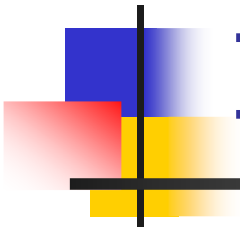


Introduction au Génie Logiciel

Lydie du Bousquet

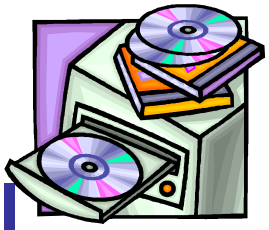
Lydie.du-bousquet@imag.fr

En collaboration avec J.-M. Favre, I. Parissis, Ph. Lalanda

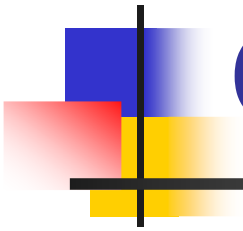
Qu’est-ce que le logiciel ?

programme, ensemble d’instructions

Caractéristiques du logiciel



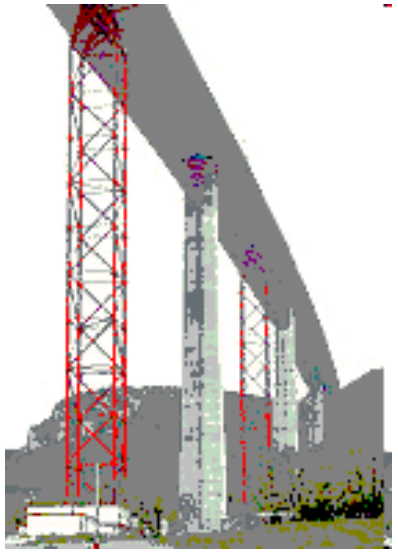
- Abstrait
  - Difficile à représenter/visualiser
- Malléable
  - A priori facile à modifier/corriger **mais**
- Souvent complexe et de grande taille
  - Besoin de gérer la complexité
  - « Effets de bord »
- Mystérieux pour les utilisateurs
  - « ça bogue », « 'sais pas comment ça marche »



Qu’est-ce que le génie logiciel ?

Qu’est-ce que le génie civil ?

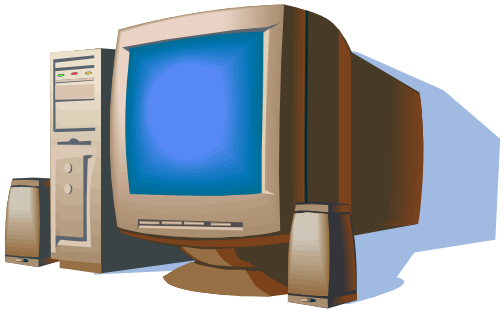
- Ensemble de techniques concernant les constructions civiles
  - Conception
  - Réalisation
  - Exploitation
  - Réhabilitation d’ouvrages



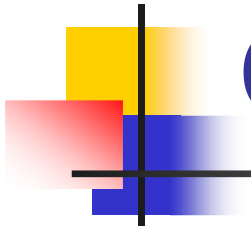
  - Répondre aux besoins de la société,
  - Tout en assurant la sécurité du public
  - Et la protection de l’environnement.

Qu’est-ce que le génie logiciel ?

- Ensemble des techniques concernant les constructions de logiciels
  - Conception
  - Réalisation
  - Exploitation
  - ~~Réhabilitation d’ouvrages~~ Maintenance



  - Répondre aux besoins des utilisateurs,
  - Tout en assurant leur sécurité
  - ~~Et la protection de l’environnement~~.

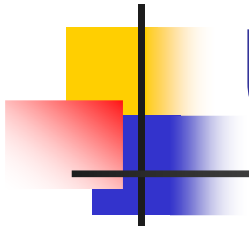


Génie Logiciel (Sommerville)

- An engineering discipline concerned with all aspects of software production
  - from early stages of specification
  - to maintaining the system after it has gone into use
- The purpose is to rigorously applies methods and techniques to produce timely, quality, satisfying software

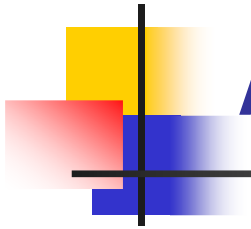
Comme pour le Génie civil, en GL, il existe des méthodes

- Pour récolter les besoins
- Estimer les coûts
- Concevoir, développer, valider
- Pour organiser les différentes étapes
- Planifier/suivre le travail



Une question de bon sens…

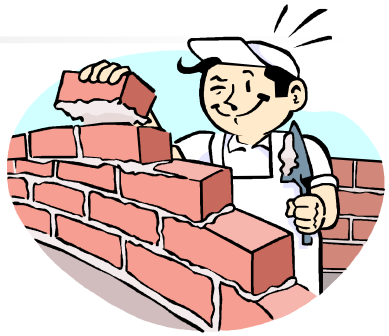
- Récolter les exigences
  - Parce qu'il faut savoir ce que le client veut avant de commencer
- Exprimer les besoins
  - Parce qu'il faut s'assurer que l'on a compris
- Concevoir, développer
- Valider, documenter
  - Parce qu'il faut montrer que le travail a été bien fait
- Installer, Former, Maintenir
  - Parce que le système va être utilisé…



Au programme de GL2

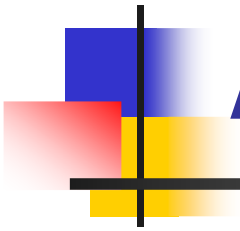
- Pour récolter les besoins
- Estimer les coûts
- Concevoir, développer, valider
- Organiser les différentes étapes
- Planifier/suivre le travail

Au programme de GL1 : outils de base

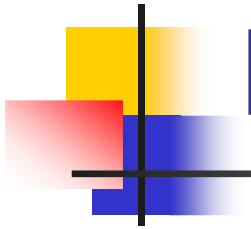


- Notion de conception objet
- Notion de modèle
- Langage pour exprimer les modèles (UML)

- Travail d'analyse

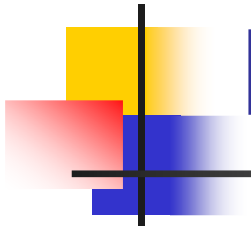


Approche orientée-objet



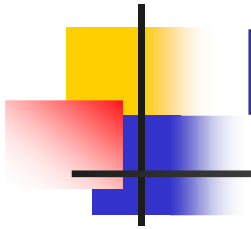
Plan

- Notion d’objet
- Principe d’encapsulation
- Principe de classification



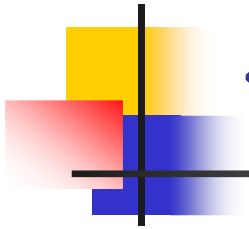
Pourquoi l'approche objet

- Depuis 1960, plus en plus de programmes
- Beaucoup de similarités
- Vers un besoin de réutilisation
  - Programmes non ou peu structurés (assembleur)
  - Blocs, fonctions/procédures, modules, packages
  - Objets
  - Composants, services
  - Patrons (de conception, d'architecture, …)



Différentes approches

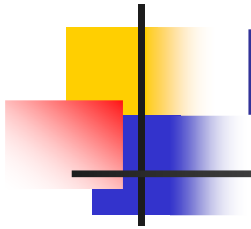
- Approche fonctionnelle
  - Orientée traitement
  - Diviser pour régner
  - Réutilisation et évolution difficiles
- Approche objet
  - Objet = données + traitements associés
  - Monde réel => modèle du monde
  - Réutilisation et évolution a priori simplifiée



« Orienté objet »

Représenter un système comme
un ensemble d'objet autonomes
qui interagissent par messages

- Chaque objet
  - représente un concept
  - a une identité unique
  - a un état
  - propose des « services » = méthodes
  - exécute un service



Exemples d'objets

Pierre

Marie

Paul

Le compte de Pierre

Le compte de Pierre et de marie

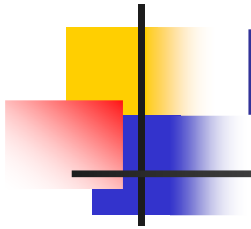
La carte bancaire de Paul

Le compte de Paul

le distributeur

La réserve de billets

la banque



Principe de réification

- Matérialiser un concept par un objet
- Permet de manipuler le concept

- En mathématique, on choisit, puis on nomme des variables : « soit x le nombre de pommes… »

- Un concept abstrait
  - L'événement « à 10h45, la carte a été introduite »
- Une relation entre 2 objets
  - « Jean possède la voiture 555 BBB 38 »

Attributs et méthodes

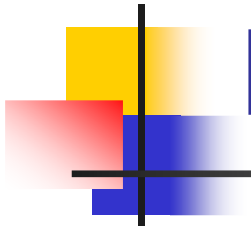
Chaque objet

- A un état caractérisé par la valeur de ses attributs
- Propose des services sous la forme de méthodes

La banque
numéro = 2453 nom = « banque du sud » …
CréerUnCompte SupprimerUnCompte …

Le compte de Paul
numéro = 88219 solde = 2000 découvertMax=-500
ConsulterSolde Créditer Débiter

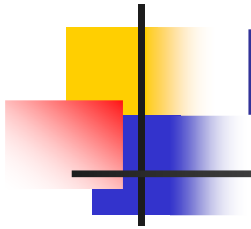
Le compte de Pierre
numéro = 88213 solde = 10 découvertMax=-100
ConsulterSolde Créditer Débiter



Envoi de message

Les objets interagissent par envoi de messages

Principe d'encapsulation

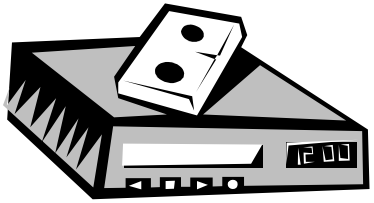


Le compte de Paul
numéro = 88219 solde = 2000 découvertMax=-500
ConsulterSolde Créditer Débiter

- Un objet n'expose jamais ses attributs directement
  - Solde d'un compte ne peut pas être modifié par un autre objet directement
- Il propose des méthodes pour le faire
  - Consulter, créditer, débiter
- Il indique les méthodes qu'il offre (interface)
- Mais cache la réalisation concrète des méthodes
  - Les autres objets ne connaissent pas le programme (code) qui effectue le débit

Grâce à l'encapsulation

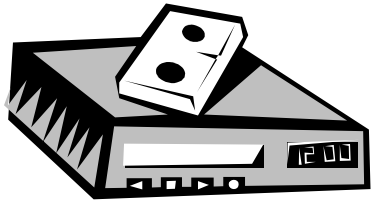
- Un client ne connaît que l'interface
  - L'objet est plus facile à comprendre
    - Il suffit de comprendre les commandes du magnétoscope
  - à réutiliser
    - Connexion avec les autres appareils ok (si l'interface est bien faite)
  - Il est protégé contre les mauvaises utilisations
    - Enlever La K7 alors que l'appareil est en lecture

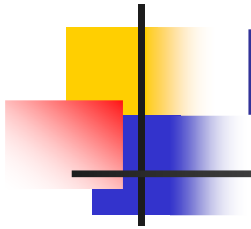


Grâce à l'encapsulation

La réalisation de l'objet peut-être modifiée sans impact sur le client

- On peut améliorer l'objet
  - Remplacer un composant
  - Mettre de nouvelles sorties
- On peut le remplacer par un équivalent
  - Changer de magnétoscope

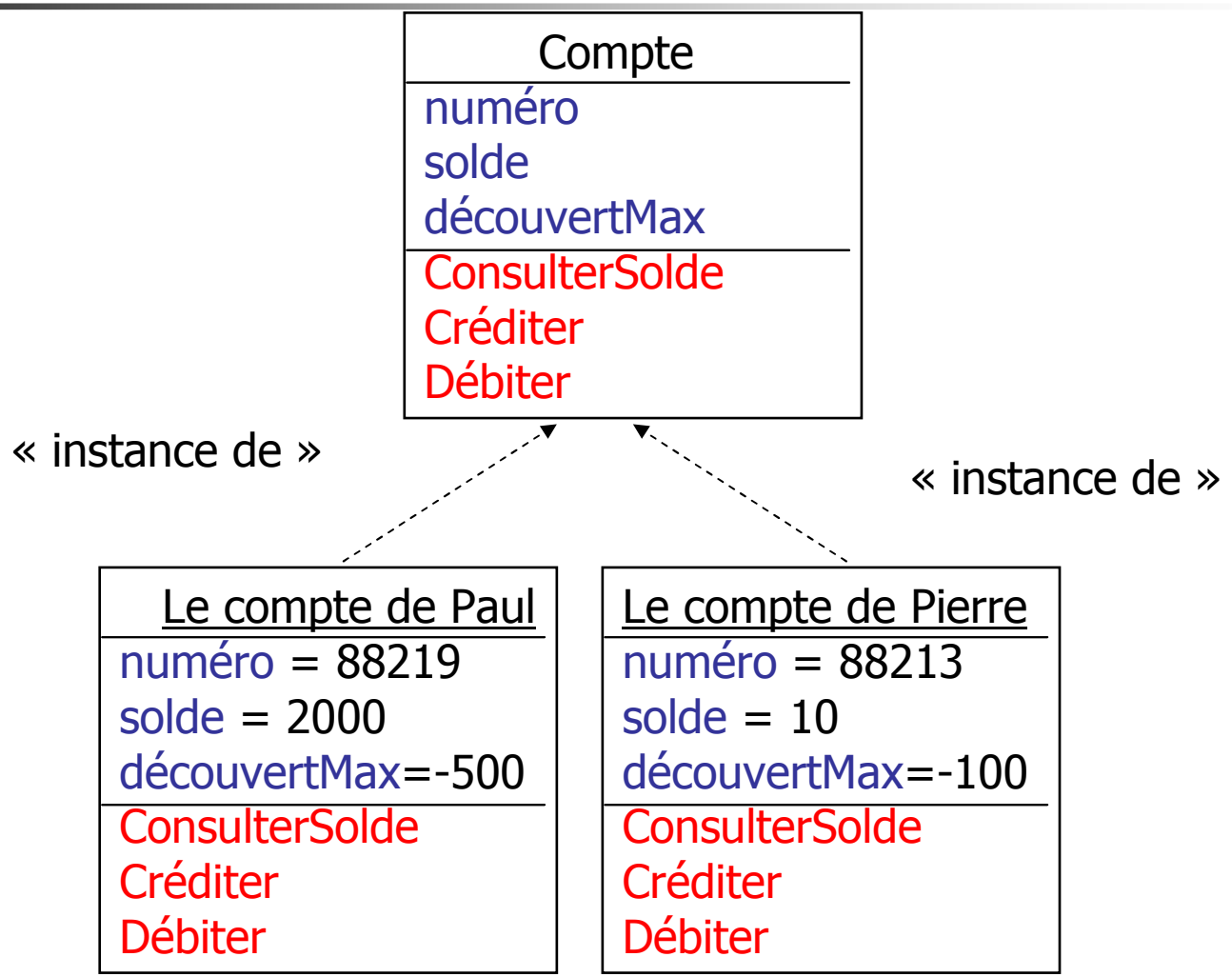




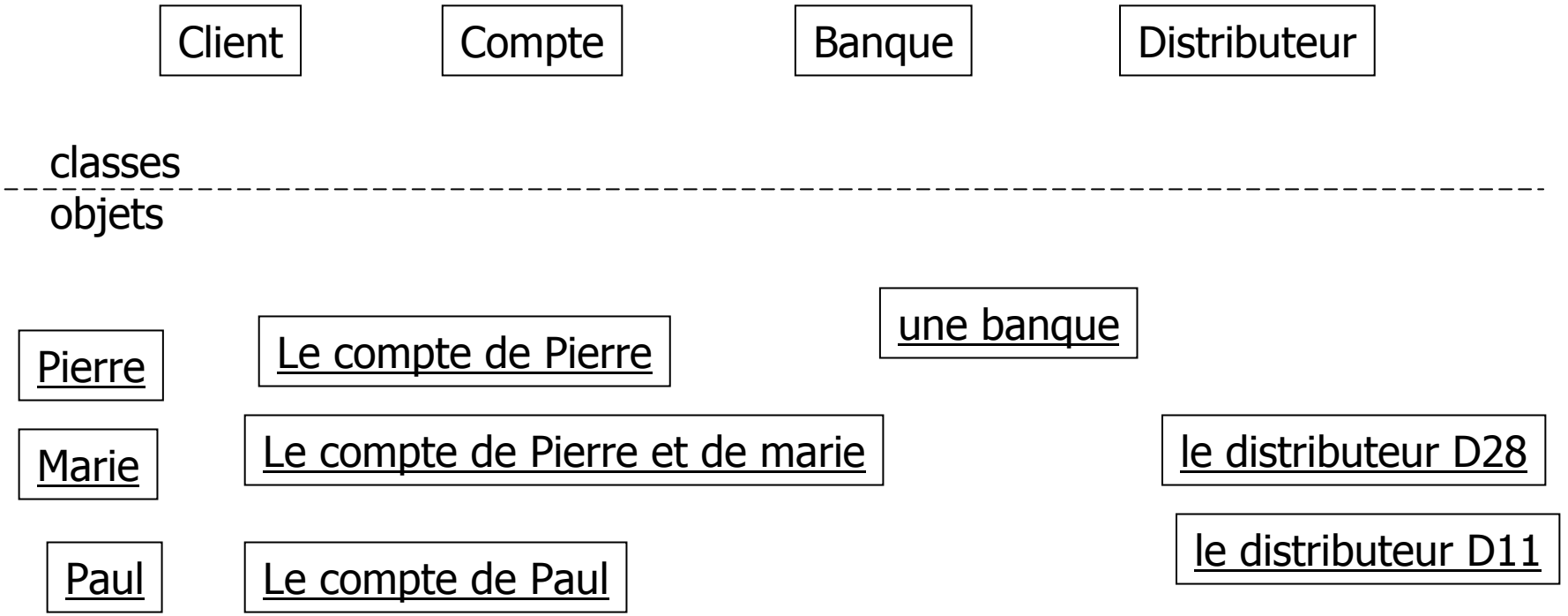
Principe de classification

- Regrouper les objets similaires en classes
- Une classe est un modèle, un moule
- Une classe est caractérisée par
  - ses attributs
  - ses méthodes
- Chaque objet est une instance d'une classe
- Une classe permet d'instancier plusieurs objets

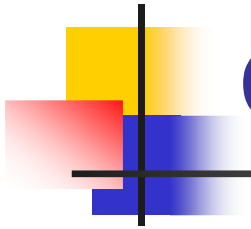
Classes vs Objets



Exemple de classification



Avantages du principe de classification

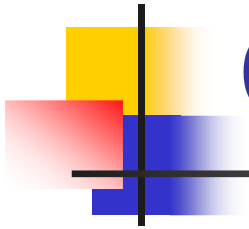


- Modèle plus simple
  - Moins de classes que d'objets
  - Les objets d'une classe sont similaires
  - L'ensemble de classe ne change pas à l'exécution
- Modèle plus général
  - Indépendant de l'état et du nombre des objets
- Modèle réutilisable
  - Une classe est réutilisable

Relations entre les classes

Description de relations au niveau du modèle

- Spécialisation/généralisation
  - Un compte courant est un cas particulier de compte
- Composition / agrégation
  - Un relevé de compte est composé de lignes
- Association
  - Tout compte est associé à au plus une banque



Concepts essentiels

- Objet
- Attribut
- Méthode
- Classe

- Réification
- Encapsulation
- Classification