

Cours 1 : UML / JAVA

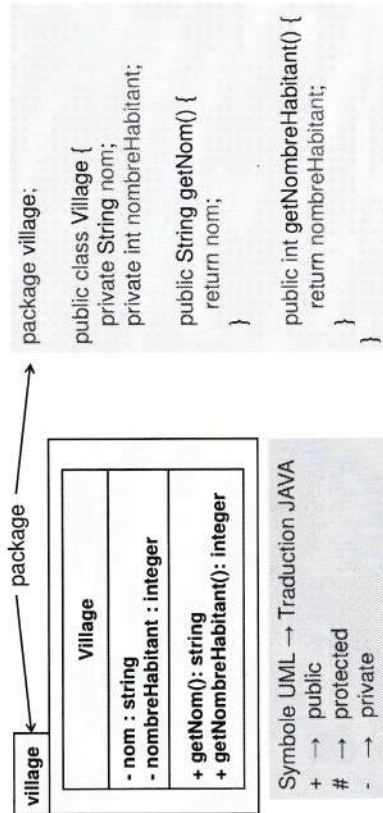
UML :

diagramme de classe, diagramme de séquence
Coder en Java des spécifications UML
Les interfaces

Auteur : CHAUDET Christelle

Parallèle UML/Java

- **Visibilité** = ce qu'un objet a le droit de voir et de modifier. Les droits d'accès de chaque composant sont définis par un mot-clé, dit modificateur.

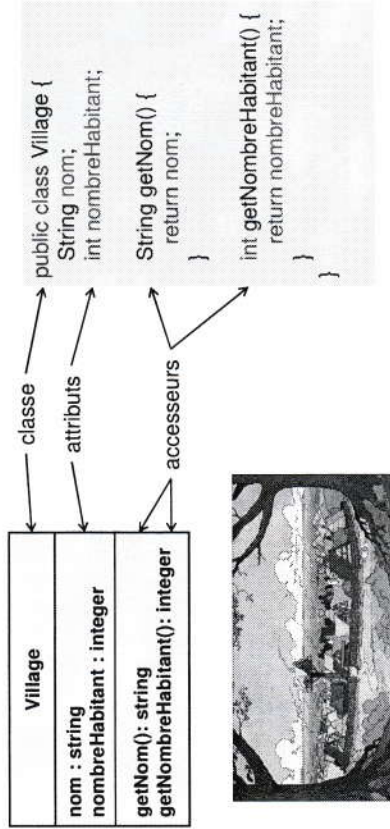


UML : Diagramme de classes / UML : Diagramme de séquence

2

Parallèle Diagramme de Classe /Java

- **Classe** = abstraction d'objets ayant des propriétés (état, comportement et relations) communes

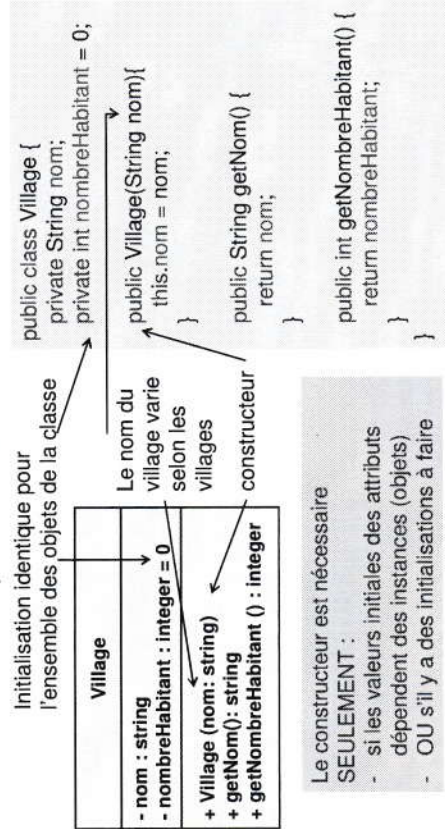


UML : Diagramme de classes / UML : Diagramme de séquence

1

Parallèle UML/Java

- **Constructeur** = permet de créer une instance d'une classe (un objet).

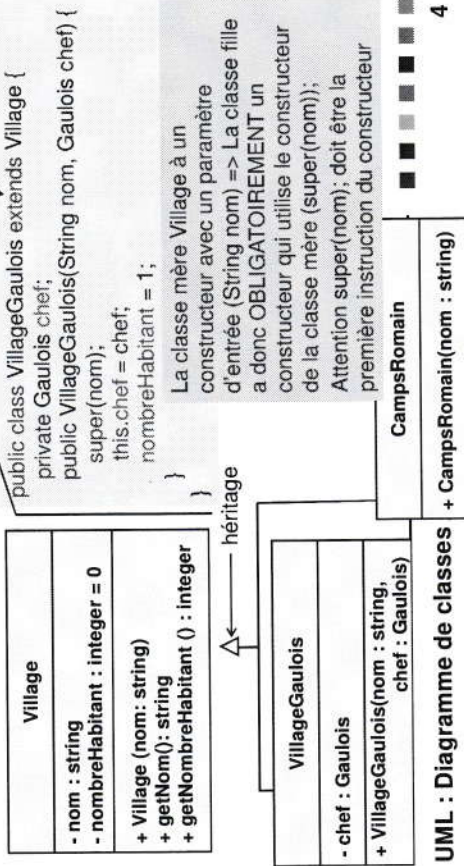


UML : Diagramme de classes / UML : Diagramme de séquence

3

Parallèle UML/Java

- **Héritage** = il s'agit ici de permettre à une classe (dite "classe fille") de récupérer des attributs, des constructeurs et des méthodes d'une autre classe (dite "classe mère")

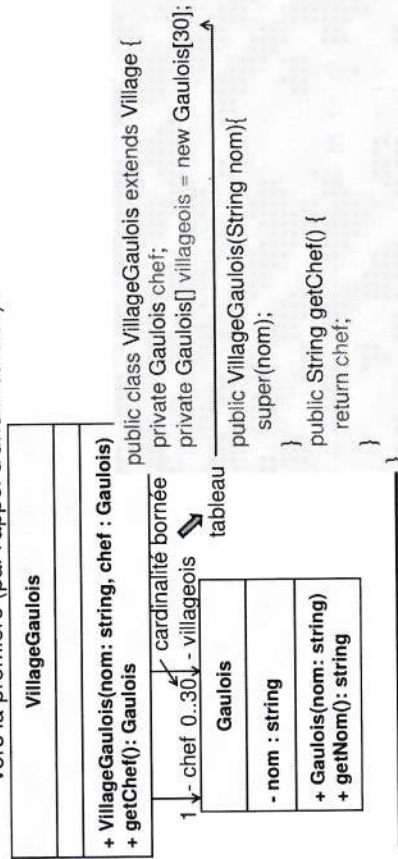


UML : Diagramme de classes / UML : Diagramme de classes

4

Parallèle UML/Java

- **Association** = il y a association (dirigée ou pas) entre deux classes, lorsqu'une des deux classes sert de type d'attribut à l'autre, et donc qu'apparaît dans le code de cette dernière l'envoi d'un message vers la première (par l'appel d'une méthode).

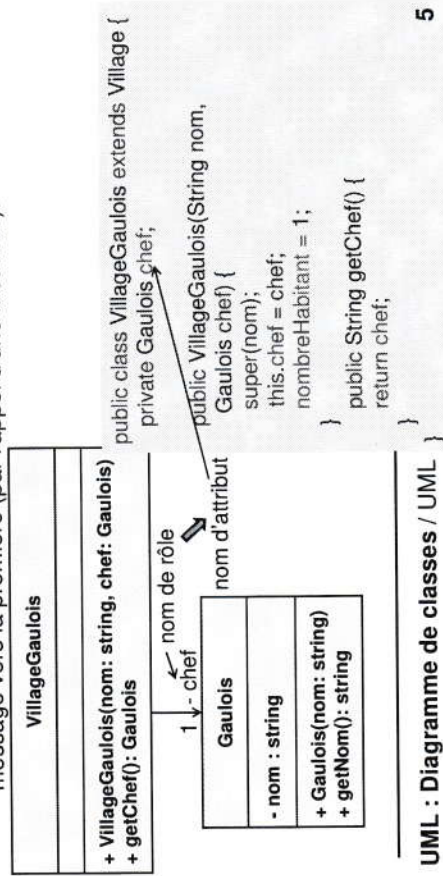


UML : Diagramme de classes / UML : Diagramme de séquence

6

Parallèle UML/Java

- **Association** = il y a association (dirigée ou non) entre deux classes, lorsqu'une des deux classes sert de type d'attribut à l'autre, et donc qu'apparaît dans le code de cette dernière l'envoi d'un message vers la première (par l'appel d'une méthode).

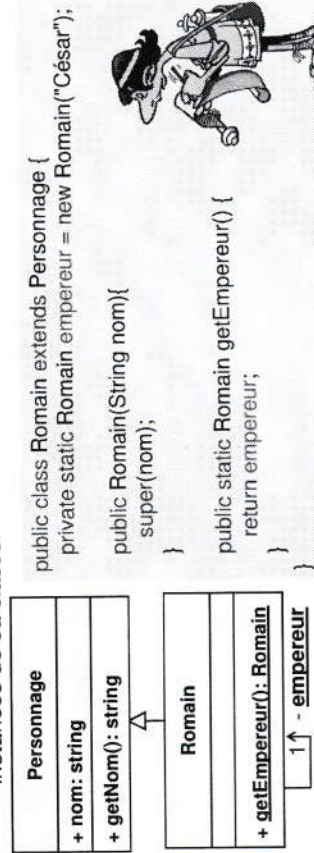


UML : Diagramme de classes / UML : Diagramme de classes

5

Parallèle UML/Java

- **Static** = Pour une méthode, le modifieur static indique qu'elle peut être appelée sans instancier sa classe (NomClasse.methode()). Pour un attribut, le modifieur static indique qu'il s'agit d'un attribut de classe, et que sa valeur est donc partagée entre les différentes instances de sa classe.



System.out.println(Romain.getEmpereur()); ➡ César

UML : Diagramme de classes / UML : Diagramme de séquence

7

Parallèle UML/Java

- **Énuméré** : type de données qui désigne un ensemble fini de valeurs ordonnées. Ces valeurs sont appelées : éléments, membre ou énumérateurs du type énuméré. Existe dans Java à partir de la version 1.5.

```
<<enum>>
GradeRomain {
    PREFET, LEGAT, CENTURION, LEGIONNAIRE;
}
```

Exemple d'utilisation d'un énuméré :

```
public static void main(String[] args) {
    System.out.println(GradeRomain.PREFET);
}
```

Affichage
console

PREFET

UML : Diagramme de classes / UML : Diagramme de séquence

8

Parallèle UML/Java

- Par convention que ce soit en UML ou en JAVA :
 - Les noms de **constants** sont entièrement en majuscule, si le nom est composé de plusieurs mots ils seront accolés par le tiret du 8.
Ex : COEFFICIENT_FORCE
 - Les noms des **types énuméré** commencent par une Majuscule, si le nom est composé de plusieurs mots ils seront accolés en mettant une majuscule à chaque nouveau mot.
Ex : GradeRomain
 - Les noms des **énumérateurs** d'un type énuméré sont entièrement en majuscule, si le nom est composé de plusieurs mots ils seront accolés par le tiret du 8.
Ex : PREFET

UML : Diagramme de classes / UML : Diagramme de séquence

10

Parallèle UML/Java

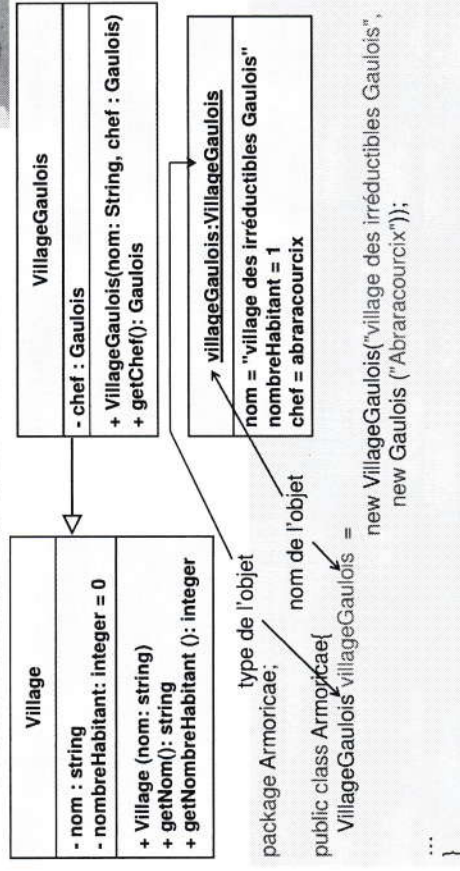
- Par convention que ce soit en UML ou en JAVA :
 - Les noms de **paquetage** sont entièrement en minuscule, Ex : village
 - Les noms de **classes** commencent par une Majuscule, si le nom est composé de plusieurs mots ils seront accolés en mettant une majuscule à chaque nouveau mot.
Ex : VillageGaulois
 - Les noms des **attributs** et des **variables** (attribut ou variable locale à une méthode) commencent par une minuscule, si le nom est composé de plusieurs mots ils seront accolés en mettant une majuscule à chaque nouveau mot.
Ex : nombreHabitant, getNombreHabitant

UML : Diagramme de classes / UML : Diagramme de séquence

9

Les objets

- Pour créer un objet il faut utiliser le constructeur de la classe.

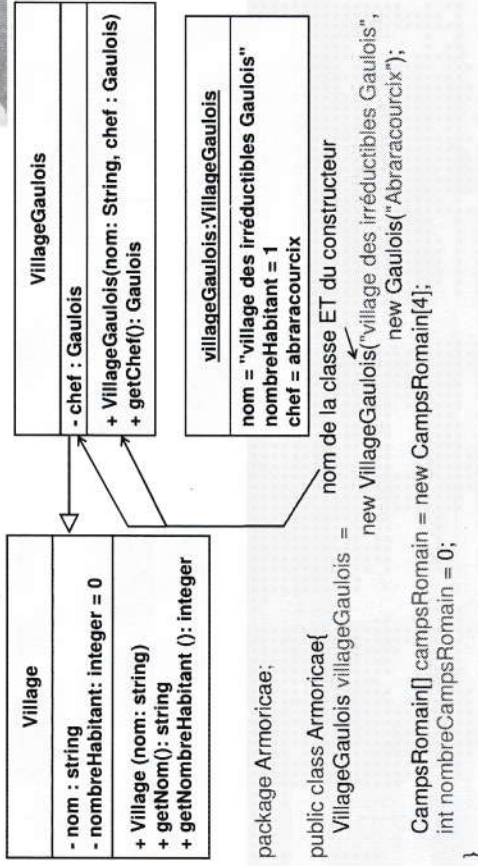


UML : Diagramme de classes / UML : Diagramme de séquence

11

Les objets

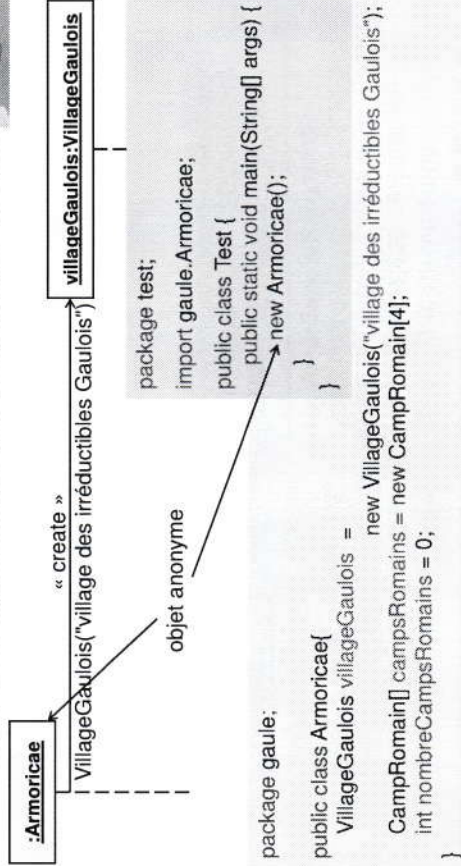
- Pour créer un objet il faut utiliser le constructeur de la classe.



12

Les objets

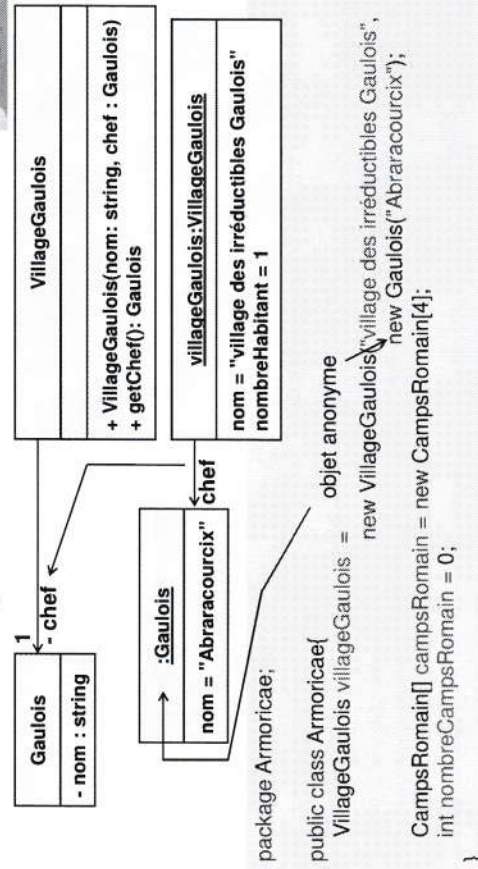
- Pour créer un objet il faut utiliser le constructeur de la classe.



14

Les objets

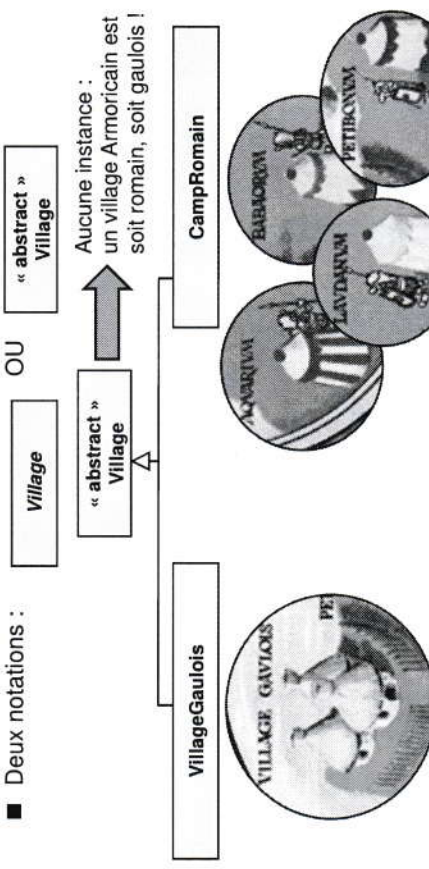
- Pour créer un objet il faut utiliser le constructeur de la classe.



13

Les classes abstraites

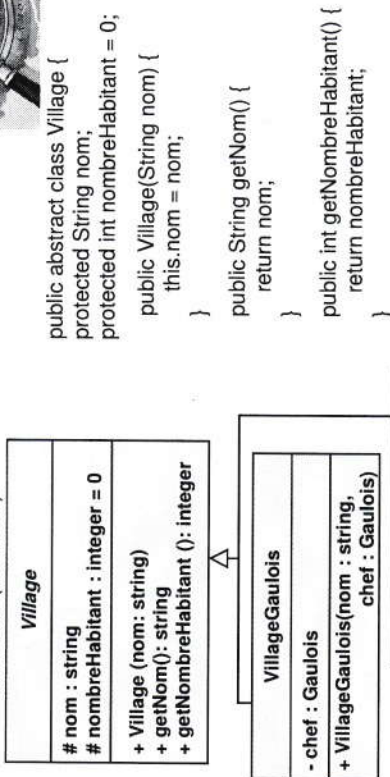
- Classe abstraite : c'est une classe qui n'est pas instanciable. Elle sert de base à d'autres classes dérivées (héritées).
- Deux notations :



15

Les classes abstraites

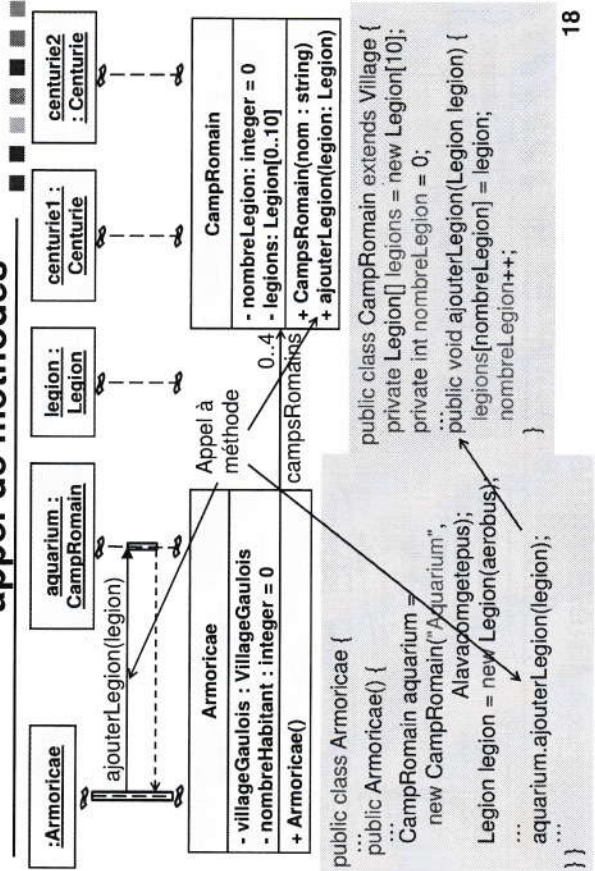
- **Classe abstraite** : c'est une classe qui n'est pas instanciable. Elle sert de base à d'autres classes dérivées (héritées).



UML : Dia + CampsRomain(nom : string) : Diagramme de séquence

16

Interactions entre objets : appel de méthodes



18

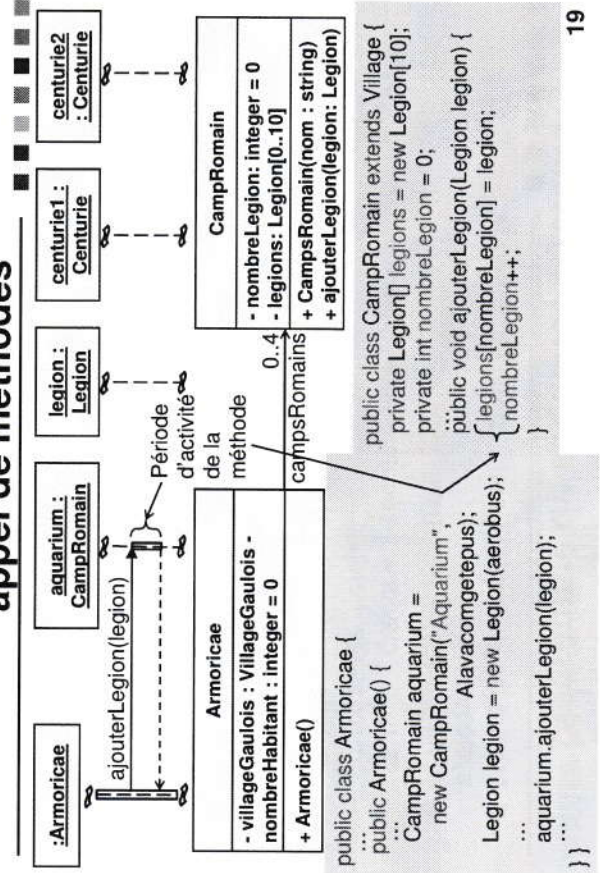
Diagramme de séquence détaillé

- Permet de décrire le comportement interne du logiciel.
- Chacun des :
 - messages seront spécifiés comme des appels à méthode de la classe réceptrice.
 - retours seront spécifiés comme les retours des méthodes. Le nom de la variable de retour est le nom de la variable qui sera affectée par la valeur retournée par la méthode.
- La période d'activité débute quand la méthode est appelée et se termine quand la méthode retourne son paramètre de sortie.

UML : Diagramme de classes / UML : Diagramme de séquence

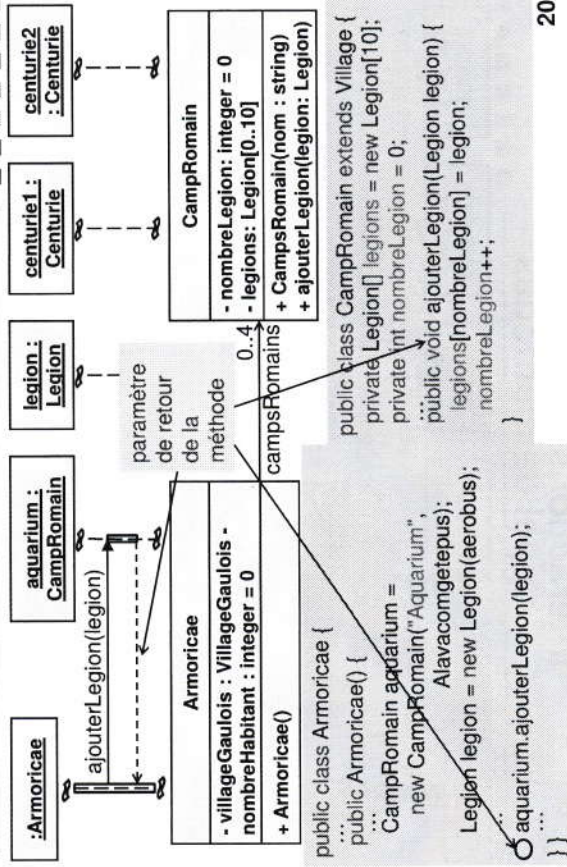
17

Interactions entre objets : appel de méthodes



19

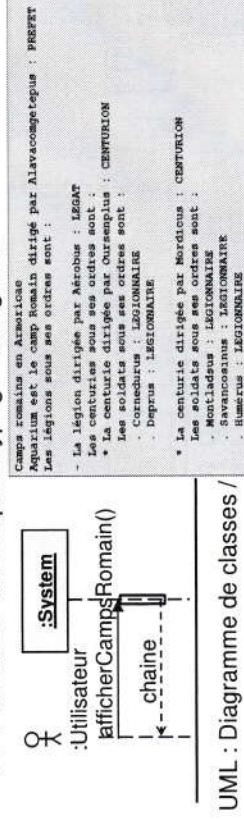
Interactions entre objets : appel de méthodes



20

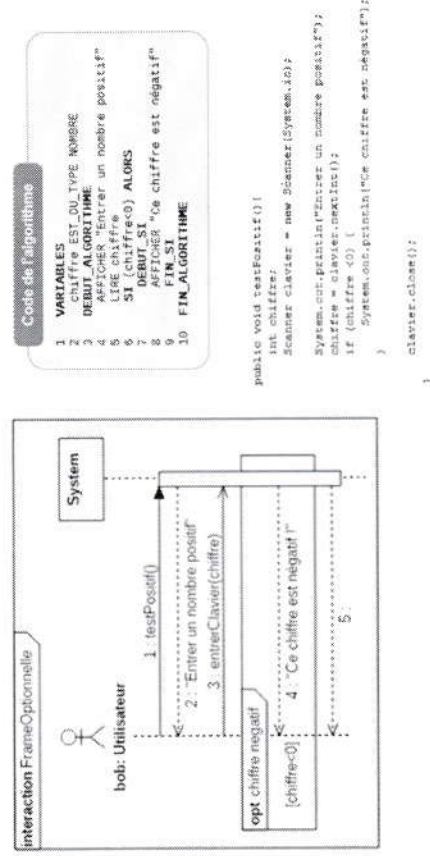
Diagramme de séquence Système

- Montre les interactions entre un (ou plusieurs) acteur(s) et le système.
- Si on pense une interaction :
 - type « clavier » / « écran » on va utiliser comme nom de message « entrerClavier(valeur) »,
 - type interface graphique on va plutôt utiliser des noms de message comme « clicBouton(valeur) »,
 - « entrerTexte(valeur) », « selectionner(valeur) » ...
- Permet de faire le prototype du logiciel.



21

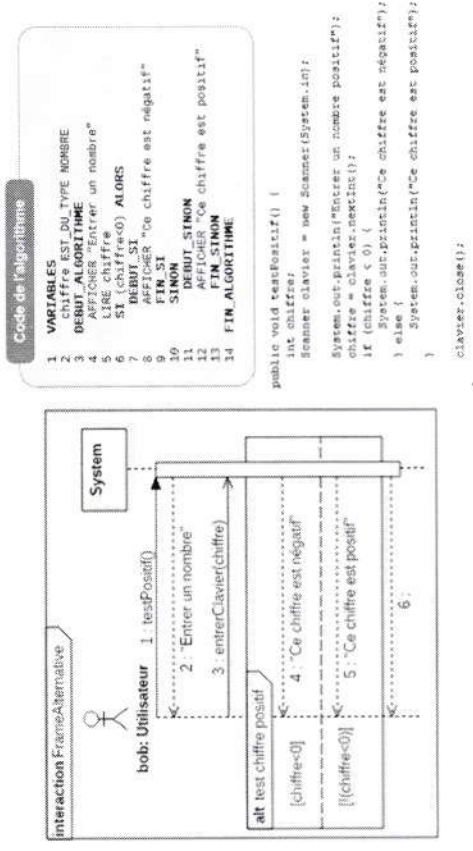
Frame Optionnelle



UML : Diagramme de classes / UML : Diagramme de séquence

22

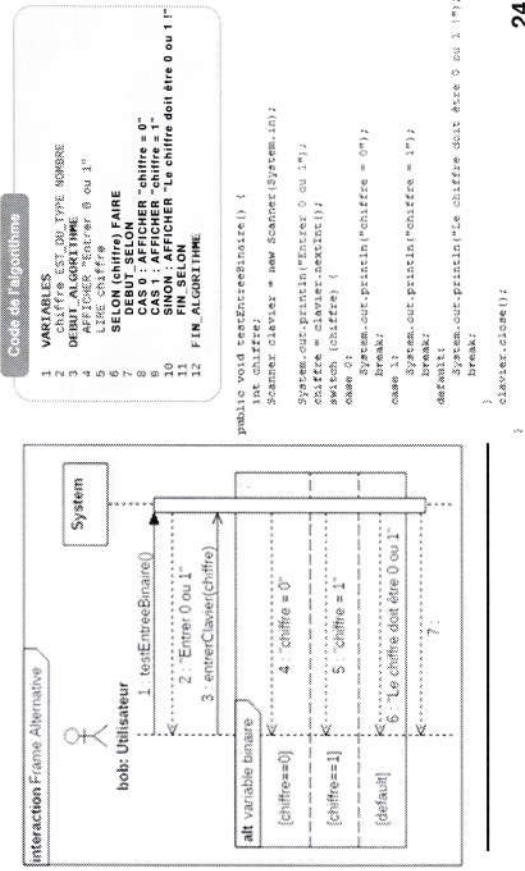
Frame Alternative



UML : Diagramme de classes / UML : Diagramme de séquence

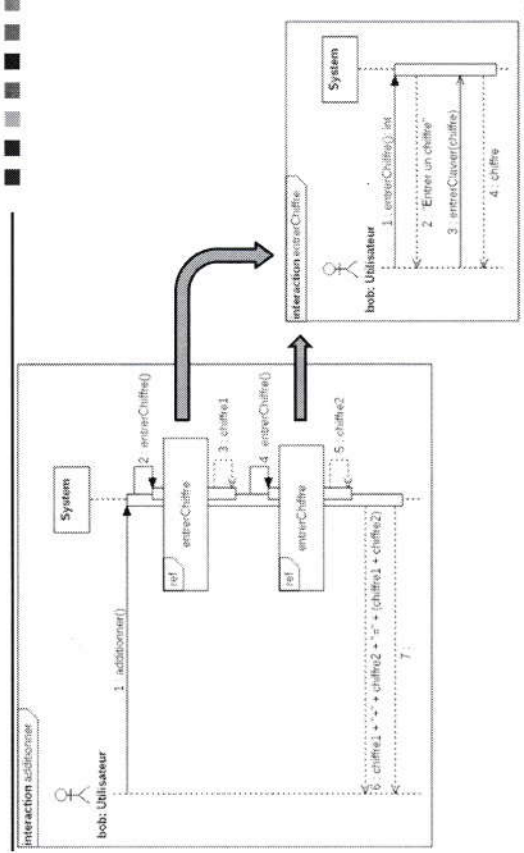
23

Frame Alternative



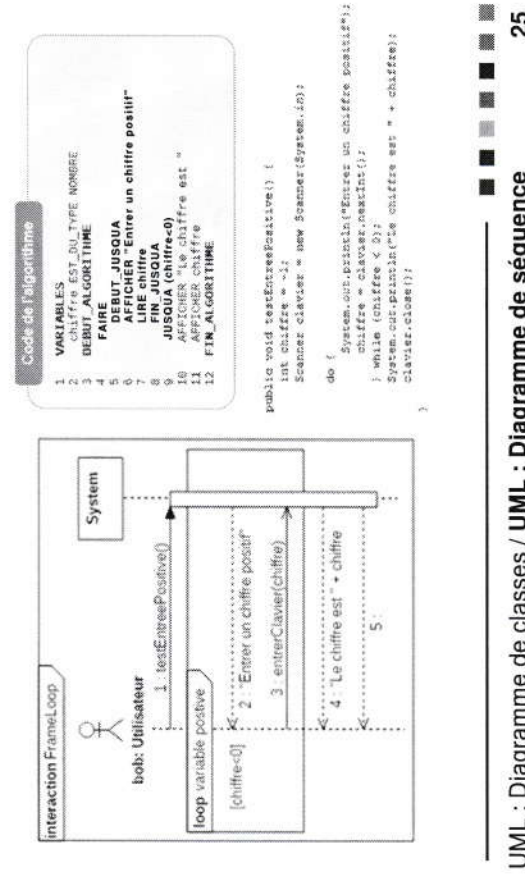
24

Référence



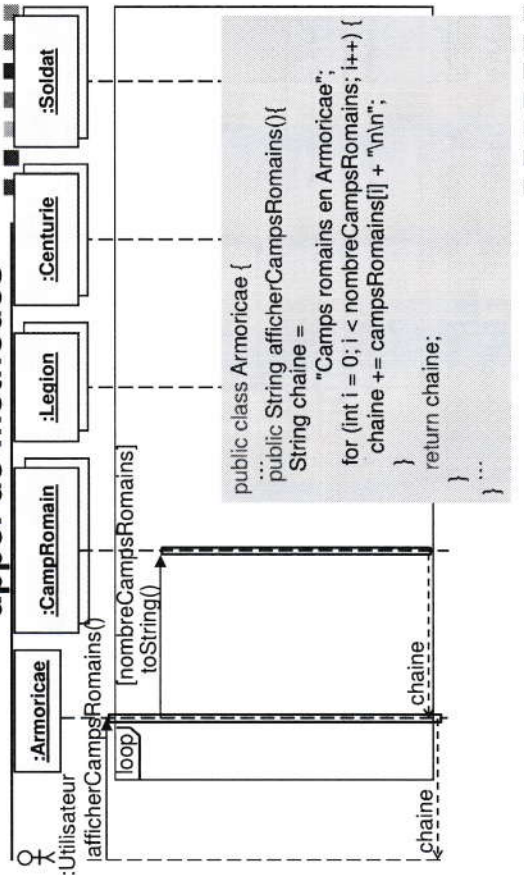
UML : Diagramme de classes / UML : Diagramme de séquence 26

Frame Loop



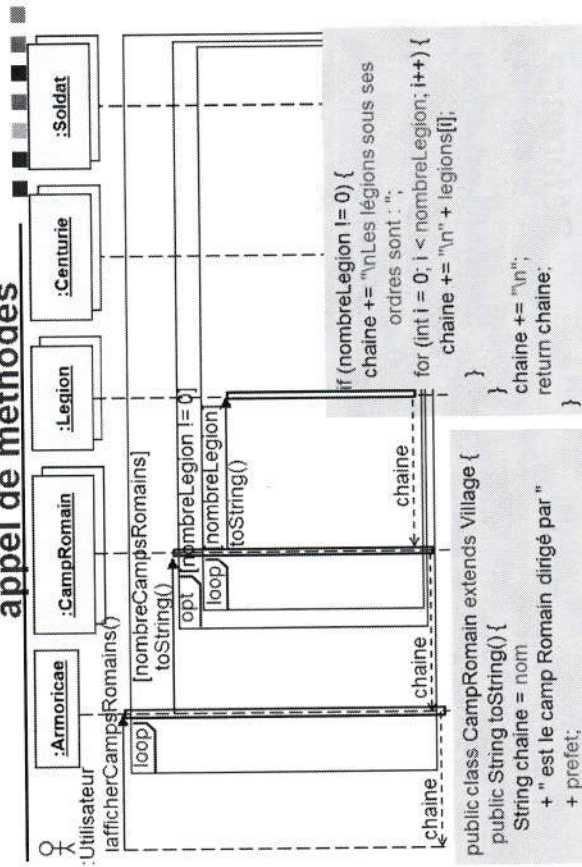
UML : Diagramme de classes / UML : Diagramme de séquence 25

Interactions entre objets : appel de méthodes



UML : Diagramme de classes / UML : Diagramme de séquence 27

Interactions entre objets : appel de méthodes



Interactions entre objets : appel de méthodes

