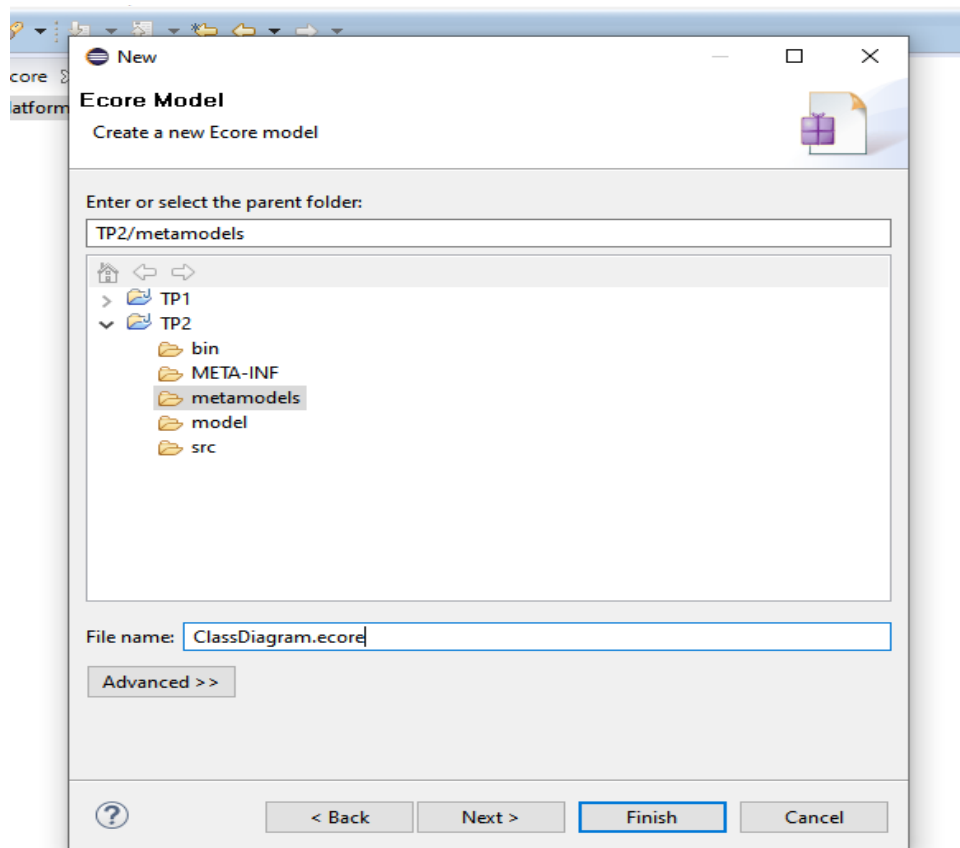
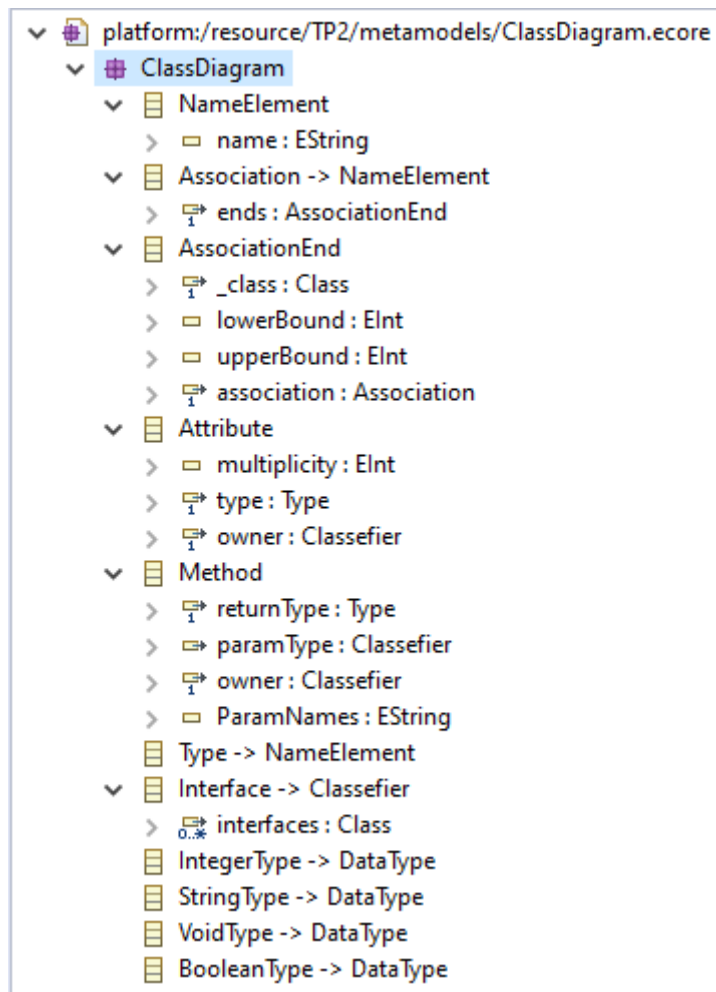
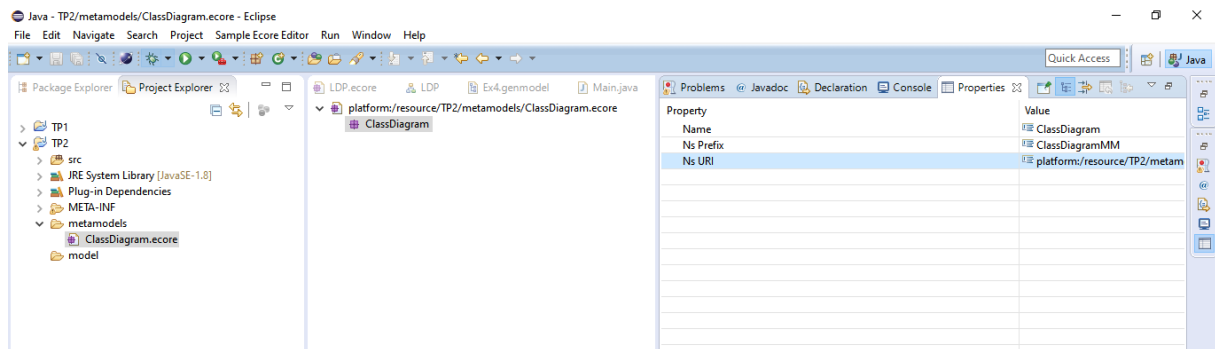
	<p align="center"><b>Institut Supérieur d'Informatique et du Multimédia de Sfax</b></p> <p align="center"><b>Ingénierie des modèles</b></p>	<p><b>Auditoire :</b> 2<sup>ème</sup> Ingénieur INFO <b>2024/2025</b> <b>Responsable de cours :</b> Mohamed Hadj Kacem <b>Enseignante TP :</b> ZAINEB SAKHRAOUI</p>
---	---	---

## TP 2 : Diagramme de classes UML

### Exercice 1 : Création d'un méta-modèle d'un diagramme de classes

**File -> New -> Other -> Eclipse Modeling Framework -> Empty EMF Project**





- ▼ **ModelBase**
  - > 0..\* allAssociations : Association
  - > 0..\* allInterfaces : Interface
  - > integerT : IntegerType
  - > StringT : StringType
  - > VoidT : VoidType
  - > boolT : BooleanType
  - > 0..\* allClasses : Class
- ▼ **Class -> Classefier**
  - > 0..\* EmplementBy : Interface
  - > 0..\* associations : AssociationEnd
- ▼ **Classefier -> Type**
  - > 0..\* methods : Method
  - > 0..\* attributes : Attribute
- DataType -> Type**

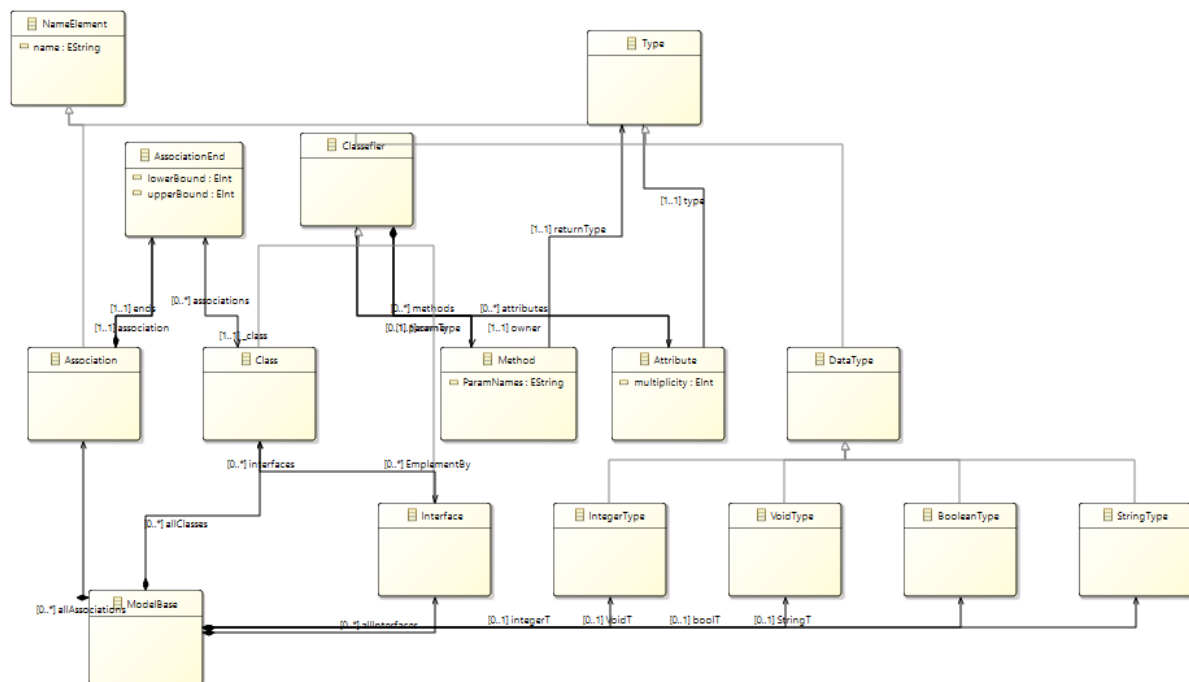


Figure 1: Un Méta-modèle d'un diagramme de classes.

## Exercice 2 : Création et édition d'un modèle (La banque)

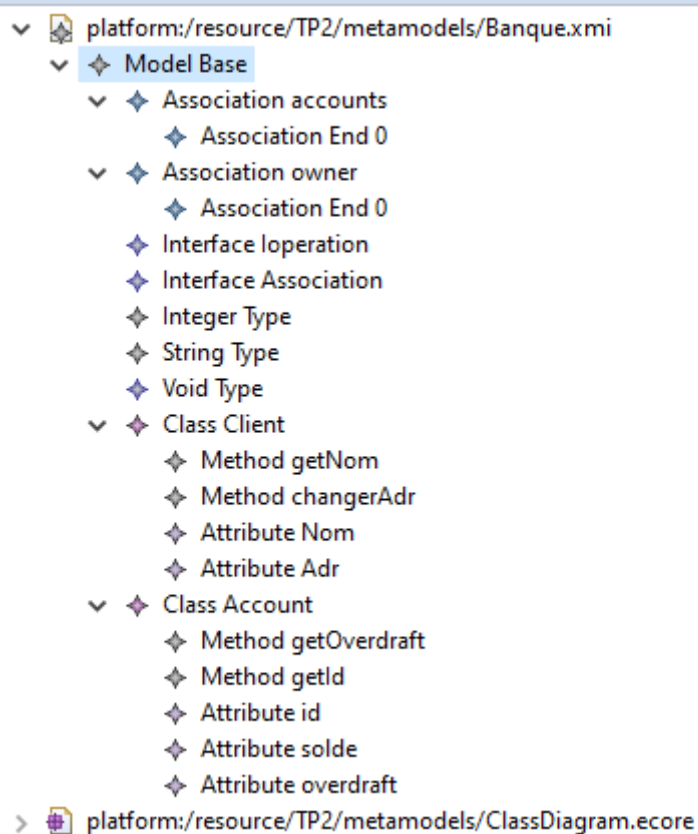
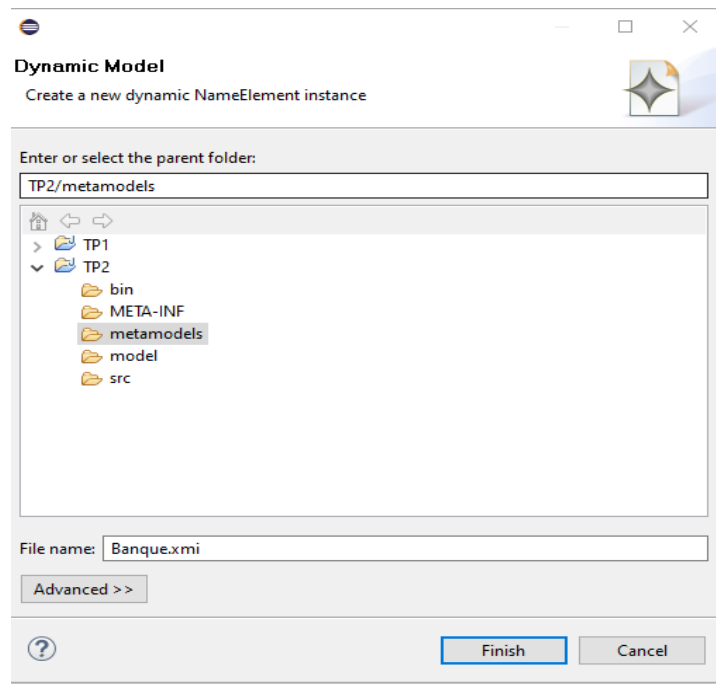


Figure 2 Création d'une instance Banque.xmi.

### Exercice 3 : Validation de modèle et de méta-modèle

1. Ajouter le contenu du code suivant qui présente un invariant vérifiant qu'un client possède un nom unique.

```
class client extends Class {  
    attribute Nom:String[1];  
    attribute Adr:String;  
    operation getAdr():String;  
    invariant NameUnique: client.allInstances()-> forall(c1,c2|c1<>c2 implies c1.Nom<>c2.Nom);  
}
```

Figure 3 class client

2. Ajouter un invariant pour garantir que toutes les associations doivent avoir un nom non vide.

```
class Association extends NameElement  
{  
    property ends : AssociationEnd[1] { composes };  
    invariant NotEmpty: self.name<> null and self.name.size() >0;  
}
```

Figure 4 associations non vide

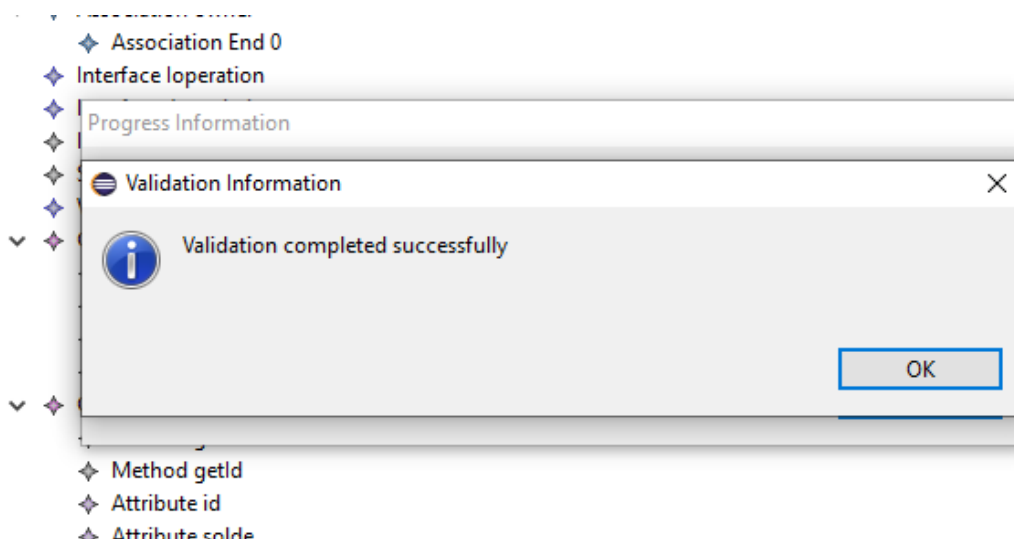


Figure 5 validation