esprit
Se former autrement

Chapitre VII : Préparation de l'implémentation (partie 2: uml vers Java)

Module Langage de modélisation UML

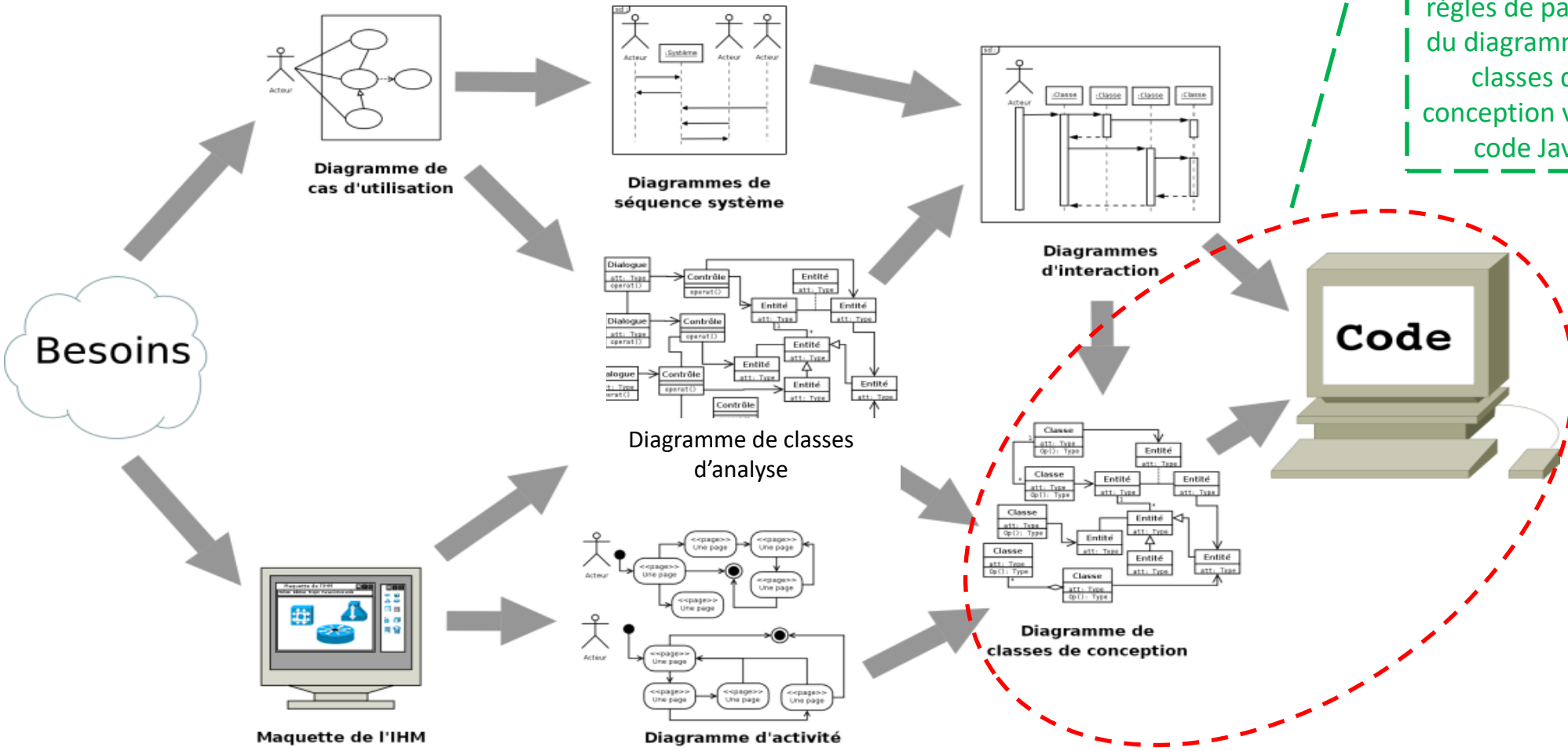
Année Universitaire
2015-2016

PLAN

- Introduction
- Règles de passage: Les Packages
- Règles de passage: Les classes
- Règles de passage: Attributs & méthodes
- Règles de passage: Associations Binaires
- Règles de passage: Associations particulières
- Exemple d’outils de génération de code

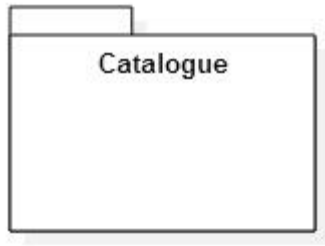
De UML vers un Langage orienté objet

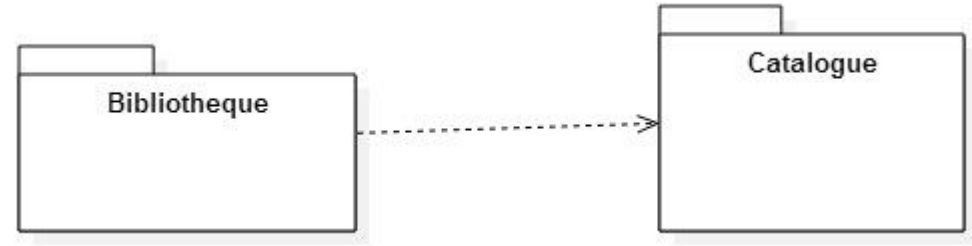
on va mettre l'accent sur les règles de passage du diagramme de classes de conception vers le code Java



Règles de passage: Les Packages

UML





JAVA

```
package catalogue;
…
```

```
package bibliotheque;
import catalogue;
```

Règles de passage: Les classes

UML

JAVA

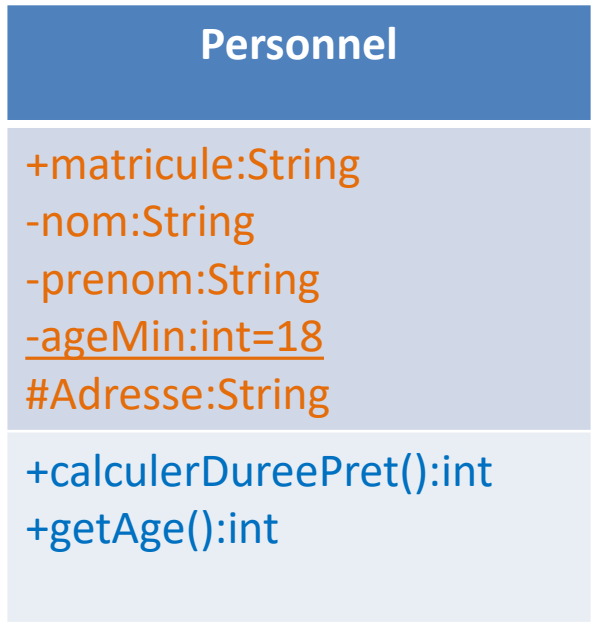
```
public class Personnel {
    …
    }
```

```
abstract public class Agent {
    …
    }
```

```
Public interface
    Irecipent {

    }
```

Règles de passage: Attributs & méthodes

UML

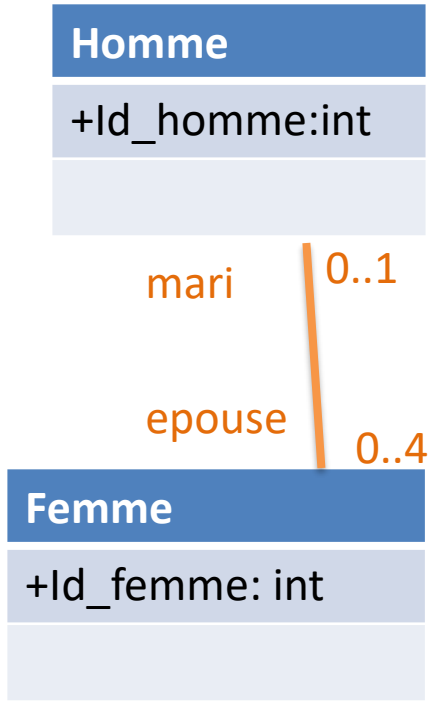


JAVA

```
public class Personnel {

public String matricule;
private String nom;
private String prenom;
private static int ageMin = 18;
protected String Adresse;

public int calculerDureePret()
{.....}
public int getAge()
{...}

}
```

Règles de passage: Associations Binaires « bidirectionnelles »

UML



JAVA

```
public class Homme {
public int Id_homme;
private Femme epouse[];
…
}

public class Femme {
Public int Id_femme;
private Homme mari;
…
}
```

Une association bidirectionnelle se traduit simplement par une paire de références, une dans chaque classe impliquée dans l'association.
Les noms des rôles aux extrémités d'une association servent à nommer les variables de type référence.

Règles de passage: Associations Binaires « navigabilité »

UML

JAVA

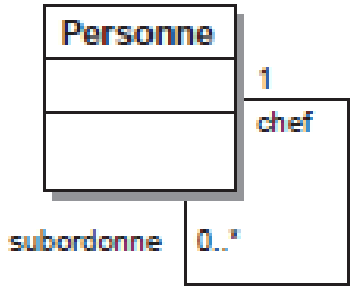
```
public class Personnel {
private Rapport lesRapport[];
…
}

public class Rapport {
…
}
```


```
public class Personnel {
…
}

public class Rapport {
private Personnel lePersonnel;
…
}
```

Les associations navigables se traduisent par du code Java qui dépend de la multiplicité de l'extrémité concernée,

Règles de passage: Associations Binaires « Reflexive »

UML



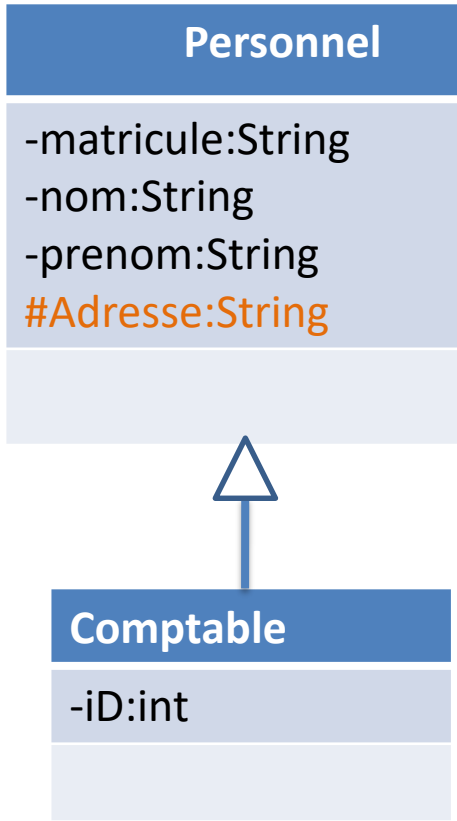
JAVA

```
public class Personne {
private Personne subordonne[];
private Personne chef ;
…
}
```

Une association réflexive se traduit simplement par une référence sur un objet de la même classe

Règles de passage: Associations Particulières «Généralisation »

UML



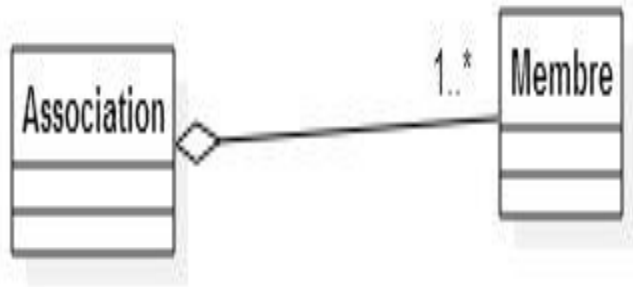
JAVA

```
public class Personnel {

    private String matricule;
    private String nom;
    private String prenom;
    protected String Adresse;

    }



public class Comptable extends Personnel {
    private int iD;
            }
```


Le concept UML de généralisation se traduit directement par le mécanisme de l'héritage dans les langages objet.

Règles de passage: Associations Particulières «Agrégation »

UML



JAVA

```
public class Association {
    private List membre;
    ...
    }

public class Membre {
    private Association monAssociation;
    ...
}
```

Règles de passage: Associations Particulières « Composition »

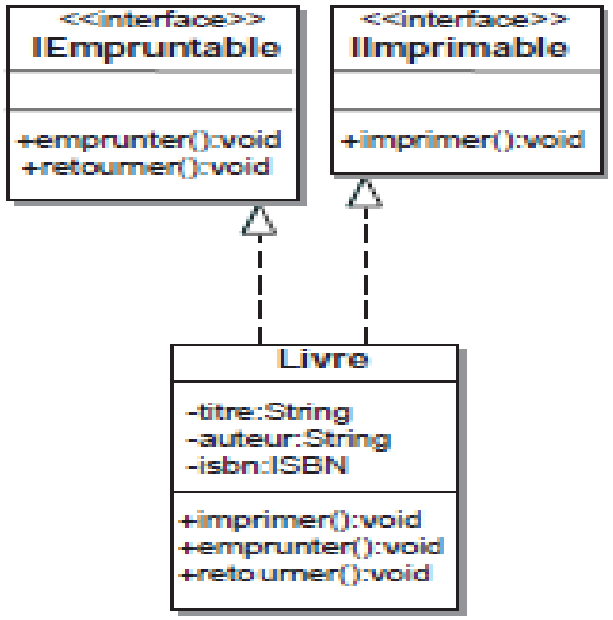
UML

JAVA

```
public class Voiture {
    private Moteur mot;
    public Voiture()
    {
        moteur = new Moteur();
    }
}

public class Moteur {
    private Voiture maVoiture;
    …
}
```

Règles de passage: Associations Particulières « Réalisation »

UML

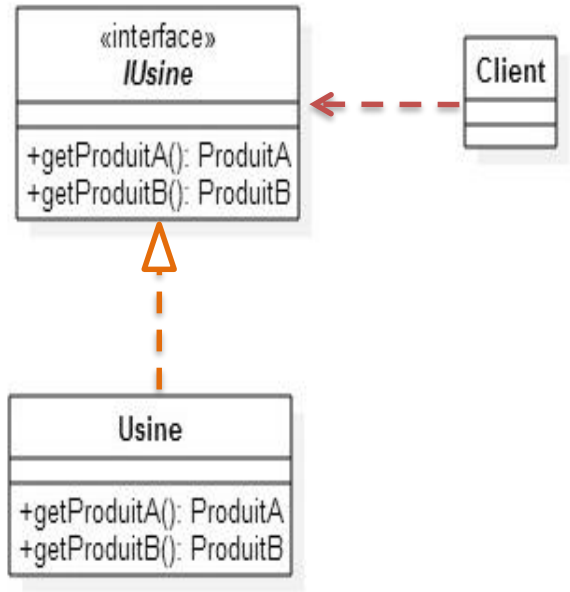


JAVA

```
public class Livre implements IImprimable, IEmpruntable {

        private String titre;
        private String auteur;
        private ISBN isbn;

        public void imprimer(){
        …
        }
        public void emprunter(){
        …
        }
        public void retourner(){
        …
        }
}
```

Une classe UML peut implémenter plusieurs interfaces.

Règles de passage: Associations Particulières « Réalisation+use »




```
class interface IUsine {
……}

 public class Usine implements Iusine{
….. }

public class Client {
IUsine u1 = new Usine();
…….

}
```

Quelques outils de génération de code

- **Papyrus UML (Eclipse Public Licence Multiplateforme)**
  - En Java
  - En C++ avec le composant CDT d'Eclipse.
  - En Ada.
- **StarUML (GPL – Windows uniquement)**

  La génération de code est possible de base en C++, C# et Java.
- **ArgoUML (Eclipse Public Licence – Multiplateforme)**

  Génération de code en Java, C++, C# et PHP.
- **Open ModelSphere (GPL – Multiplateforme)**

  Il est capable de générer du code Java à partir du diagramme de classes.