

# ECLIPSE CADRU DE MODELARE

Tema Curs

INGINERIE SOFTWARE

Luca Diana – Sorina

Grupa 441A

Facultatea de Electronica Telecomunicatii si Tehnologia Informatiei

Universitatea Politehnica Bucuresti

## CUPRINS:

### ① MODELUL GENERATOR

### ② GENERAREA CODULUI

## **① Modelul Generator**

- **Introducere in genmodel**
  - Rolul modelului generator
  - Legaturile dintre ecore si genmodel
- **Configuratia generatorului**

### **Introducere**

#### **Rolul modelului generator**

In afara de modelul ecore, avem nevoie de asemenea si de modelul generator. Acesta furnizeaza informatiile specifice platformei, opus fata de modelul ecore care detine doar informatiile independente de platform.

Modelul generator este utilizat pentru a genera cod. Acest model ne permite sa configuram codul generat cum dorim, de exemplu:

- ce pachet sa utilizam;
- cum sa afisam structura modelului

#### **Ce trebuie sa configuram?**

Genmodel-ul contine un set de optiuni pentru generarea unui editor plug-in pentru Eclipse bazat pe modelul ecore.

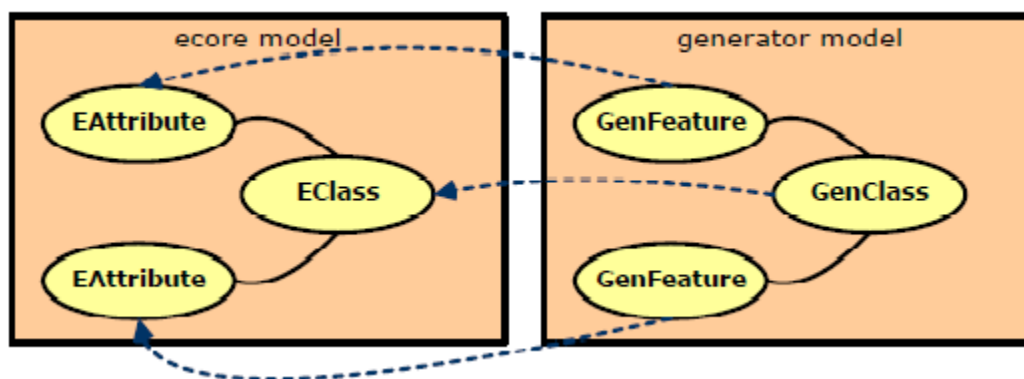
Organizarea codului si meta data

- Copy right text
- Pachetul de informatii
- etc

Optiuni de prezentare:

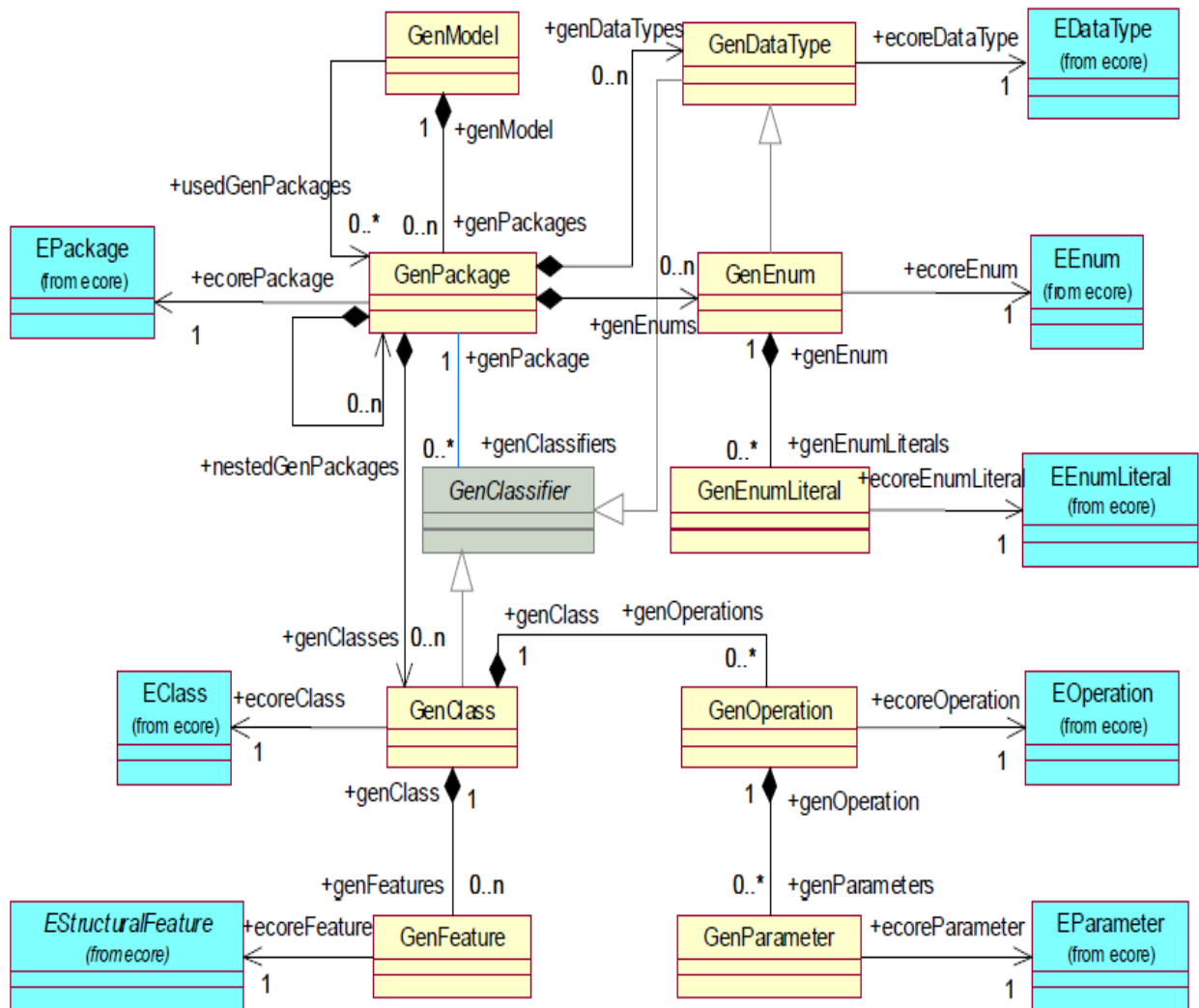
- Arata ca un copac?
- Care atribut reprezinta eticheta
- etc

**Relatiile dintre ecore si modelul generator:**



Modelul generator detine cate un element pentru fiecare din elementele modelului ecore. Obiectele in genmodel contin:

- O referinta la un element ecore
- Optiuni pentru generare de cod



## **Configuratia modelului generator**

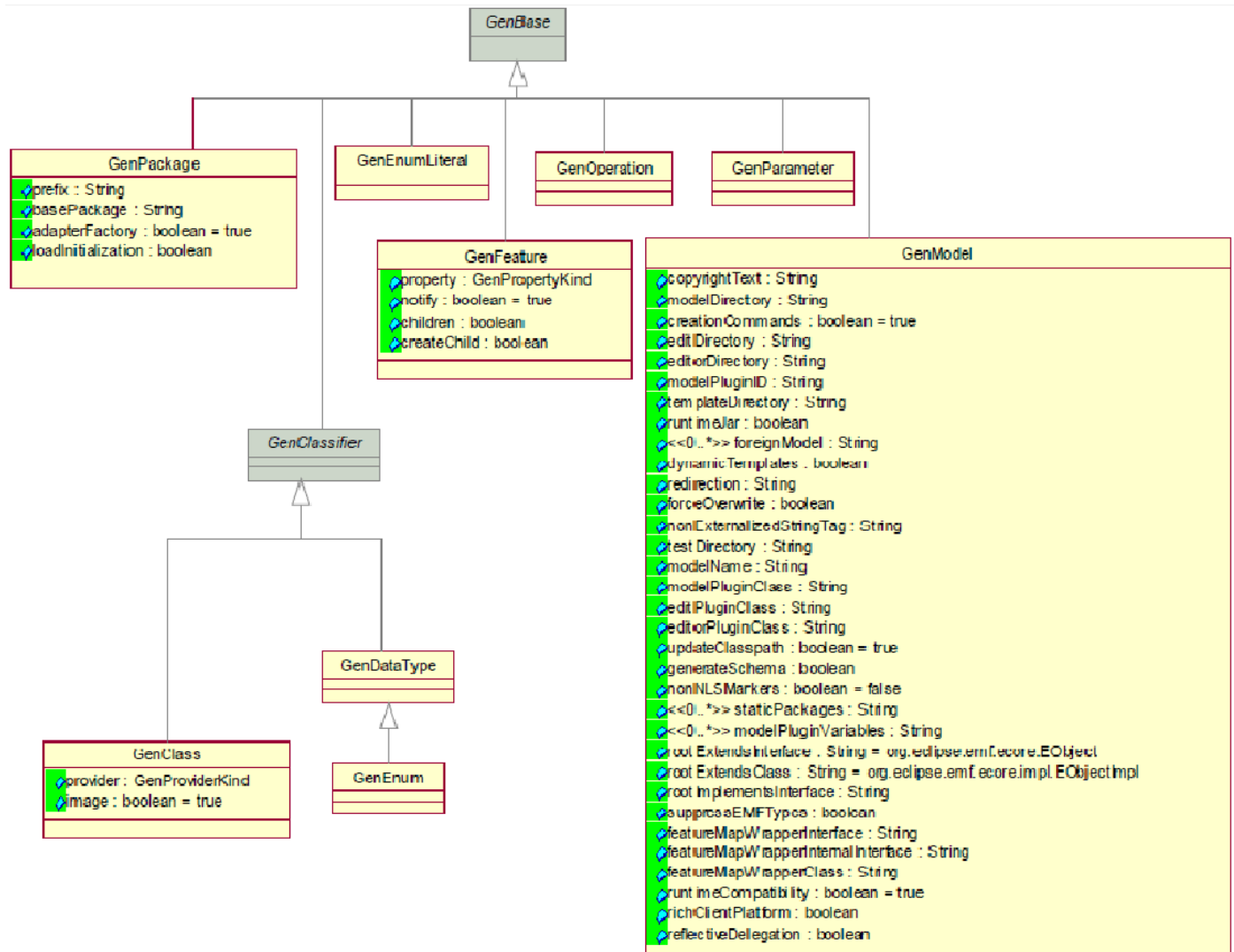
- Ce poate fi configurat?
- Metamodel-ul pentru genmodel
- Efectele si optiunile de configurare (referinta)

Optiuni generale de configurare (si subcategorii):

- All
- Edit
- Editor
- Model
- Model Class Defaults (implicite)
- Model Feature Defaults
- Templates and Merge

Elementele modelului:

- GenPackage
- GenClass
- GenFeature



## Cum se creeaza un model generator?

Folosirea wizard'urilor incluse in EMF pentru a crea un nou model EMF

Definirea bazei PIM:

- Model ecore?
- Definitia schemei XML?
- Interfete JAVA?
- Modelul Rational Rose?

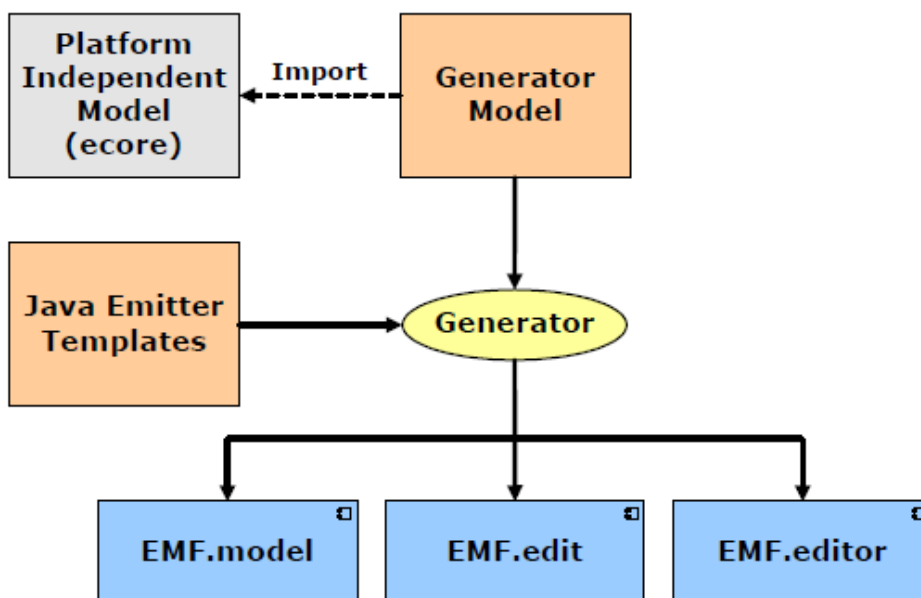
Definirea pachetelor pentru includere

Generarea de model

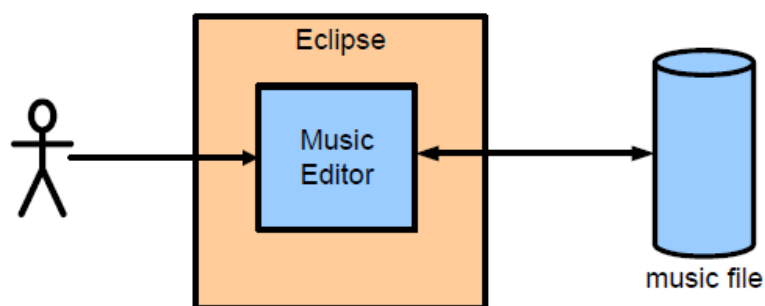
## ② Generarea codului

- Ce genereaza?
- Ce nu genereaza?
- “Overriding” codului generat

Privire de ansamblu asupra generarii de cod:



Imagine despre implementarea generata:

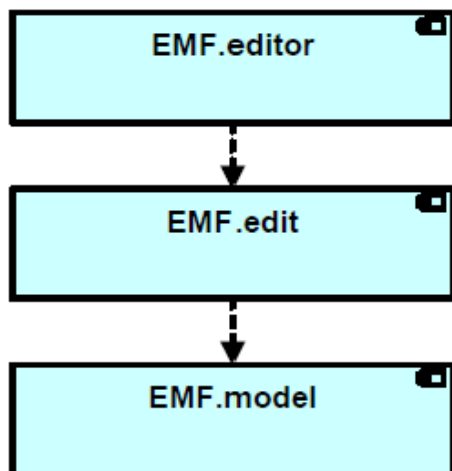




Generatorul EMF creeaza un editor pentru continut (volum) pe baza PIM

- Plug-in pentru Eclipse
- Serializare implicita XML

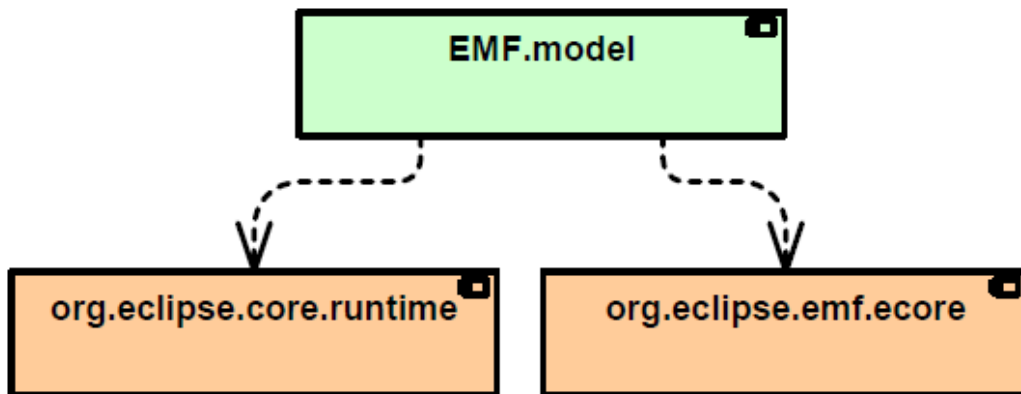
### Plug-in-uri create de EMF:



- **EMF.editor** : contine prezentarea codului
- **EMF.edit** : contine prezentarea adaptorilor independenti
- **EMF.model** : detine modelul bazat pe codul PIM

EMF poate crea trei plug-in-uri diferite.

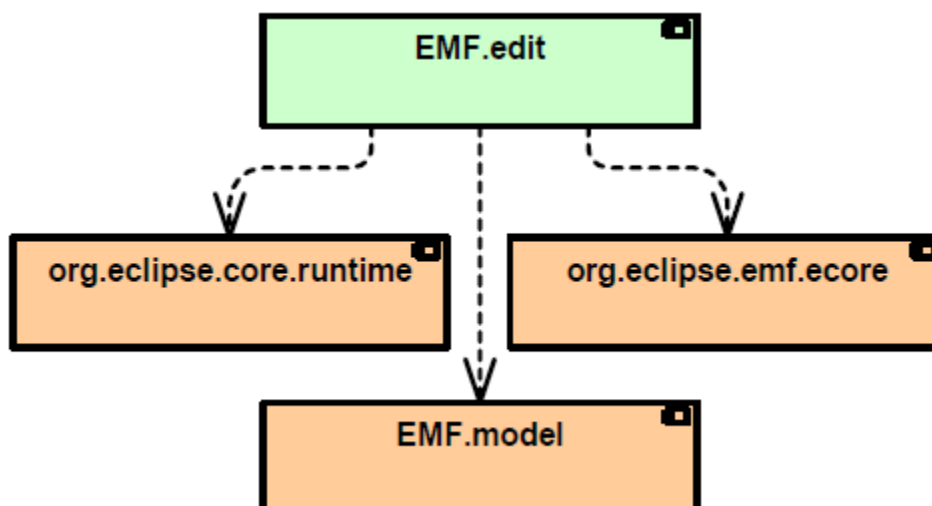
### EMF.model –ul:



Codul acestui model prevede o implementare complete a PIM –ului  
Persistenta mecanismului extrem de eficienta in top-ul fisierelor XML  
Continuitate 100% de la model la cod:

- Este previzibil tot codul generat
- De obicei, codul necesita modificari mici sau deloc

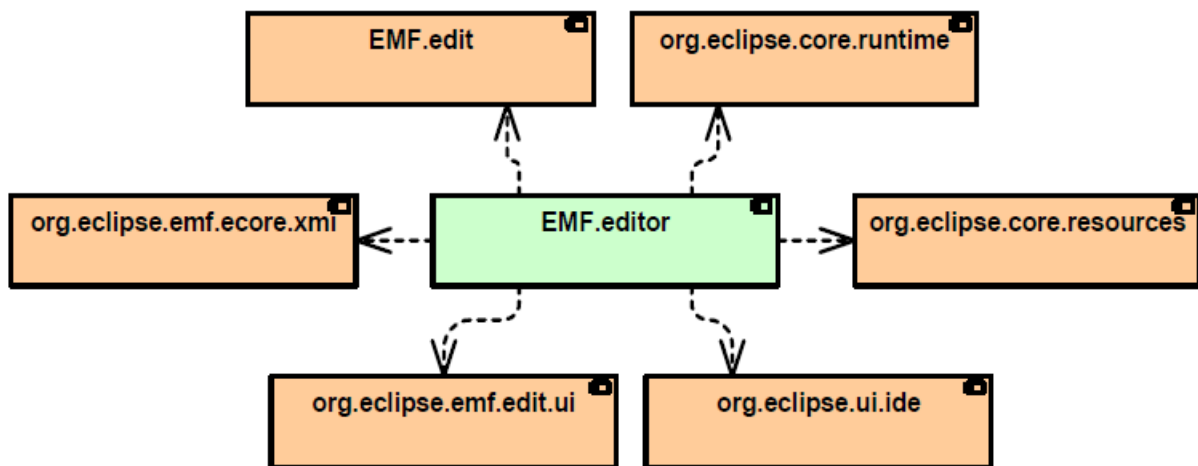
### EMF.edit –ul:



EMF.edit –ul actioneaza ca o prezentare, strat autonom adaptand obiecte model:

- Eticheta furnizori
- Trei modele
- Comenzi

### EMF.editor –ul:



EMF.editor –ul prevede codul SWT/JFace, acesta interactionand direct cu user-ul.

### Cum se poate schimba codul generat?

- Toate codurile generate detin comentariul java-doc : “@generated”

```

/**
 * This returns the image for the artist type.
 * @generated
 */
public Object getImage(Object object) {
    return getResourceLocator().getImage("full/obj16/Artist");
}
  
```

- Pentru a prelua codul din generator, se va schimba tag –ul: @generator

```
/**
 * This returns the image for the artist type.
 * @generated NOT
 */
public Object getImage(Object object) {
    return complexCalculationOfImage( object );
}
```

Doar codul cu tag –ul @generated va fi inlocuit atunci cand se efectueaza ulterior generare de cod.

## Java Emitter Template (JET)

- 1 Ce reprezinta JET?
- 2 Cum functioneaza JET?
- 3 JET outside EMF
- 4 JET in EMF
- 5 Manipularea de template-uri cu JET

1

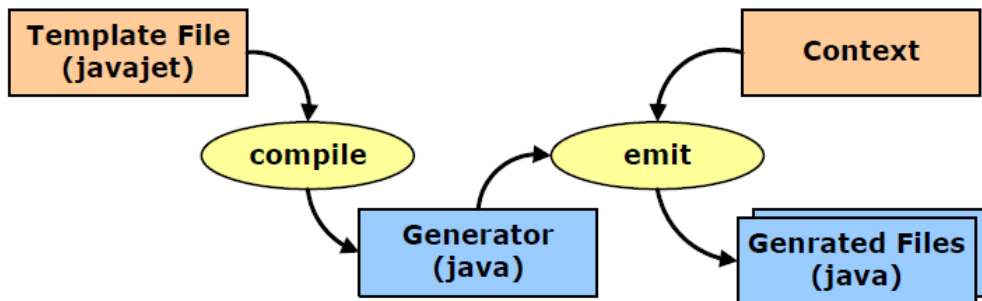
JET este folosit in EMF pentru generarea si fuziunea codului sursa java

JET este o parte din lansarea EMF

Termini cheie in JET:

- Fisierile sablon: - contin scripturi si structuri care definesc cartografierea la codul de modele EMF
- Generatoare : - template –urile genereaza cod; siruri de caractere.

2



JET, a fost initial construit din codul sursa al compilatorului Tomcat JSP  
JET, lucreaza asemanator cu JSP, astfel:

- Template –urile sunt la JET asa cum JSP –urile sunt la J2EE
- Generatorii sunt la JET ce Servletii sunt la J2EE
- Generarea codului sursa este la JET ce este iesirea HTML la J2EE

### Sintaxa JET:

- JET foloseste sintaxa JSP
  - <% ... %>
- Difera contextual explicit
  - JSP - implica pagina, cerere, sesiune
  - JET – implica obiecte: - stringBuffer (de tipul StringBuffer) // si – argument (de tipul Object)
- Daca stii JSP atunci cu rapiditate vei fi efficient in JET

### EXEMPLU: template javajet:

```

<%@ jet package="com.idata.hello" class="HelloWorldTemplate" %>

class Test {
    public void main(String args[]) {
        System.out.println("Hello world");
    }
}
  
```

Acesta a fost un template java foarte usor.

**Bibliografie:**

- **Mastering Eclipse Modeling Framework ; Vladimir Bacvansky, Petter Graff [EclipseCon2005\_Tutorial28,pdf]**
- ...\\org.eclipse.emf.codegen.ecore\_2.0.1\\templates\\model
- ...\\org.eclipse.emf.codegen.ecore\_2.0.1\\templates\\edit
- ...\\org.eclipse.emf.codegen.ecore\_2.0.1\\templates\\editor