



Université Ibn Zohr
Faculté des sciences – Département Informatique

Génie Logiciel « Ch04. Conception (suite) »

Ayoub SABRAOUI

Master OTI –S1

Année universitaire 2016/2017

Objectifs de la conception

Objectifs

- Préparer l'implémentation du diagramme de classes et des opérations
- Avoir une vision d'ensemble avant de programmer

Intérêt de la conception abstraite, à base de modèles

- Échapper aux détails et contraintes des langages de programmation
- Vue haut niveau permettant de factoriser les comportements
- Documentation lisible et à part du code

Objectifs de la conception

Préciser et compléter le diagramme de classes

- Préciser les attributs et les associations (visibilité, attributs et associations dérivés, implémentation des associations...)
- Ajouter les classes de contrôle et d'interface

Décrire les opérations complexes

- Par de pseudo-code ou des diagrammes de séquence
- Identifier les méthodes auxiliaires nécessaires
- Identifier le profil précis des méthodes (paramètres, exceptions)
- Répartir les méthodes dans les classes
- Choisir les algorithmes pour implémenter les opérations

Réfléchir à la réalisation des contraintes et des exigences non fonctionnelles (unicité d'un attribut, contraintes de sécurité, de performance...)

3

Diagrammes de séquence (conception)

Objectif : Représenter les communications avec et au sein du logiciel

- Représentation temporelle des interactions entre les objets
- Chronologie des messages échangés entre les objets et avec les acteurs

En conception : Décrire la réalisation des cas d'utilisation sur le système décrit par le diagramme de classes

- Point de vue interne sur le fonctionnement du système
- Description au niveau de l'instance (état du système à un instant)
- Description de scénarios particuliers
- Représentation des échanges de messages
 - Entre les acteurs et le système, entre les objets du système
 - De façon chronologique

4

Diagrammes de séquence (conception)

Éléments du diagramme de séquence

- Acteurs
- Objets (instances)
- Messages (cas d'utilisation, appels d'opération)

Principes de base : Représentation graphique de la **chronologie** des **échanges de messages** avec le système ou au sein du système

- « Vie » de chaque entité représentée verticalement
- Échanges de messages représentés horizontalement

5

Utilisation en phase de conception

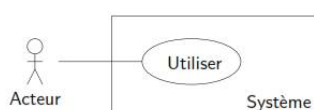


Diagramme de cas d'utilisation

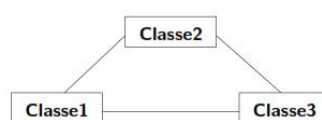


Diagramme de classes

Objectif : Description de la **réalisation d'un cas d'utilisation** sur le Système décrit par le **diagramme de classes**

Problème : **Communication** entre les acteurs et le système vu comme un ensemble d'objets

6

Utilisation en phase de conception

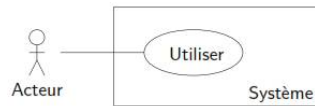


Diagramme de cas d'utilisation



Diagramme de classes



Communication entre acteurs et système via une **interface** (texte, web, physique...)

7

Utilisation en phase de conception



Diagramme de cas d'utilisation

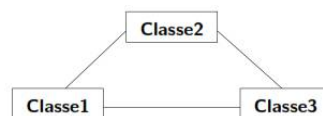
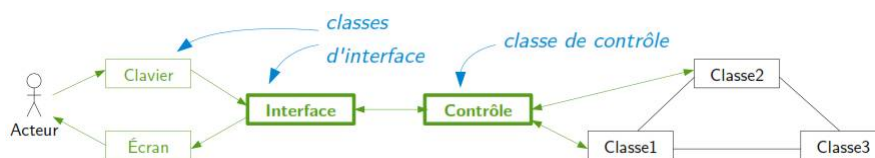


Diagramme de classes



Communication entre acteurs et système via une **interface** (texte, web, physique...)

Solution : Création de **classes de contrôle** et de **classes d'interface** qui :

- gèrent les interactions avec les acteurs
- encapsulent le résultat des opérations

8

Utilisation en phase de conception

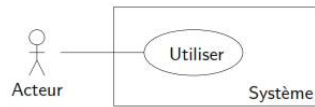


Diagramme de cas d'utilisation

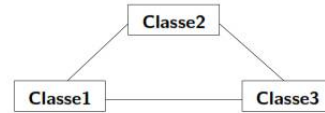


Diagramme de classes

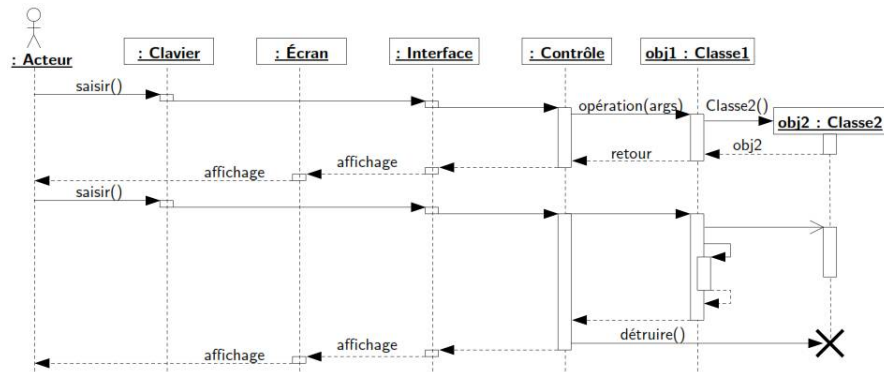
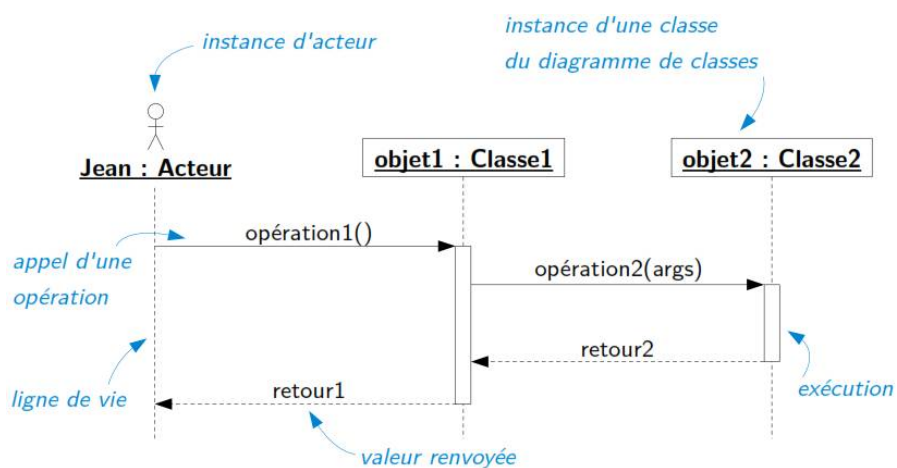


Diagramme de séquence du cas d'utilisation Utiliser

9

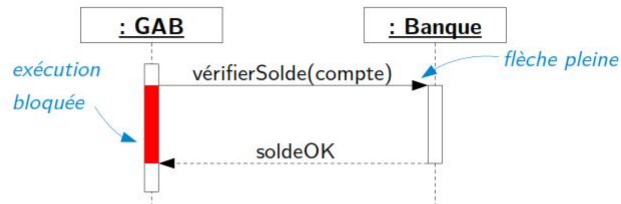
Eléments de base



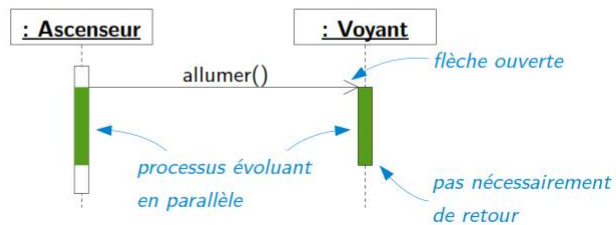
10

Types de messages

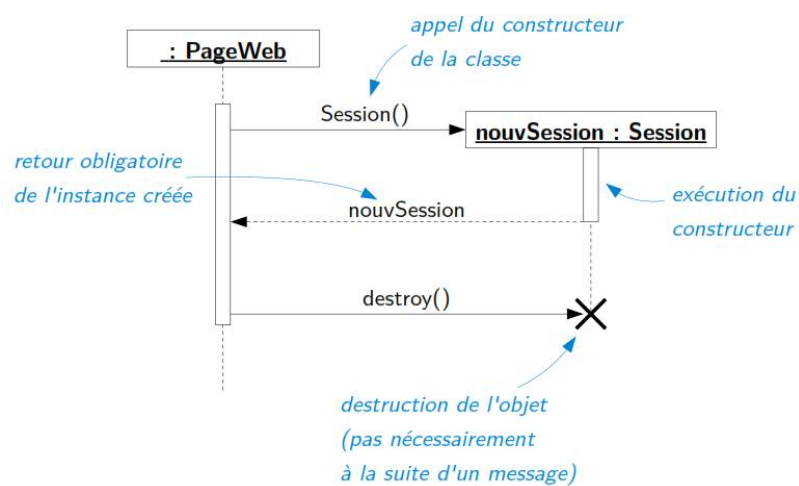
Message synchrone : Émetteur **bloqué** en attente du retour



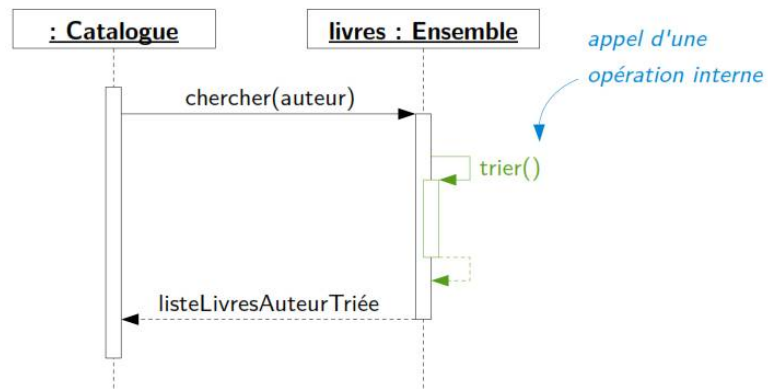
Message asynchrone : Émetteur **non bloqué**, continue son exécution



Création et destruction d'objet



Message réflexif



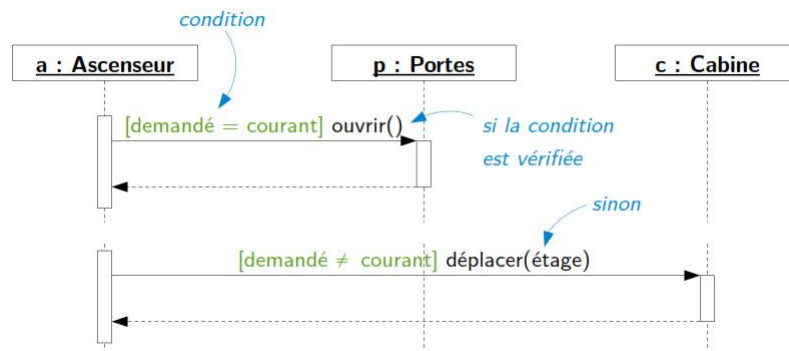
13

Alternative

Principe : Condition à l'envoi d'un message

Notation :

- Deux diagrammes



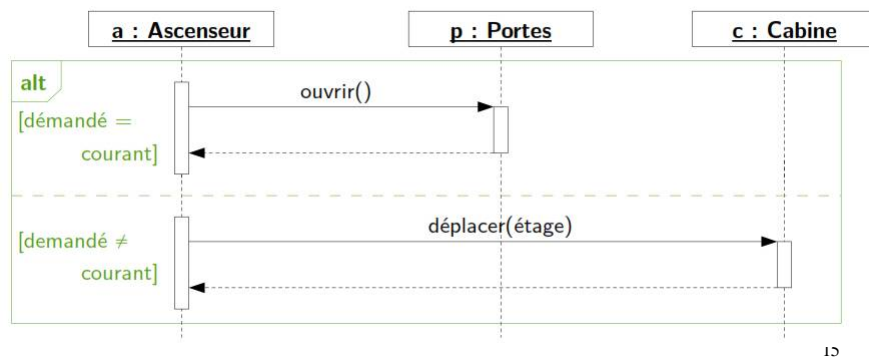
14

Alternative

Principe : Condition à l'envoi d'un message

Notation :

- Deux diagrammes
- Bloc d'alternative **alt**

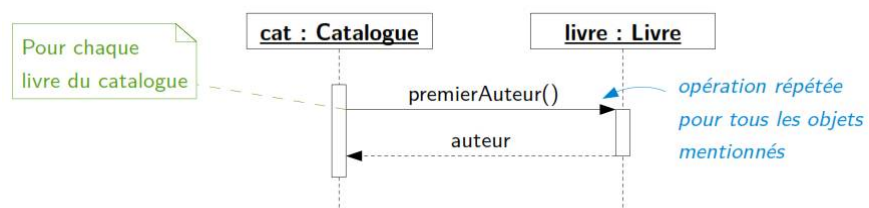


Boucle

Principe : Répéter un enchaînement de messages

Notation :

- Note

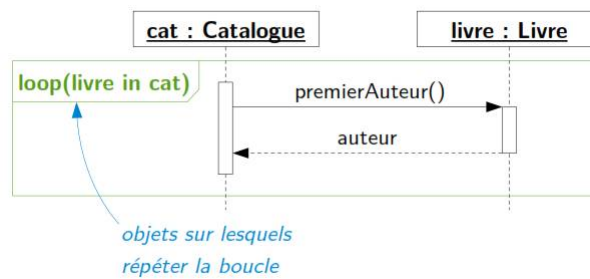


Boucle

Principe : Répéter un enchaînement de messages

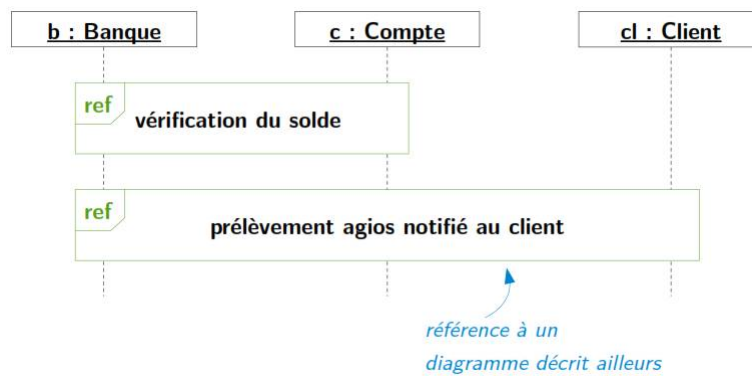
Notation :

- Note
- Bloc de boucle **loop**



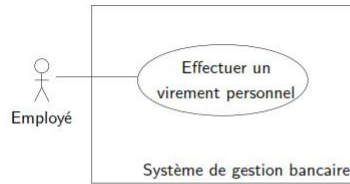
17

Référence à un autre diagramme



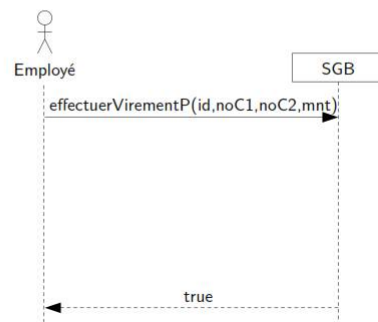
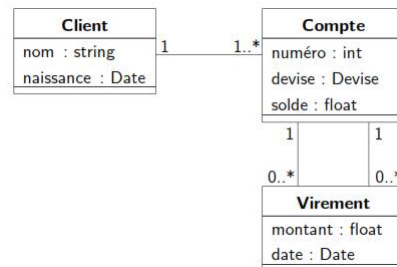
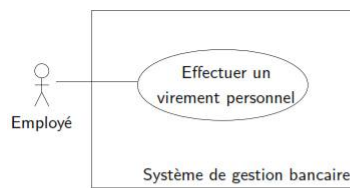
18

Exemple – Analyse



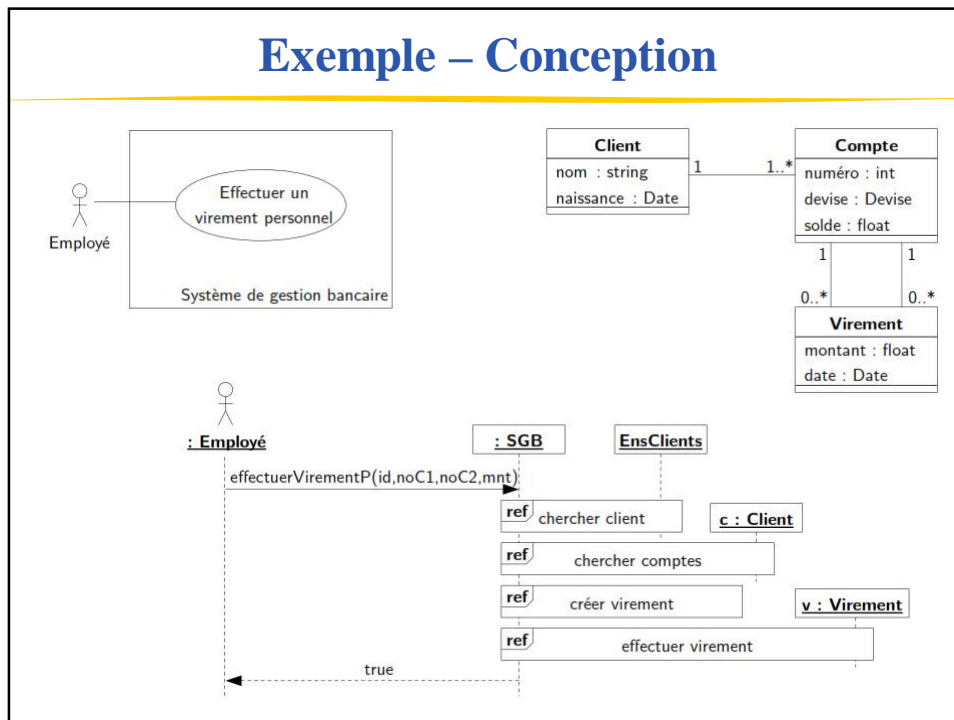
19

Exemple – Conception

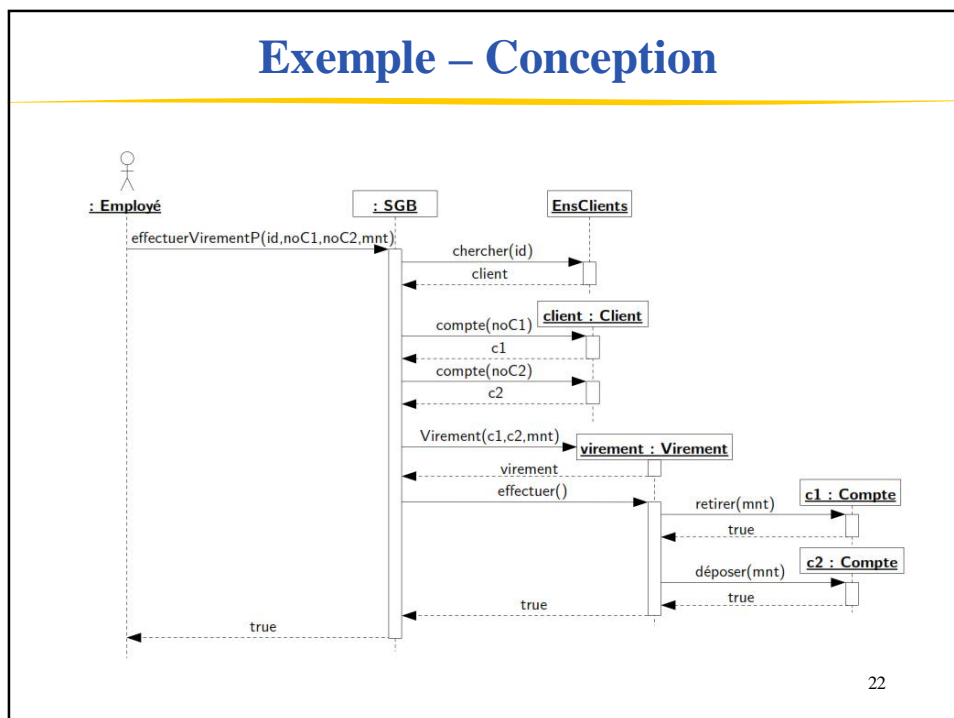


20

Exemple – Conception



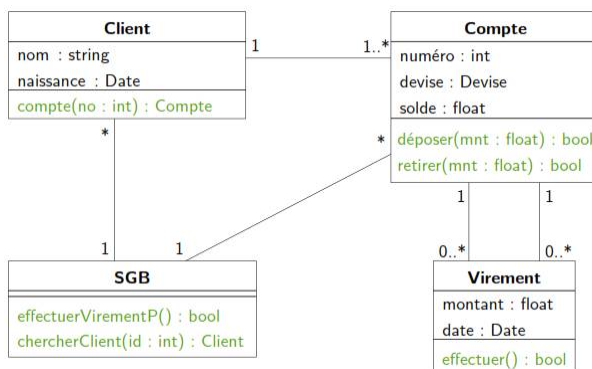
Exemple – Conception



22

Exemple – Conception

Diagramme de classes **complété** avec les opérations nécessaires



23

Quelques règles

Messages entre acteurs et interface

- Opérations liées au **cas d'utilisation** (même nom)
- Arguments (saisis) et valeurs de retour (affichées) **simples**: texte, nombre

Messages au sein du système

- **Opérations** du diagramme de classes
- Si message de **objA : ClasseA** vers **objB : ClasseB**, alors opération du message dans ClasseB

24