



Orientieren Sie sich zur Implementierung der Features an den Screencasts zur Übung oder wenden Sie sich bei Fragen an die Betreuer.

1. Wizard

In diesem Feature ist das Editorprojekt (Stand nach HA4) um einen Wizard zu erweitern:

- Erstellen Sie einen Wizard, mit dessen Hilfe ein neues Scenario-Diagramm erzeugt werden kann
 - Der Wizard soll zwei Seiten umfassen:
auf Seite 1 ist der Package-Name anzugeben,
auf Seite 2 ist das Projekt auszuwählen und der Diagrammname zu definieren.
 - Nach der Benutzereingabe sollen ein neues Graphiti-Diagramm und für die Speicherung drei separate Dateien angelegt werden. Die Diagrammdatei „,<diagrammname>.sd“ für die grafischen Diagramm-Anteile, „,<diagrammname>.sm“ für die Scenariomodell-Anteile, „,<diagrammname>.cm“ für die EMF Klassenmodell-Anteile
 - In der Diagrammdatei „,<diagrammname>.sd“ sollte nach dem Erzeugen ein leeres Diagramm vorhanden sein
 - Die Scenariomodell-Datei „,<diagrammname>.sm“ sollte ein leeres ScenarioModel enthalten.
 - Die Klassenmodell-Datei „,<diagrammname>.cm“ sollte über ein leeres EPackage mit dem festgelegten Packagenamen beinhalten.



2. Codegenerierung Klassenmodell

Erweitern Sie das Editorprojekt um die EMF-Codegenerierung aus der Klassen-Modell-Datei:

- Erstellen Sie einen Eclipse Command, welcher einen Dateibrowserfenster öffnet und auf auswählen der Klassenmodell-Datei „diagrammname.cm“ EMF-Code für die bisher benötigten Java-Klassen generiert:
 - Definieren Sie die notwendigen Extensions in der Datei „plugin.xml“. Diese sind dann im Laufzeit-Eclipse zugänglich da dort das Editorprojekt als Plugin geladen wird.
 - Implementieren Sie eine Handler-Klasse, welche einen ElementTreeSelectionDialog öffnet und auf das auswählen einer cm-Datei das enthaltene Klassenmodell in ein Ecore-GenModel lädt. Abschließend soll der Handler die Generierung anstoßen.
 - Modellieren Sie ein einfaches Scenariodiagramm, welches zwei ScenarioObjects, unterschiedlichen Typs und ohne Attribute, sowie einen ScenarioLink umfasst. Lassen Sie anschließend daraus Code generieren.



3. Codegenerierung Scenario Testcode

In diesem Feature soll das Diagrammprojekt um die Scenario-Modell-Codegenerierung erweitert werden:

- Vorbereitung:
 - Erstellen Sie im Runtime-Eclipse ein Plugin-Projekt in welchem Sie ein ScenarioDiagramm modellieren.
 - Importieren Sie ihr ScenarioModel-Projekt in den Runtime-Workspace (ohne es zu kopieren) und machen Sie es ihrem Plugin-Projekt in der Datei MANIFEST.MF unter Dependencies bekannt.
 - Installieren Sie die in der Übung gezeigten Eclipse-Plugins und machen Sie sie ebenfalls in der Datei MANIFEST.MF bekannt.
- Erstellen Sie einen MW2E Workflow (z.B. *codegenWorkflow.mwe*), der die Codegenerierung aus einer ebenfalls zu erstellenden Xpand Template-Datei (*codegenTemplate.xpt*) anstößt.
- Passen Sie die Template-Datei so an, dass sie einen JUnit-Test generiert. Zusätzlich wird eine Extension-Datei und eine Java Helper-Klasse notwendig sein um Imports und Assert-Anweisungen zu generieren.
- Im JUnit-Test soll eine Testmethode enthalten sein, welche die im Diagramm modellierte Objektstruktur aufbaut und mit Hilfe von Assert-Anweisungen überprüft ob sie korrekt initialisiert wurde.