



CONCEPTION AVANCÉE

COURS 1: UML 2 JAVA

lazaar@lirmm.fr

PLANNING ANALYZE DESIGN

**SOFTWARE
ENGINEERING**

VALIDATION
AND VERIFICATION

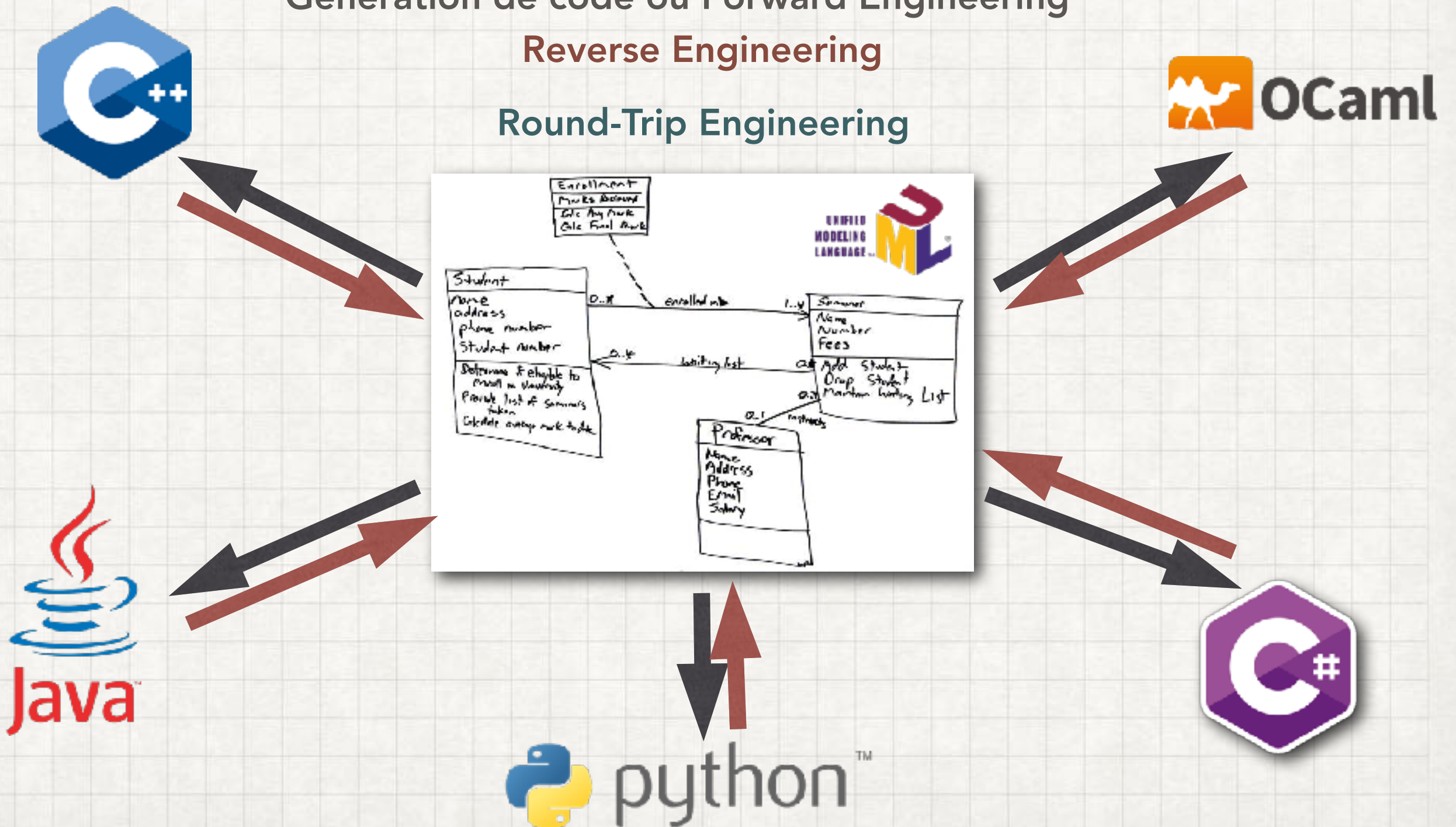


Principe du Pivot

Génération de code ou Forward Engineering

Reverse Engineering

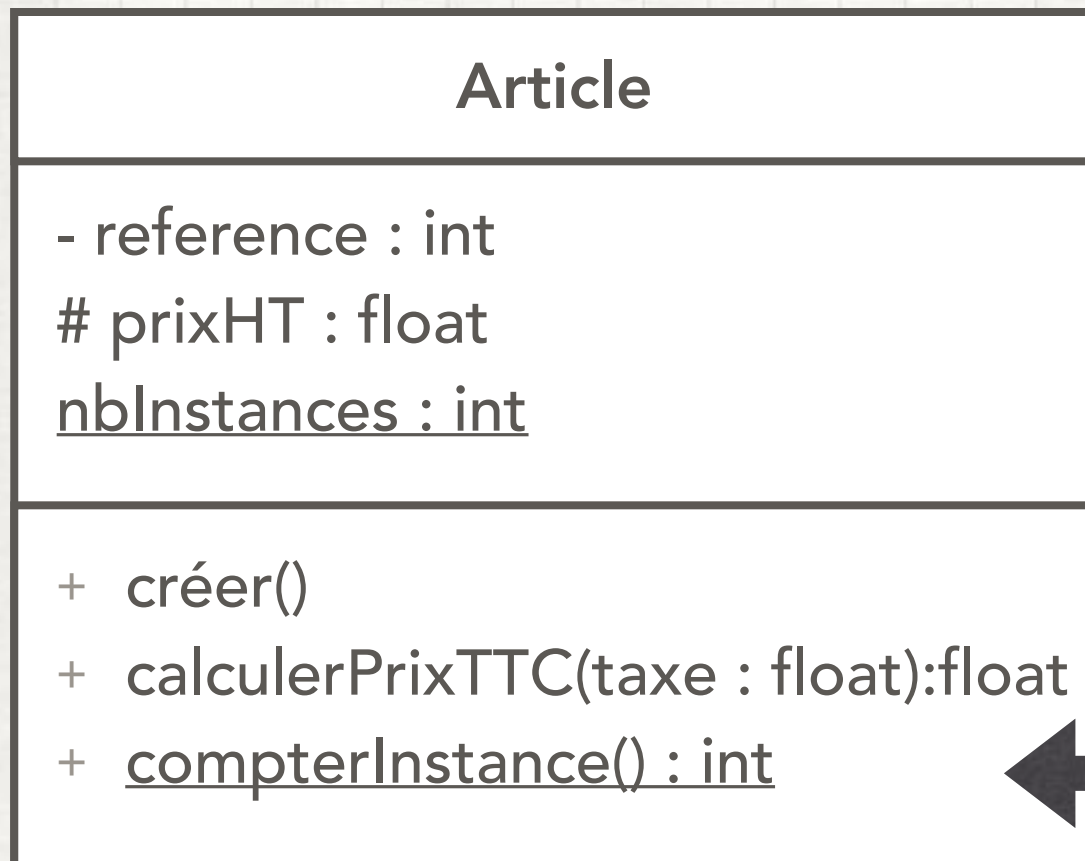
Round-Trip Engineering



Génération de Code

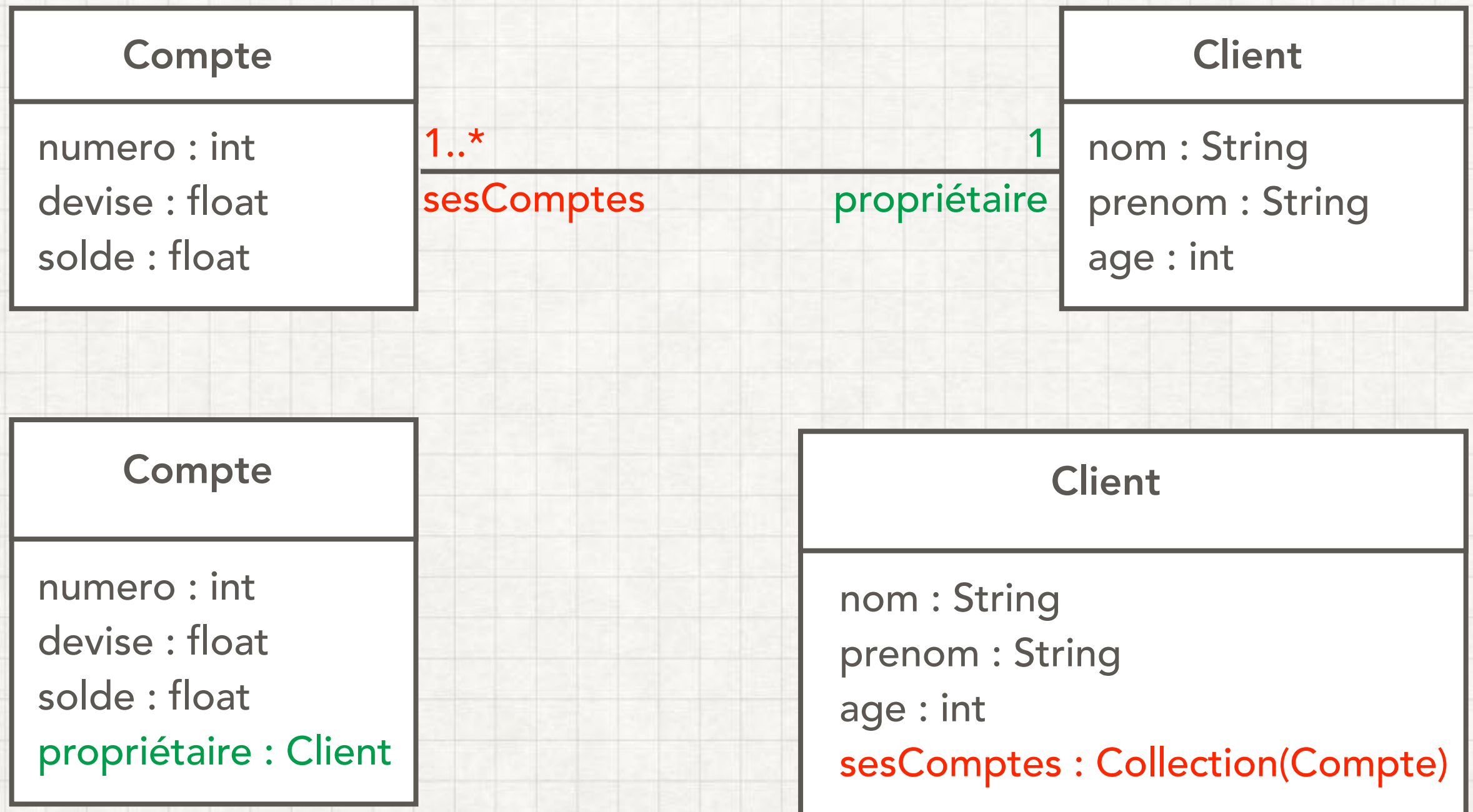
- Plus de productivité
- Moins d'erreurs
- Le métier de développeur de code passe à développeur de générateur de code
- Un ensemble de règles décrivant la correspondance
UML <-> Java

Classes 2 Java



```
public class Article {  
  
    private int reference;  
    protected float prixHT;  
    static int nbInstances;  
  
    public Article(){  
        // TODO: constructor  
    }  
  
    public float calculerPrixTTC(float taxe){  
  
        //TODO: calculerPrixTTC  
        return 0;  
    }  
  
    public static int compterInstance(){  
  
        //TODO: compterInstance  
        return 0;  
    }  
}
```


Associations 2 Java



Mais aussi !

- Si composition => attention à bien respecter la contrainte de non partage de composants
- Héritage => `extends`
- Implémentation d'une interface => `implements`
- ...

Encore plus de productivité

- Génération automatique des méthodes
- Diagramme d'activité par opération
- Diagramme de séquence
- Notes associées aux méthodes

```
ArrayList success = new ArrayList();  
for (Iterator it = personnes.iterator() ; it.hasNext();) {  
    Personne current = (Personne) it.next();  
    if (current.getNom().compareTo(nom)==0) success.add(current);  
}  
Personne[] res = new Personne[0];  
return (Personne[]) success.toArray(res);
```

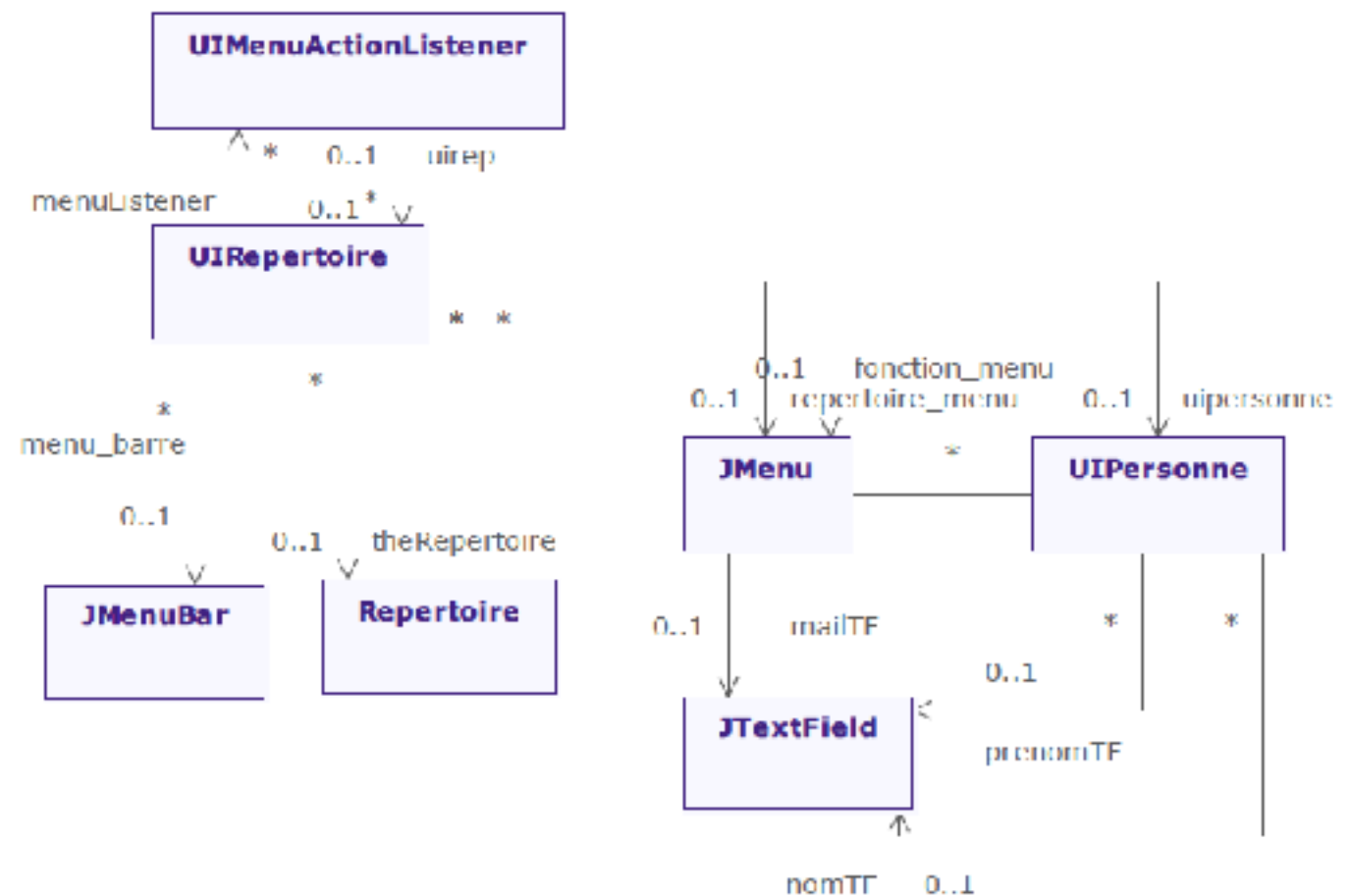
Repertoire
+ajouterPersonne(in p : Personne)
+supprimerPersonne(in p : Personne)
+rechercherPersonnesParNom(in nom : string);Personne[1..*]
OperationTest();Personne[1..*]

Reverse Engineering

- Retour vers un langage de plus haut niveau d'abstraction
- Extraire l'ensemble des vues sur l'application
- Réfléchir sur la structure d'un système existant
- Enrichir et répondre à de nouveaux besoins

Reverse Engineering

- Certains outils font le reverse de toute l'API Java => Pollue le diagramme de classe



Round-Trip Engineering

- Java 2 UML and UML 2 Java
- Un développement logiciel de haut niveau
- Permet de garder modèles et code en parfaite synchronisation
- Ex: RSA pour Rational Software Architect

$$f^{-1}(f(x)) = x$$