

Analyse 1

ANA2

Geneviève Cuvelier (CUV)

Christine Leignel (CLG)

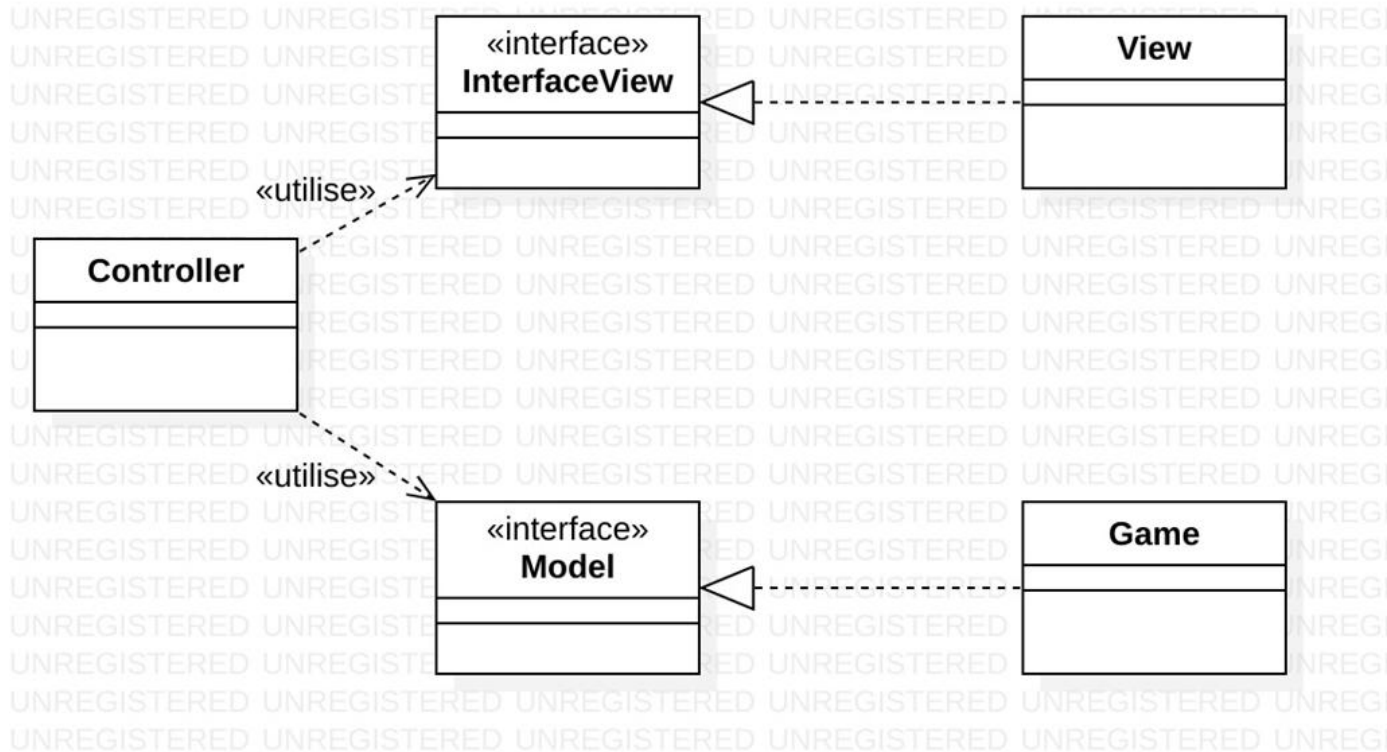
Thibaut Nicodème (TNI)

Pantélis Matsos (PMA)

Où en sommes-nous ?

1. Qu'est-ce que l'analyse?
2. Diagramme d'activités
3. Les classes et objets
4. Les associations 1-1 et 1-N
5. Les associations N-N
6. Les compositions et énumérations
7. Les classes associations
8. L'héritage
9. Les interfaces

Interfaces - Exemple Humbug

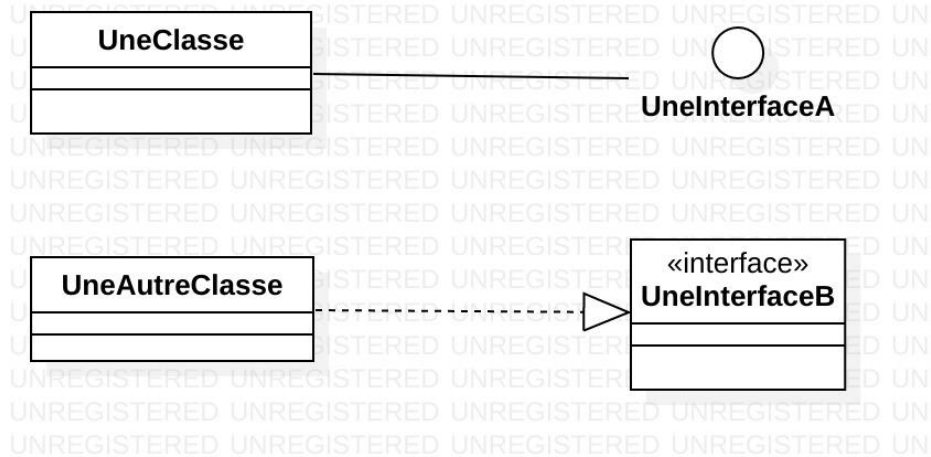


Interfaces - SoC

MVC **Separation of Concerns**

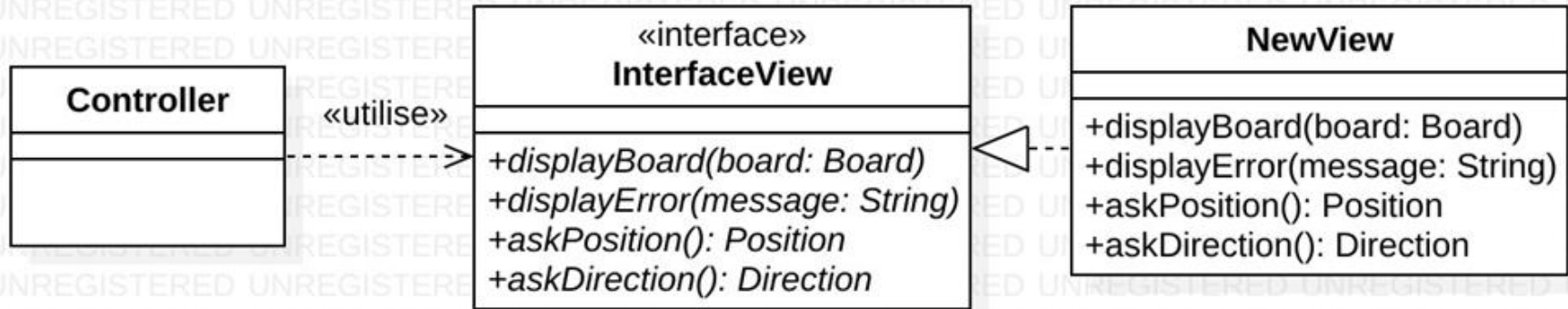
Modularité

Interfaces - Définition



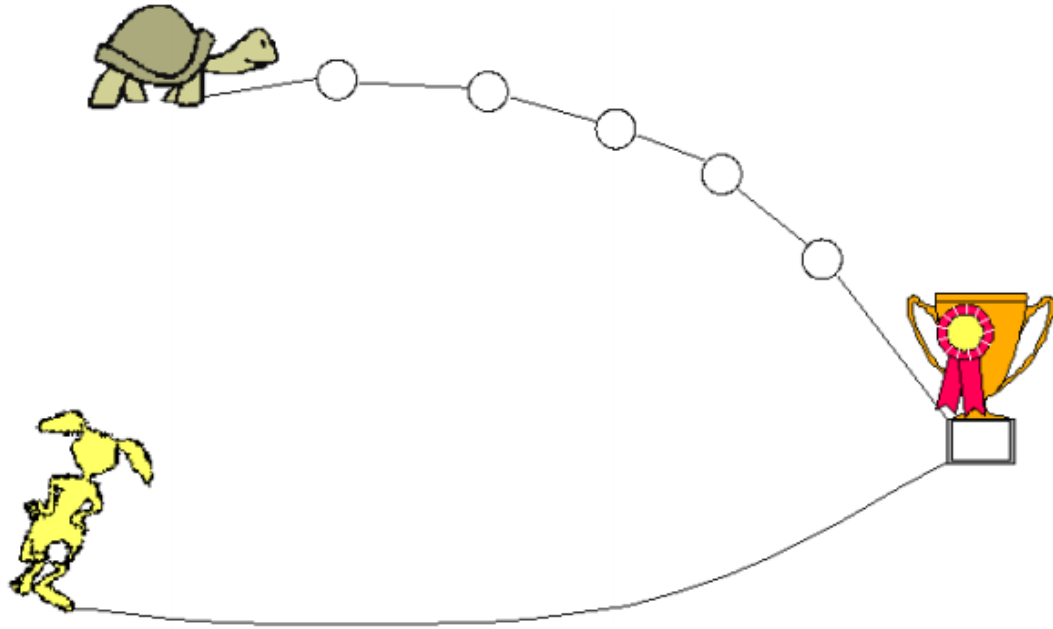
Une interface est une collection de méthodes qui décrit le service d'une classe ou d'un composant.

Interfaces - Exemple Humbug



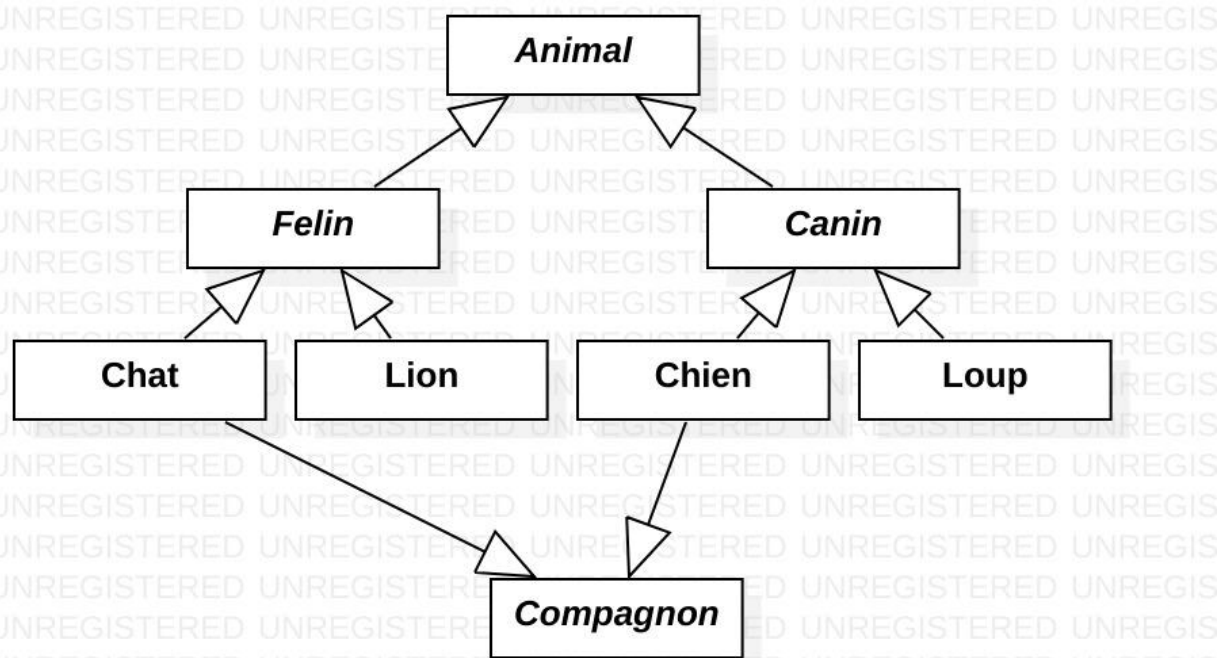
Interfaces - Exercice - Le lièvre et la tortue

MVC

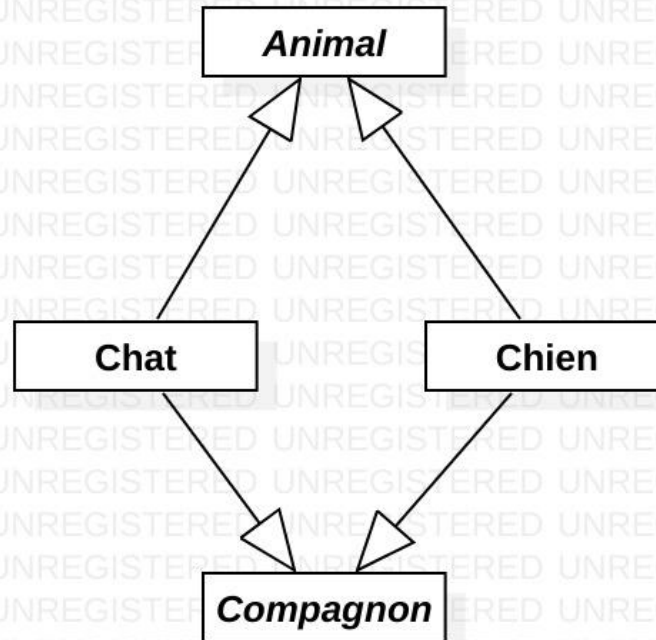


+ Interfaces

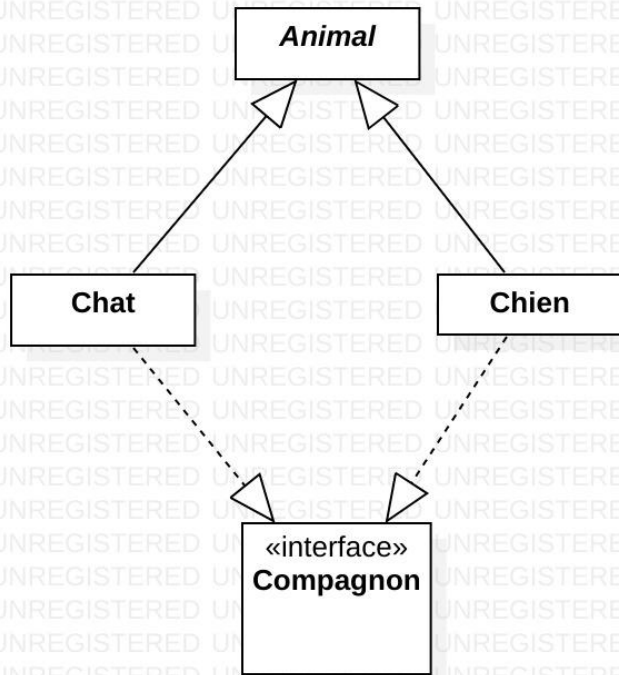
Interfaces - Héritage multiple



Interfaces - Héritage multiple



Interfaces - Héritage multiple



```
1 public class Animal{
2     public void manger() { ... }
3
4 }
```

```
1 public class Chien extends Animal implements Compagnon {
2     public void etreAmical() { ... }
3     public void jouer() {... }
4
5     @Override
6     public void manger() {... }
7 }
```

```
1 public interface Compagnon {
2
3     public abstract void etreAmical();
4     public abstract void jouer();
5 }
```

Interfaces - Résumé

Comment savoir s'il faut

1. une classe
2. une sous-classe,
3. une classe abstraite
4. une interface?



Pensez à

1. un test EST-UN **raté**
2. une version plus spécifique
3. un patron
4. un rôle

Interfaces - Exercice

Faites les 4 diagrammes de classes correspondants

1. `public interface Foo { }`
`public class Bar implements Foo { }`
1. `public interface Ding { }`
`public abstract class Dong implements Ding { }`
1. `public abstract class Zig implements Puce { }`
`public class Alfred extends Zig { }`
`public interface Puce { }`
1. `public class Pim { }`
`public class Pam extends Pim { }`
`public class Poum extends Pam { }`

Diagramme de packages

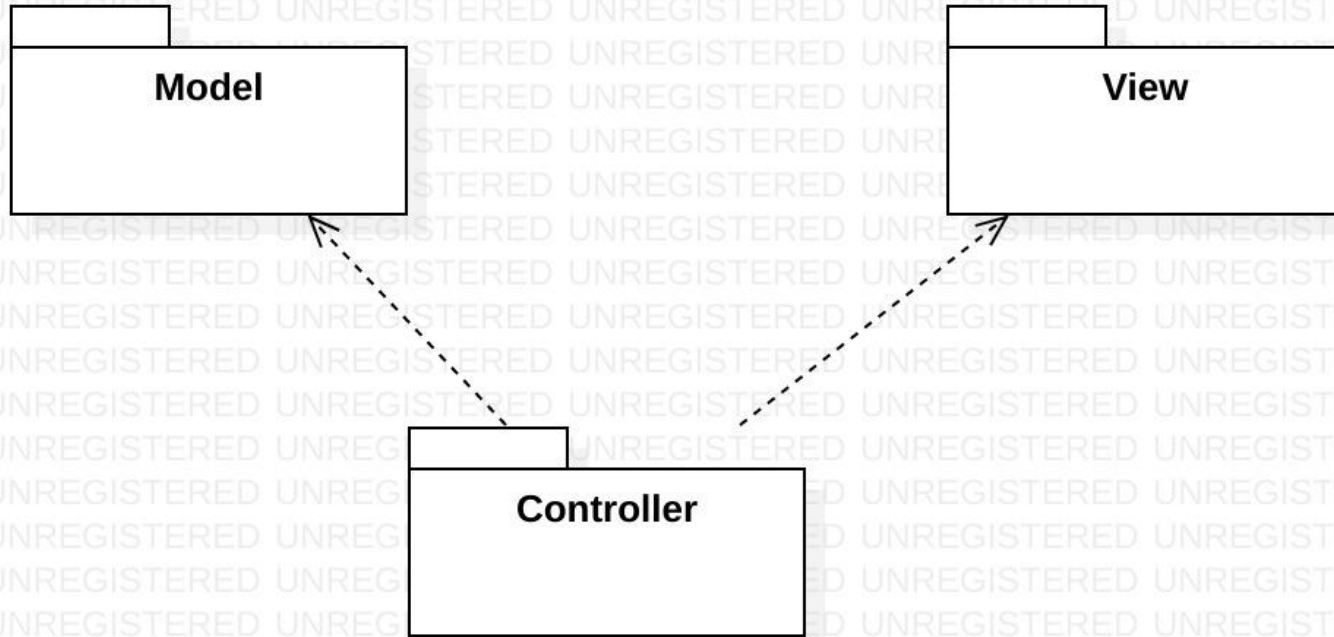


Diagramme de packages -Définition

Un package (paquetage en français) est un élément d'organisation.

- regroupe des éléments (classes, objets, interfaces, package,)
- donne un espace de noms

Diagramme de packages - Exemple Humbug

- package g12345.humbug;
- package g12345.humbug.model;
- package g12345.humbug.view.text;
- package g12345.humbug.controller;

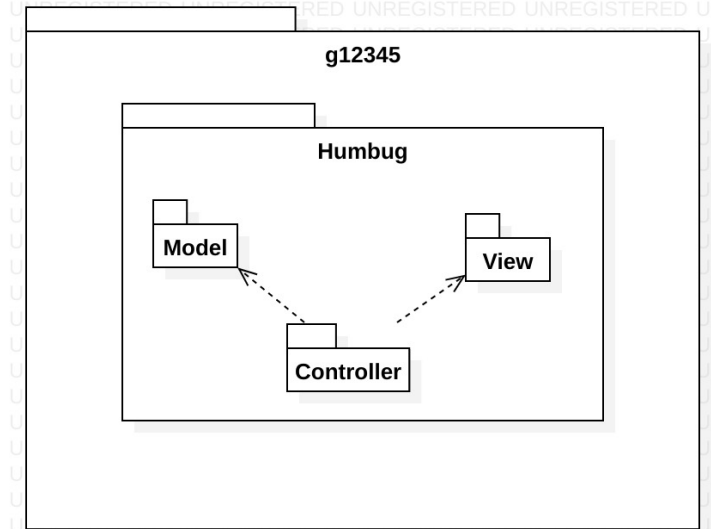
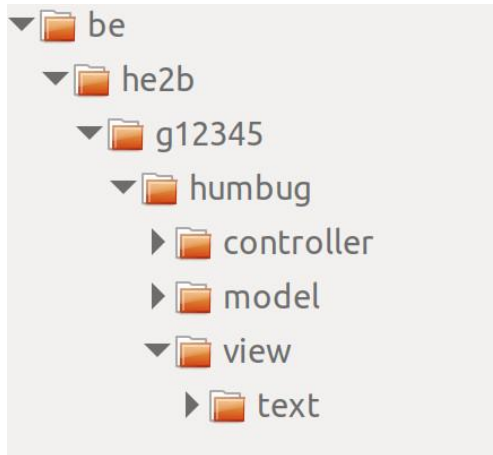


Diagramme de packages - Propriétés

- Un élément ne peut appartenir qu'à un seul package
- Si un package est détruit, tout ce qu'il contient est détruit
- Le package en haut de la hiérarchie est la racine de l'ensemble de l'application.

Diagramme de packages - Dépendances

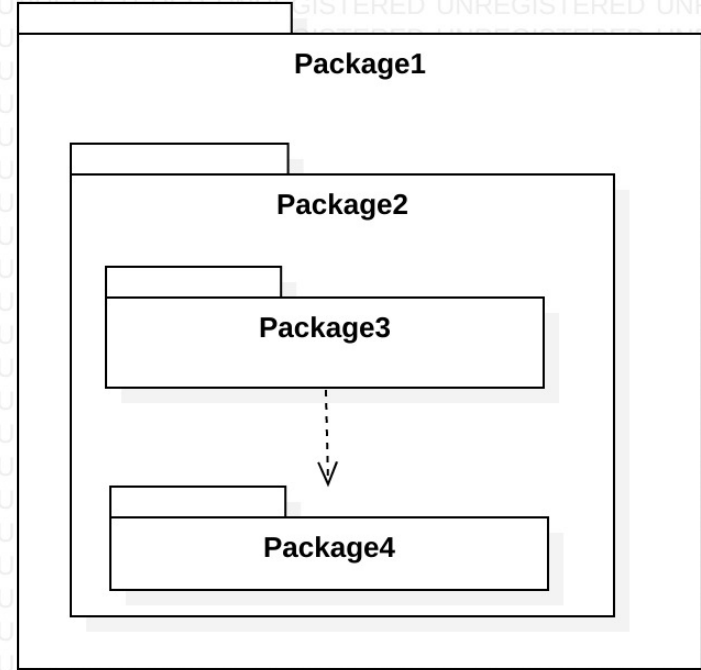
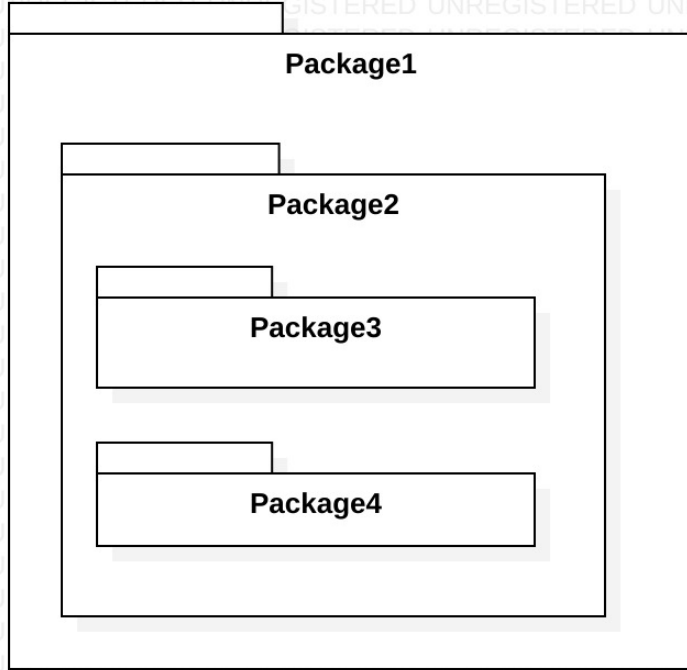


Diagramme de packages - Découpe en package

Deux design pattern seront utilisés

- Forte cohésion : Les éléments assemblés doivent être cohérents entre eux.
- Faible couplage: Les dépendances entre les groupes d'éléments doivent être minimales.

Diagramme de packages - Découpe en package

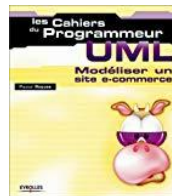
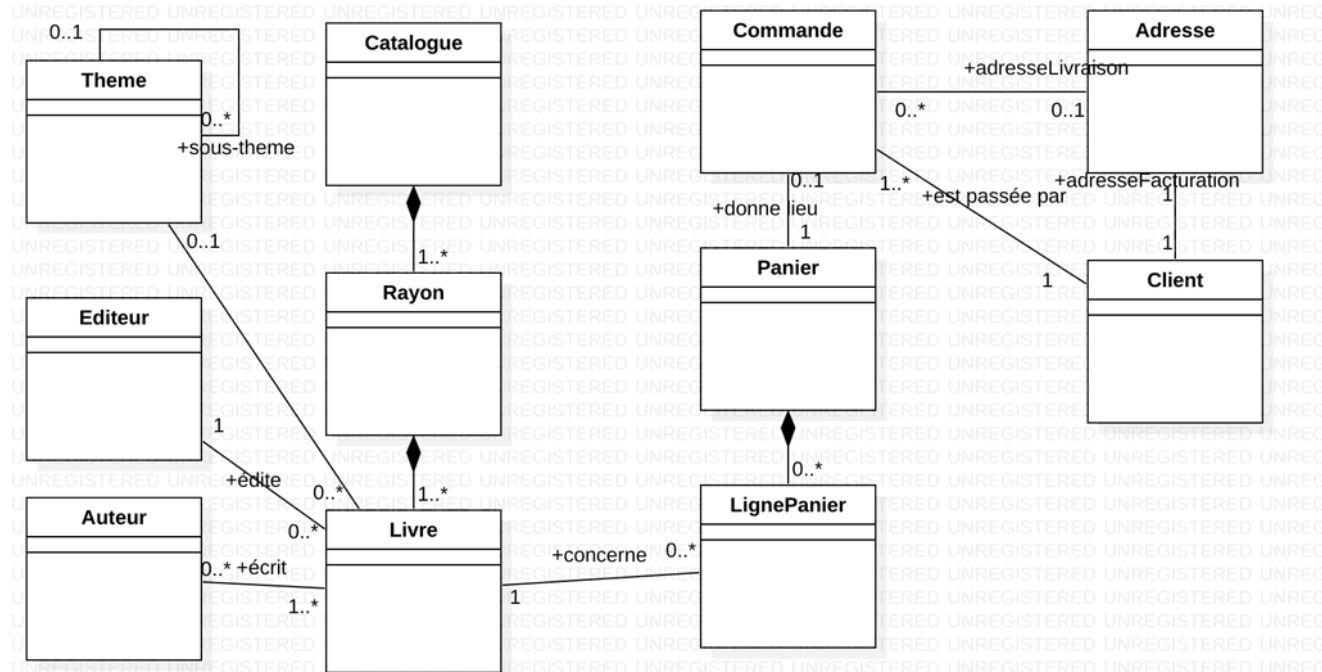
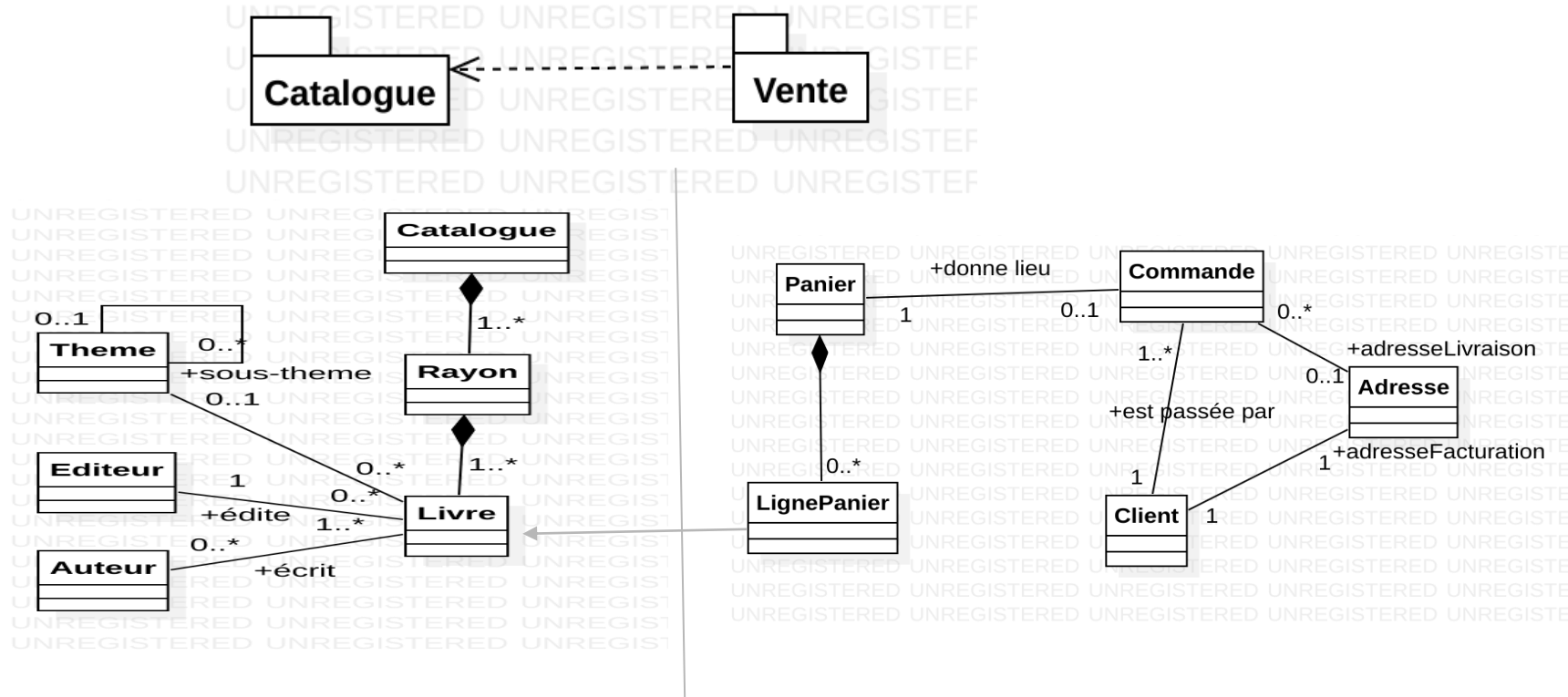


Diagramme de packages - Découpe en package



Interface & Package - Références

Ses slides sont inspirés de

- "The UML user guide" G. Booch, J. Rumbaugh, I. Jacobson
- "Modélisation objet avec UML" P-A Muller, N Gaertner
- "UML Modéliser un site e-commerce" Pascal Roques
- "Java Tête la première" Kathy Sierra & Bert Bates - Ed. O'Reilly