

# Obeo Designer Training – Basics

**Goals:**

- Know how to design a simple modeling environment with Obeo Designer
- Learn the basic methodology for building an MDA environment

**Our advantages:**

- Training designed by the creators of Obeo Designer and eclipse foundation committers
- Many practical exercises (50%)

**Duration:** 4 days

**Audience:** Architects, Developers

**Prerequisites:** Basic knowledge of *java* development and use of the *Eclipse* IDE.

## 1 - Presentation of the MDA and Eclipse

- Overview of the Eclipse projects
- Introduction to the MDA concepts and the DSM approach
- Introduction to the Eclipse modeling technologies (EMF, GEF, GMF)

**Exercises:**

- Installation and configuration of an environment
- Creation of models using the tree editor

Duration: 2.5 h

## 2 - Presentation of Obeo Designer

- The Obeo Designer approach
- The Obeo Designer architecture
- The fundamental concepts of viewpoint-driven modeling
- Overview of the navigation languages: OCL and Acceleo
- The user features of Obeo Designer

**Exercises:**

- Using the model editor
- Guided tour of the views and wizards accessible to viewpoint users

Duration: 2.5 h

### 3 - Introduction to meta-modeling with EMF

- Basic concepts of Eclipse EMF
- The *Ecore* meta-model
- The *Ecore* data types
- Detailed explanation of the generation model (GenModel)
- Links between meta-models
- Use of profiles with Eclipse UML

#### *Exercises:*

- Development of a basic EMF modeler, with a dedicated meta-model
- Creation of models compliant with this meta-model
- First customizations of the modeler

Duration: 2 h

### 4 - Realization of an Acceleo generation module

- The Acceleo syntax and the template editor
- The instructions to request elements of the model
- The reflective editor
- Acceleo services
- Indentation strategy for templates and generated code
- Acceleo and UML Profiles and stereotypes
- Use of the requester

#### *Exercises:*

- Realization of an XHTML generator
- Navigation and advanced requesting with system services
- Realization of java classes generators
- Generation from UML or from a DSL

Duration: 5 h

### 5 - Model/code synchronization

- Presentation of Obeo Traceability
- Managing the model/code synchronization
- Managing the model/model synchronization
- Managing user code
- Generating traceability reports

#### *Exercises:*

- Using Obeo Traceability on the previous examples
- Using the different traceability views inside Eclipse

Duration: 2 h

## 6 - Creation of simple viewpoints

- General presentation of the viewpoint description file (.odesign)
- Creation of a graphical representation
- Creation of a tabular representation
- Creation of a pivot table representation
- Defining the “semantic”/“representation” mapping
- Defining a palette of actions
- Defining layers

### *Exercises:*

- Creation of “class-diagram-like” graphical viewpoints
- Discovery of available shapes
- Creation of tabular and pivot table viewpoints

Duration: 7 h

## 7 - Creation of dynamic viewpoints

- Creating conditional styles
- Using the “Model Request Interpreter” view
- Defining model manipulation tools
- Importing existing viewpoint configurations
- Defining filters
- Integrating java code with the representation definition

### *Exercises:*

- Creation of an advanced representation with wizards and context menus
- Creation of dynamic analysis layers
- Creation of a viewpoint with a recursive mapping

Duration: 7 h