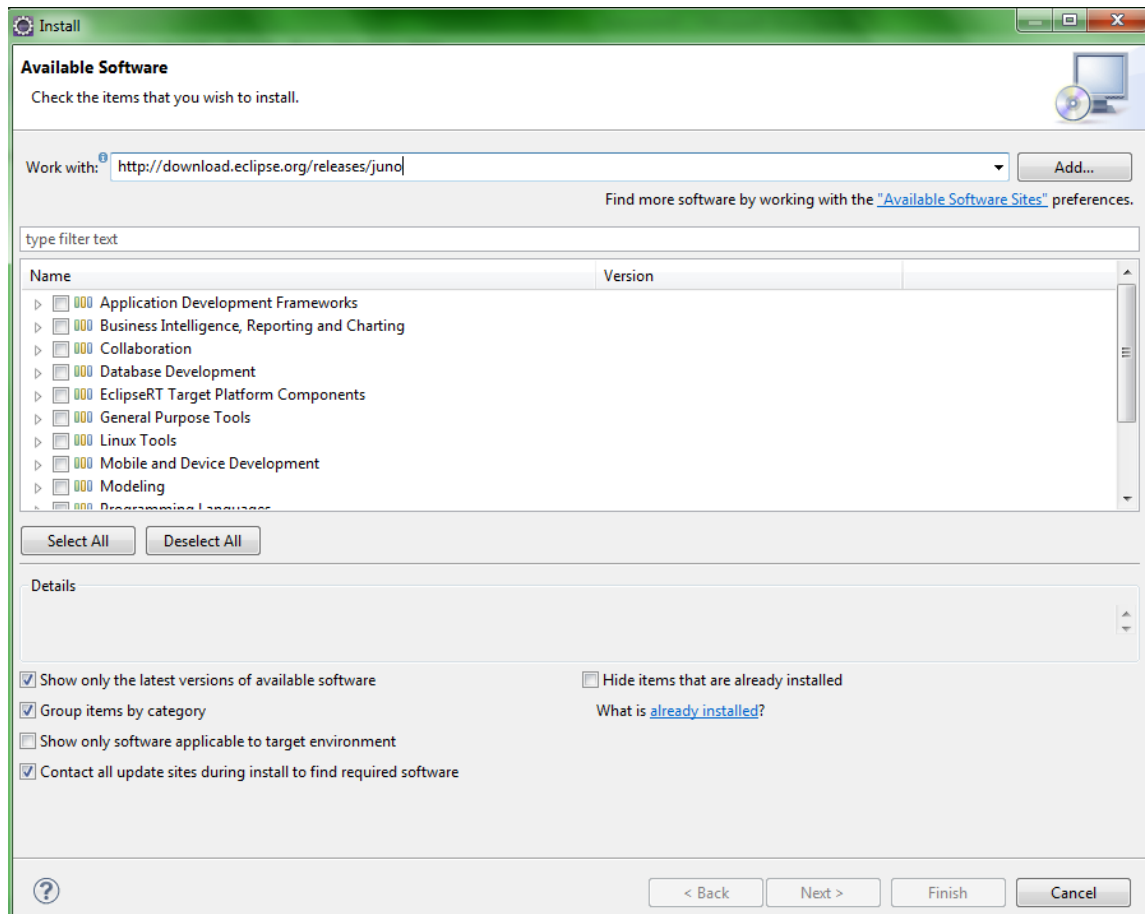
<http://www.eclipse.org/downloads/>

Starten Sie Eclipse und gehen Sie unter 'Help' -> 'install new software' wie in Abb.1 gezeigt.



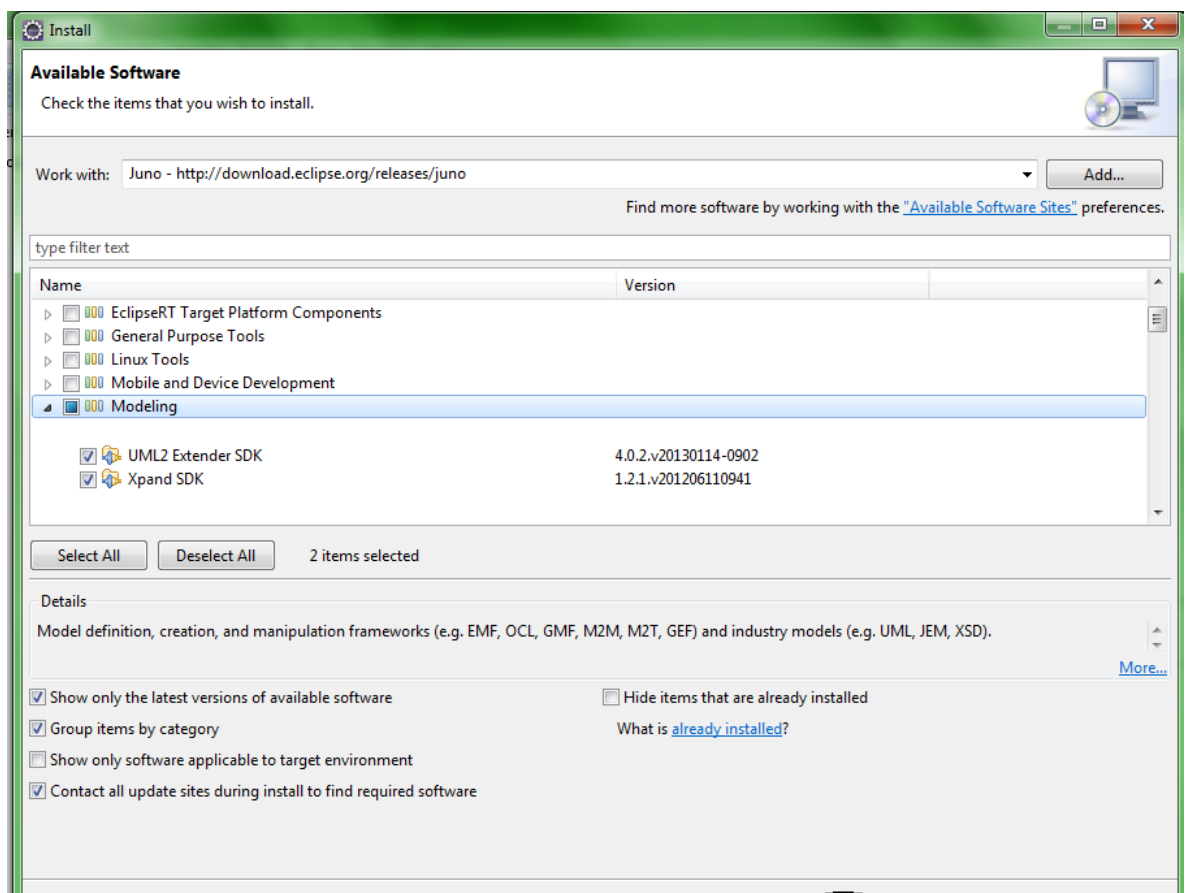
Nun installieren wir die benötigten Plugins:
Als erstes gehen Sie auf die Downloadseite von Juno wie in Abb. 2 gezeigt.

Abb. 2:



1. Plugin XPAND SDK und das UML2 Extender SDK siehe Abb. 3.

Abb. 3:

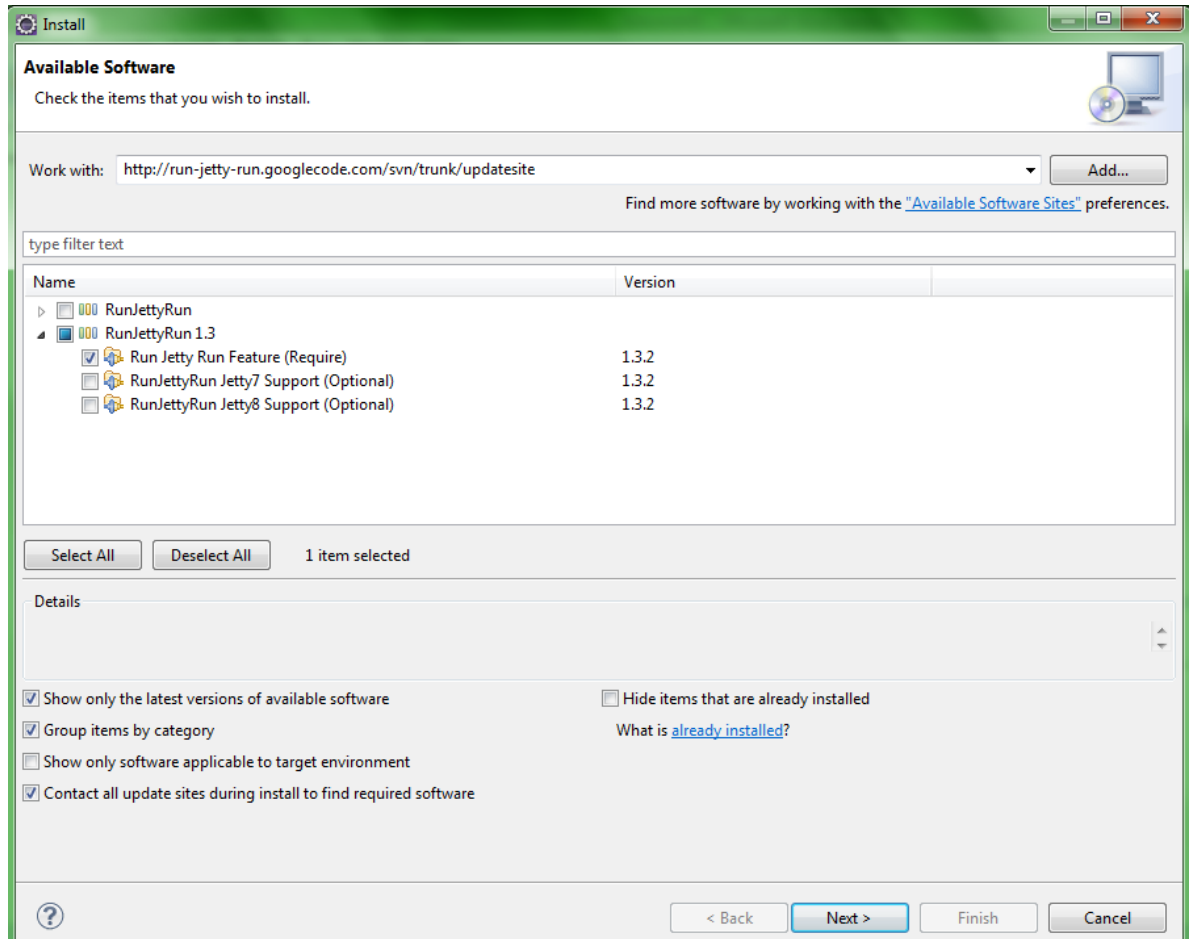


Sie drücken auf Next> und folgen der Installation.

2. Plugin:

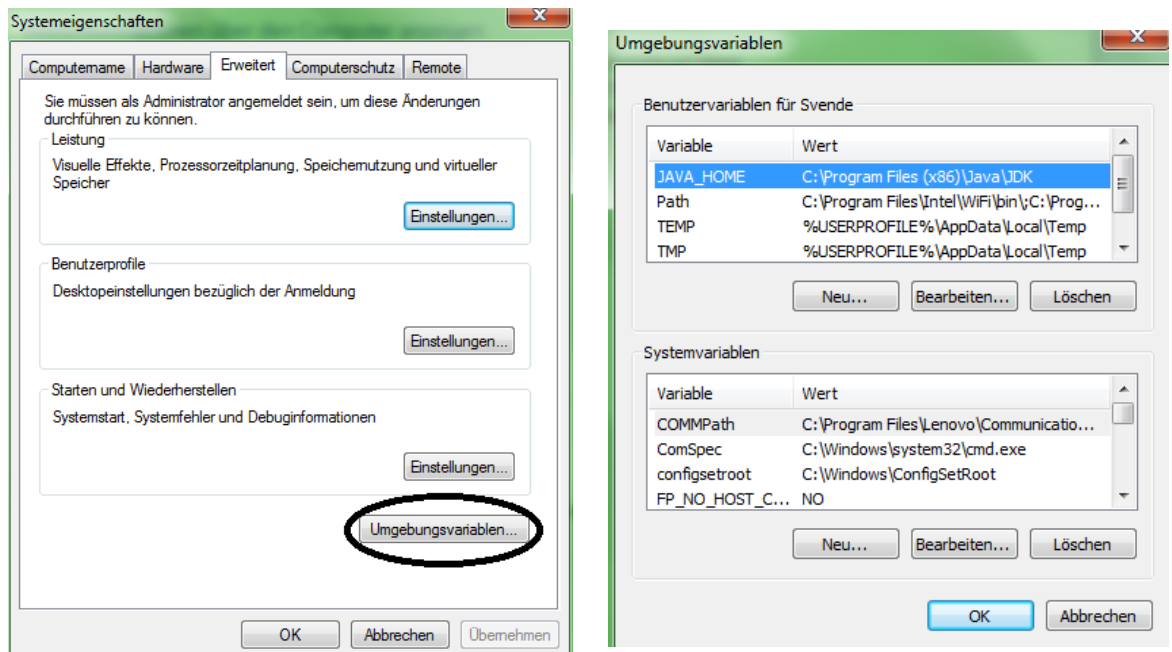
Nun geben Sie die Seite <http://run-jetty-run.googlecode.com/svn/trunk/updatesite> ein und installieren 'Run Jetty Run Feature (Require)' aus RunJettyRun 1.3 siehe Abb. 4.

Abb. 4:



Nun installieren wir Maven von <http://maven.apache.org/download.html> Und setzen die Umgebungsvariable für Maven JAVA_HOME Mit dem Pfad Ihrer JAVA-Version, wenn Sie in die Konsole gehen und 'mvn --version' eingibt, können Sie testen ob die Umgebungsvariable richtig gesetzt ist Siehe Abb. 5.

Abb. 5:

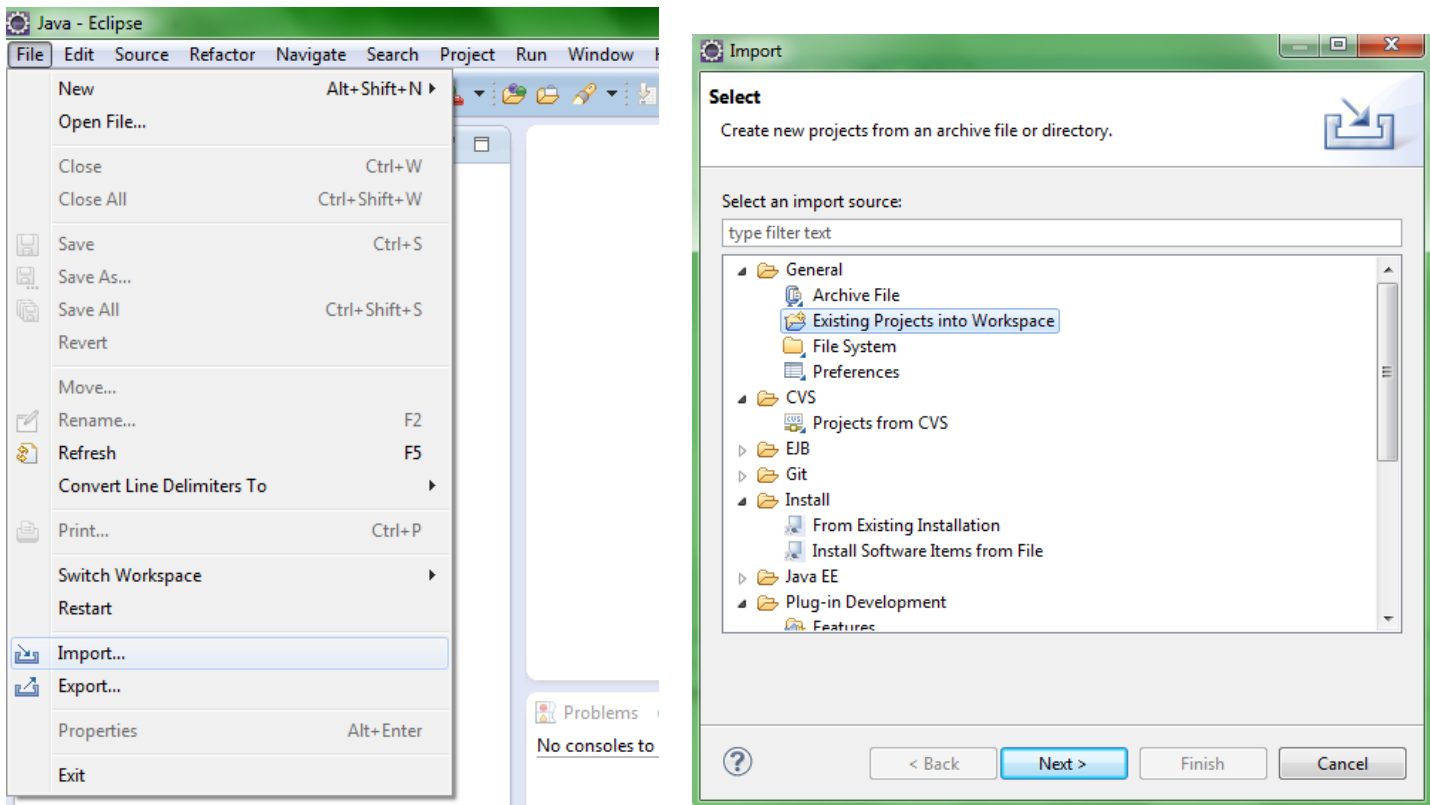


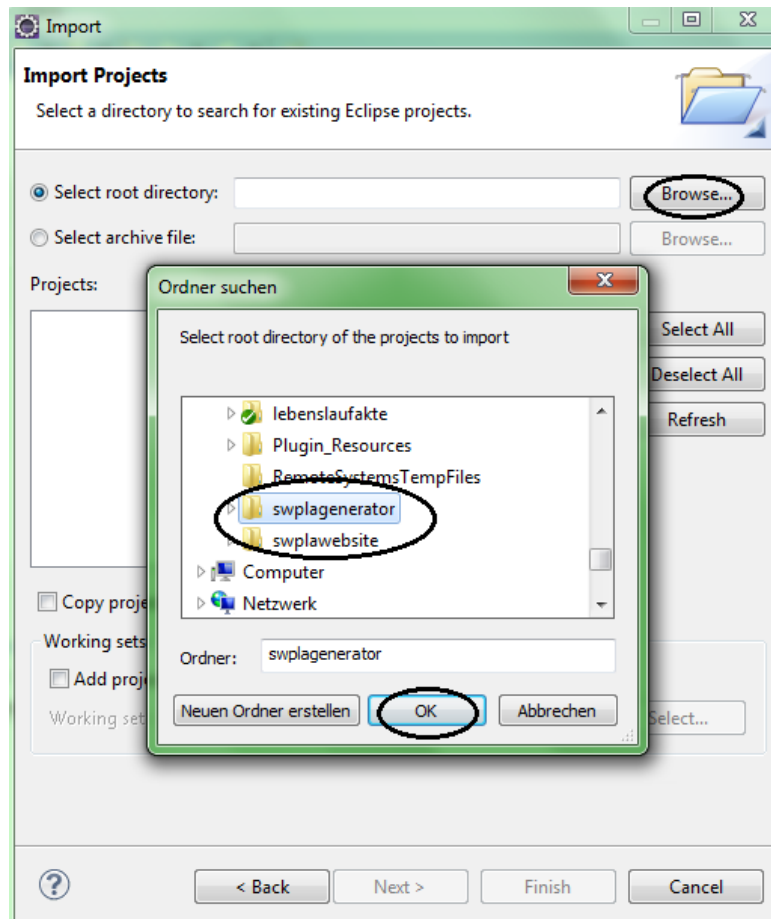
Jetzt kopieren Sie das Programm in ihren Eclipse Workspace. (Üblicherweise befindet er sich in Ihrem Benutzerordner)

Jetzt laden Sie die Dependencies mit Maven:

Rufen Sie die Konsole auf und gehen Sie in Ihrem workspace in den Ordner swpla.website, und rufen dann „mvn eclipse:eclipse“ auf.

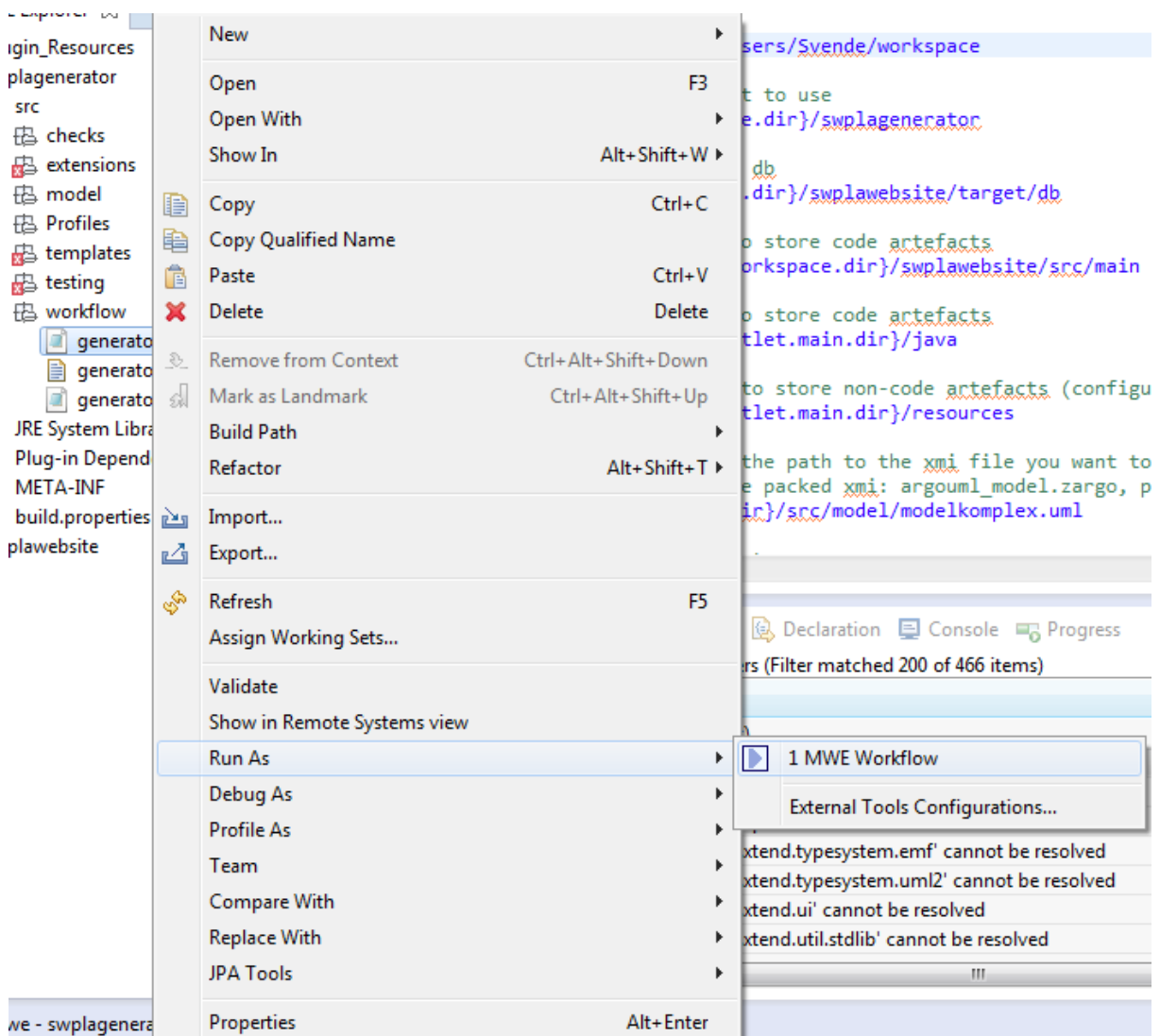
Jetzt importieren wir das Projekt in Eclipse:



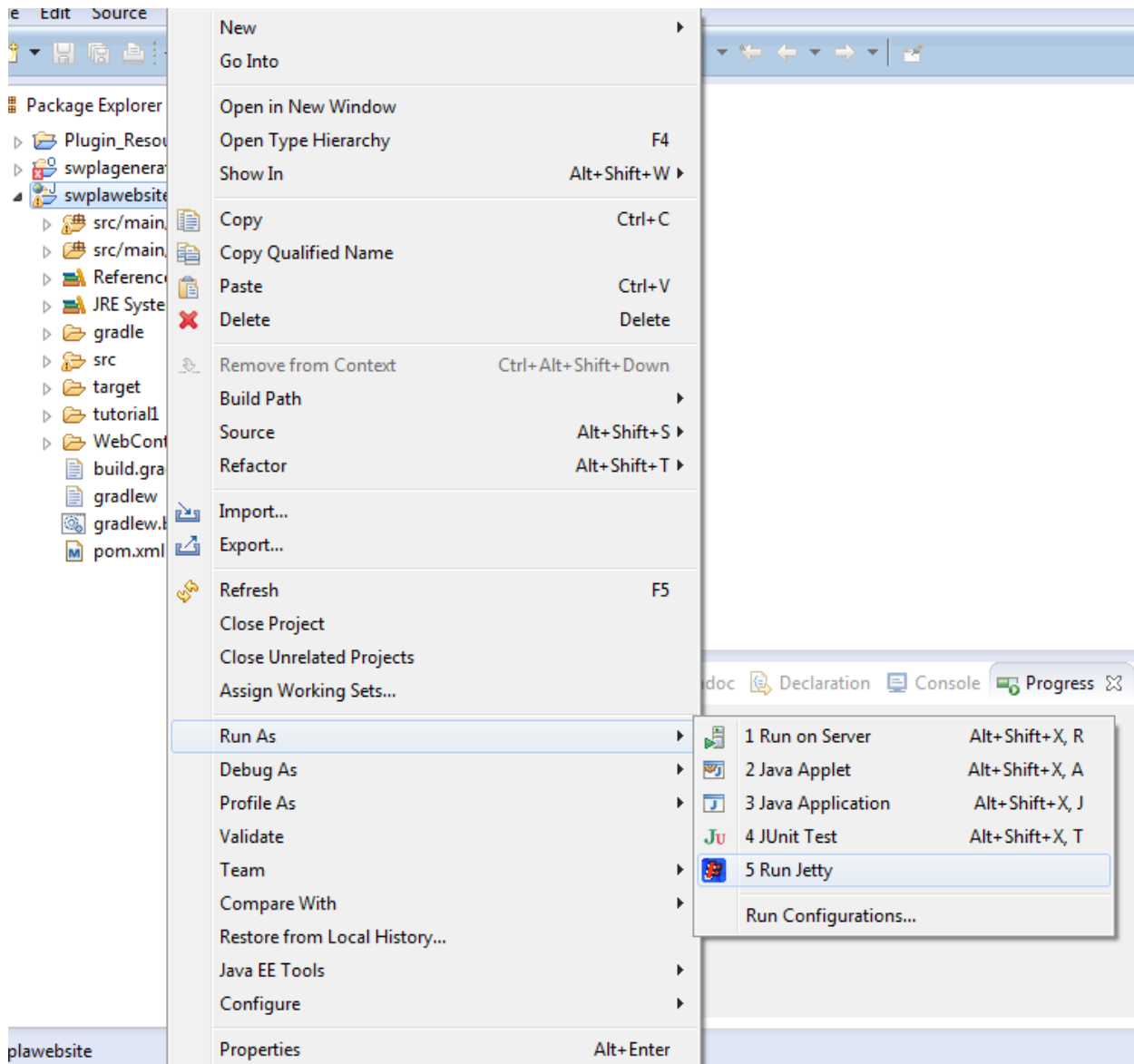


(Falls swpla.website nun tutorial1 heißt nennen Sie es wieder swpla.website (es ist wichtig das es genau so heißt!))

Jetzt können Sie die generatorPrim.mwe als workflow ausführen (rechtsklick -> run as 'mwe workflow')



Nun führen Sie die swpla.website per rechtsklick mit Jetty aus (rechtsklick -> run as 'Run Jetty')



Nun können Sie sich die generierten Seiten auf <http://localhost:8080/swplawebsite/> ansehen und verwenden.

5.3. Verwendung der Website

5.3.1 Grundlegendes

Prinzipiell alles auf Instanzen bezogen. Warum? Instanzen als konkrete Konstrukte, Beziehungen machen nur Sinn im Zusammenhang zwischen Instanzen.

5.3.2 Hintergrundwissen

Package-Struktur:

Die PackageStruktur spiegelt auch die Struktur der Website wieder. Tapestry mappt auch die Packages auf bestimmte URL's, so wird der Create der Klasse 1 auf die URL /Klasse1/Create/ gemappt. Dabei ist zu beachten, dass Tapestry hier auch automatisch die URL verkürzt bei Prä- oder Suffixen, so heißt das Create Template CreateKlasse1.tml, Tapestry erkennt Klasse1 dann als Suffix und nimmt es nicht nochmal in die URL auf.

Struktur der Packages

Die Packages sind in folgende Punkte aufgeteilt:

1. src/main/java, dieses Package enthält die Java-Quellcode Dateien, welche Attribute und Methoden einbinden
2. src/main/resources enthält weitere Ressourcen wie Templates(.tml) und properties. Diese Dateien bestimmen das Aussehen der Seiten.
3. src/main/webapp enthält spezielle Webanwendungs-Dateien. Beispiel dafür ist die Layout.css, die das Design der Seite beschreibt und unabhängig von Tapestry ist.

innerhalb la.website befinden sich die wichtigsten Dateien:

- *.pages enthält die Tapestry Seiten
- *.components enthält selbst erstellte Tapestry Komponenten
- *.entities enthält die Hibernate-Entities. Dies sind Java-Klassen, die benutzt und persistiert werden um die Daten zu verwalten.
- *.data enthält Klassen, die indirekt benutzt werden, wie Enumerations oder DAO's
- *.services enthält ServiceModule. Dies sind verschiedene Module zum Einbinden von Bibliotheken, zur Konfiguration der Webanwendung etc.

Zur vollständigen Funktionalität werden in Tapestry immer 2 Dateien benötigt. Eine Template Datei und eine namensgleiche Java-Quellcode Datei.

5.3.3 Beschreibung der Webseite

Die Startseite der Webseite ist eine Übersicht der Klassen inklusive der angelegten Instanzen.

Zum Anlegen der Daten wird eine Tab-Komponente verwendet. Außerdem werden hiermit Assoziationen zugeordnet.

Der Aufbau:

1 Primitive Attribute

The screenshot shows a navigation bar with buttons for 'STARTSEITE', 'INSTANZEN', 'BEZIEHUNGEN', 'DOKUMENTE', 'VERLAUF', and 'KONTAKT'. Below the navigation bar is a green button labeled 'CREATE NEW ANIMAL'. The main content area features a tabbed interface with four tabs: 'ATTRIBUTE', 'OBLIGATORISCHE ASSOZIATIONEN', 'OPTIONALE ASSOZIATIONEN', and 'DOKUMENTE'. The 'ATTRIBUTE' tab is active, displaying a form with two input fields: 'NAME:' and 'ALTER: *'. The 'ALTER' field has a red asterisk and a value of '0'. Below the form is a red button labeled 'ERSTELLE NEUE INSTANZ DER KLASSE ANIMAL'.

In der Mitte des Bildes können Sie die Tabkomponente sehen. Hier werden die Daten der Klasse eingetragen.

2 Obligatorische Assoziationen

The screenshot shows the same navigation bar and 'CREATE NEW ANIMAL' button as in the previous image. The main content area features a tabbed interface with four tabs: 'ATTRIBUTE', 'OBLIGATORISCHE ASSOZIATIONEN', 'OPTIONALE ASSOZIATIONEN', and 'DOKUMENTE'. The 'OBLIGATORISCHE ASSOZIATIONEN' tab is active, displaying a dropdown menu with the selected option 'HATHAUSTIER - PERSON'. Below the dropdown are two input fields: 'NAME:' and 'VORNAME:'. Below the form is a red button labeled 'ERSTELLE NEUE INSTANZ DER KLASSE ANIMAL'.

Hier werden die Daten der Klassen die mit der angelegten Klasse in Assoziation, wenn diese eine Multiplizität größer 0 oder eine Komposition/Aggregation hat.

Man kann die Eingabetabelle aus und einklappen:

STARTSEITE INSTANZEN BEZIEHUNGEN DOKUMENTE VERLAUF KONTAKT

CREATE NEW ANIMAL

ATTRIBUTE OBLIGATORISCHE ASSOZIATIONEN **OPTIONALE ASSOZIATIONEN** DOKUMENTE

HATHAUSTIER - PERSON

ERSTELLE NEUE INSTANZ DER KLASSE ANIMAL

3 Optionale Assoziationen

STARTSEITE INSTANZEN BEZIEHUNGEN DOKUMENTE VERLAUF KONTAKT

CREATE NEW ANIMAL

ATTRIBUTE OBLIGATORISCHE ASSOZIATIONEN **OPTIONALE ASSOZIATIONEN** DOKUMENTE

ERSTELLE NEUE INSTANZ DER KLASSE ANIMAL

Hier werden die Daten der übrigen Klassen, die mit Assoziation mit unserer angelegten Klasse verbunden sind, angelegt.

Fehlerabfang:

Sie müssen alle Mussfelder ausfüllen sonst erscheint ein solches Fenster:

STARTSEITE INSTANZEN BEZIEHUNGEN DOKUMENTE VERLAUF KONTAKT

CREATE NEW ANIMAL

FORMERROR

You must correct the following errors before continuing.

- You must provide a value for Alter.
- You must provide a value for File.


ATTRIBUTE OBLIGATORISCHE ASSOZIATIONEN **OPTIONALE ASSOZIATIONEN** DOKUMENTE

NAME:

ALTER: **△ You must provide a value for Alter.**

ERSTELLE NEUE INSTANZ DER KLASSE ANIMAL

6. FAQ

Fehler	Behebung
Es bricht ab, nachdem ich den Generator gestartet habe	Bitte vergewissern Sie sich das keine Fehler im UML sind oder Klassen ohne Attribute oder mit leeren Packages
Ich bekomme einen JAVA Versionsfehler Java compiler level does not match the version of the installed Java project facet	Kontrollierne Sie in der pom die JAVA Version es muss Version 1.7 sein <pre data-bbox="804 568 1398 999"> <build> <plugins> <plugin> <artifactId>maven-compiler- plugin</artifactId> <configuration> <source>1.6</source> <target>1.6</target> </configuration> </plugin> </plugins> </build> </pre>
Jetty startet bei mir nicht richtig es kommt der Fehler :INFO::Started SelectChannelConnector@0.0.0.0:8080	Bitte prüfen Sie ob ihre Umgebungsvariablen richtig gesetzt wurden
Bei Eintragung eines Attributwertes erscheint diese Meldung: 	Bitte füllen Sie alle Pflichtfelder aus

7. Glossar

Attribute	Eigenschaften eines Objektes, in Bezug auf Instanzen wie bspw. "Anlage" sind dies bspw. "Leistung" oder "Größe" und in Bezug auf Relationen wie "besitzt" sind dies insbesondere beteiligte Klassen aber auch die zugehörige Instanz einer Relationsklasse.
Bearbeiten	Bezeichnet das Löschen oder Editieren eines Objektes.
Beziehungen	konkrete Realisierung einer Relation.
Codegenerator	Ein Programm das bei Eingabe eines UML-Diagramms, einen Quelltext ausgibt.
Datenbank	Menge aller gespeicherten Daten.
Editieren	Bezeichnet das Bearbeiten von Attributen eines Objektes, jedoch nicht das Löschen eines Objektes.
Funktionsstüchtig	Der Code heißt funktionsstüchtig, wenn er ausgeführt werden kann und die beabsichtigten Funktionen implementiert sind.
Instanzen	konkrete Realisierungen einer Klasse.
Klasse	Menge von Attributen.
Löschen	Zerstört ein Objekt.
Objekte	Menge an Instanzen und Beziehungen.
Persistenzschicht	Die Persistenzschicht ist zuständig für die persistente Ablage von Objekten.
Relation	stellt eine Beziehung zwischen zwei Klassen dar.
Apache Tapestry	Webframework für die Programmiersprache Java
Xpand	Xpand ist eine statisch typisierte Templatesprache mit speziellen, für die Codegenerierung wichtigen Features.
TOPCASED	(Toolkit in Open Source for Critical Applications & Systems Development; deutsche Quelloffene Werkzeugsammlung für die Entwicklung kritischer Anwendungen und Systeme) ist ein CAE-Werkzeug.
Hibernate	englisch für Winterschlaf halten) ist ein Open-Source-Persistenz- und ORM-Framework für Java.
Eclipse (IDE)	Software-Plattform und Integrierte Entwicklungsumgebung
UML- Unified Modeling Language	(Vereinheitlichte Modellierungssprache), kurz UML, ist eine graphische Modellierungssprache zur Spezifikation, Konstruktion und Dokumentation von Software-Teilen und anderen Systemen

8. Entwicklerteam und Support

Name, Vorname	Zu besetzende Rolle
Scheeren, Nicole	Teamleiter, Qualitätsmanagement
Misch, Danny	Technischer Assistent
Landwehrkamp, Svende	Recherche, Dokumentation
Mathei, Bastian	Recherche, Modellierung
Feirrer, Raphael	Tests
Draeger, Christian	Implementierung Chefentwickler des Generatorteam
Hainke, Kai	Recherche, Implementierung Chefentwickler des Hompageteams

Für weitere Informationen zum Programm oder zum Projektverlauf können sie auch auf unsere Homepage gehen: <http://pcai042.informatik.uni-leipzig.de/~swp13-la/>

Email des technischen Assistent: dannymisch@gmx.de