



02

Chapitre

Un paradigme : conception orientée objet

1I2AC1 : Génie logiciel et Conception orientée objet

Régis Clouard, ENSICAEN - GREYC

« N'importe quel programmeur peut écrire du code que l'ordinateur comprend. Les bons programmeurs écrivent du code que les humains peuvent comprendre. »

Martin Fowler

Révolution du génie logiciel (2001)

2

- Nouvelle définition du génie logiciel

Le génie logiciel est l'art et l'ensemble des moyens techniques, industriels et humains qu'il faut réunir pour construire, distribuer et maintenir des logiciels de qualité

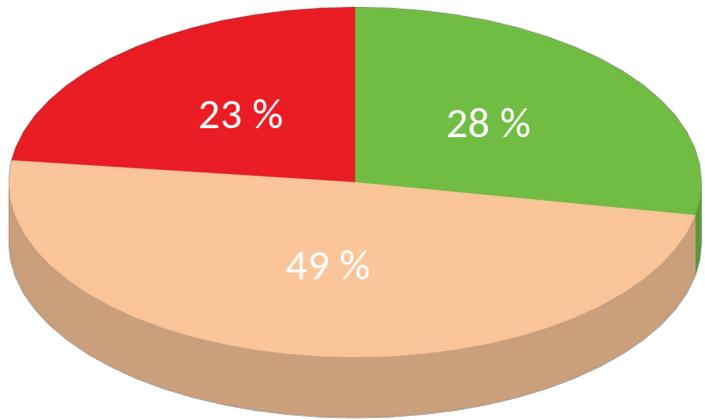
Le génie logiciel aujourd'hui

3

- Nouvelle méthode de gestion de projet
 - **Cycle itératif et incrémental**
- Nouveau paradigme de conception logicielle
 - **Conception orientée objet**
- Nouveau formalisme de modélisation
 - **UML**

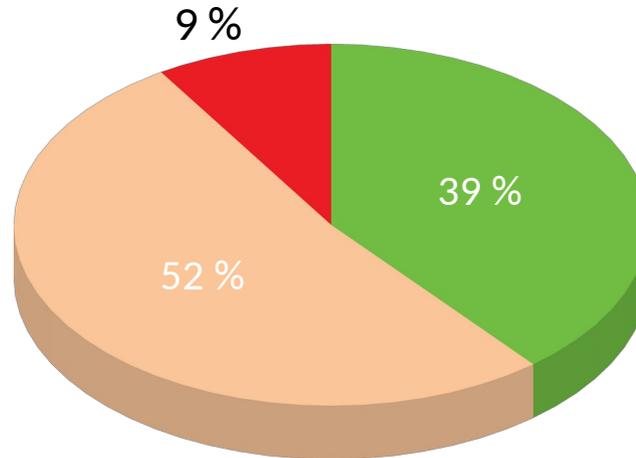
Effet sur la réussite des projets

Rappel 2000

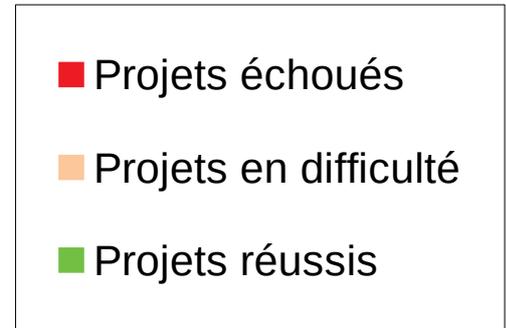


(35.000 projets étudiés)

2015



(10.000 projets étudiés)



Source : the Standish Group

Plan du chapitre

1

Le paradigme
objet

Paradigmes de programmation

- Donnez quelques paradigmes de programmation

Paradigme procédural vs Paradigme objet

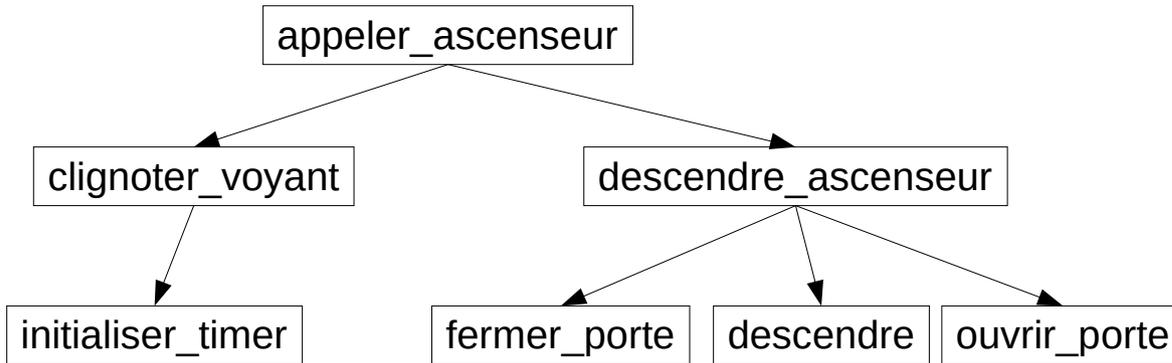
■ Deux points de vue sur le problème → Point

■ **Procédural**

- Point de vue sur les opérations
- Les données sont inertes

■ **Objet**

- Point de vue sur les données
- Les données sont animés



Grappe d'appels

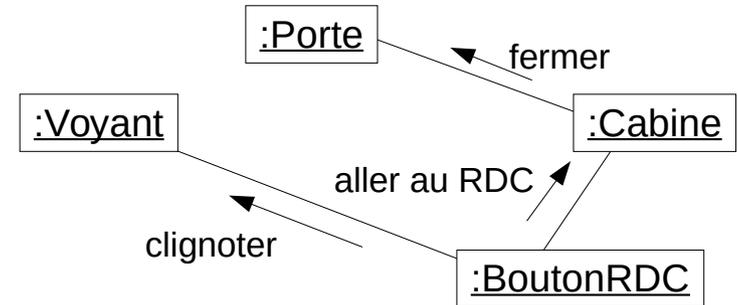


Diagramme de collaboration

Paradigme procédural vs Paradigme objet

8

Exemple

**Comptage des étudiants
présents en cours**

Conception procédurale vs Conception objet

9

■ Algorithmique

- Question à résoudre : **Que veut-on faire ?**
- Solution : la séquence d'appels de procédures

■ Modélisation

- Question à résoudre : **De quoi parle t-on ?**
- Solution : la liste des objets avec les bons services

Conception orientée objet (COO)

Conception orientée objet :

Si je disposais d'un chapeau magique, quel type de données voudrais-je voir sortir du chapeau pour m'aider à résoudre le problème ?



Conception orientée objet

11

Exemple

**Guichet automatique de billets
(GAB ou *ATM*)**

Conception procédurale vs Conception objet

■ Algorithmique

- Avantages
 - ▶ Proche de la machine
- Limites
 - ▶ Inaccessible aux clients
 - ▶ Inadaptée à la complexité des gros logici

■ Modélisation

- Avantages
 - ▶ Adapter aux gros logiciels : approche cartésienne de la conception
 - ▶ Implémentation repoussée le plus tard possible
 - ▶ Accessible aux clients
- Limite
 - ▶ Vision fractionnée du logiciel

Programmation procédurale vs Programmation orientée objet

13

- La différence ne concerne que quelques mots clés
 - ▶ 6 mots clés suffisent pour passer du C au Java :
`class, extends, implements, interface, new, public`
 - Mais ce sont deux paradigmes différents
- Conséquence :
 - Le paradigme objet ne s'apprend pas par le langage

Concepts de la conception objet

14

- La conception orientée objet s'appuie sur 5 concepts :
 - 1) Objet et principe d'encapsulation
 - 2) Classe
 - 3) Association
 - 4) Héritage
 - 5) Polymorphisme

Plan du chapitre

1
Le paradigme
objet

2
Les objets
(encapsulation)

Notion d'objet

- Physiquement
 - **Objet = structure en C incluant des pointeurs sur des procédures**
- Par exemple une voiture

```
at_013_sr: Car
color = blue
quantity= 42 l
power = 100 hp
move()
stop()
refuel()
```

```
typedef struct s_car {
    int color;
    int quantity;
    int power;

    void (*move)();
    void (*stop)();
    void (*refuel)(int);
} Car;
```

```
Car at_01_sr;

at_01_sr.power = 110;
at_01_sr.refuel(10);
```

Notion d'objet

17

■ Conceptuellement

- **Objet = propriétés + services**
- **Propriété (Attribut) : donnée membre**
 - ▶ Possédant une valeur
 - ▶ Évoluant au cours du temps
- **Service (Méthode) : procédure membre attachée à l'objet**
 - ▶ Utilisant potentiellement les données membres pour fonctionner
 - ▶ Déclenchée par appel explicite à partir de l'objet

at-013-sr: Car

color = blue
quantity= 42 l
power = 100 hp

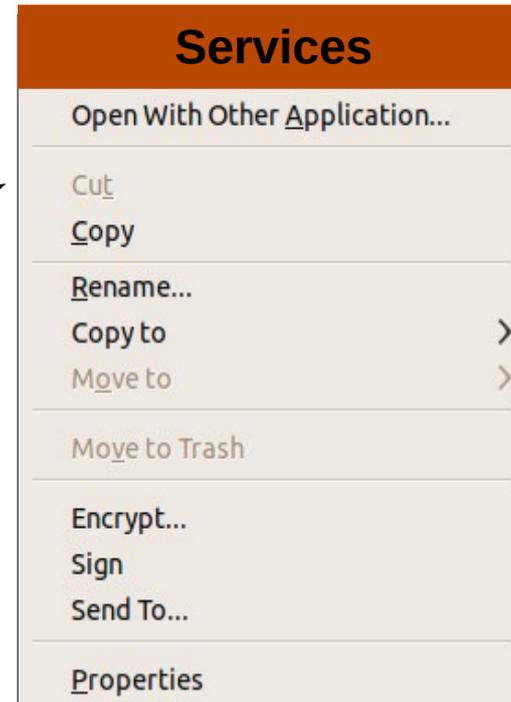
move()
stop()
refuel()

Principe d'encapsulation

- objet = fournisseur de services
≠ structure de données



CD/DVD Drive



Principe d'encapsulation

19

- Exemple : parcours d'une chaîne de caractères pour la mettre en majuscule
- En C
 - En Java

Principe d'encapsulation

- Éprouvez la différence essentielle entre :

1) `at_01_sr.color = "blue";`

2) `at_01_sr.setColor("blue");`

Principe d'encapsulation

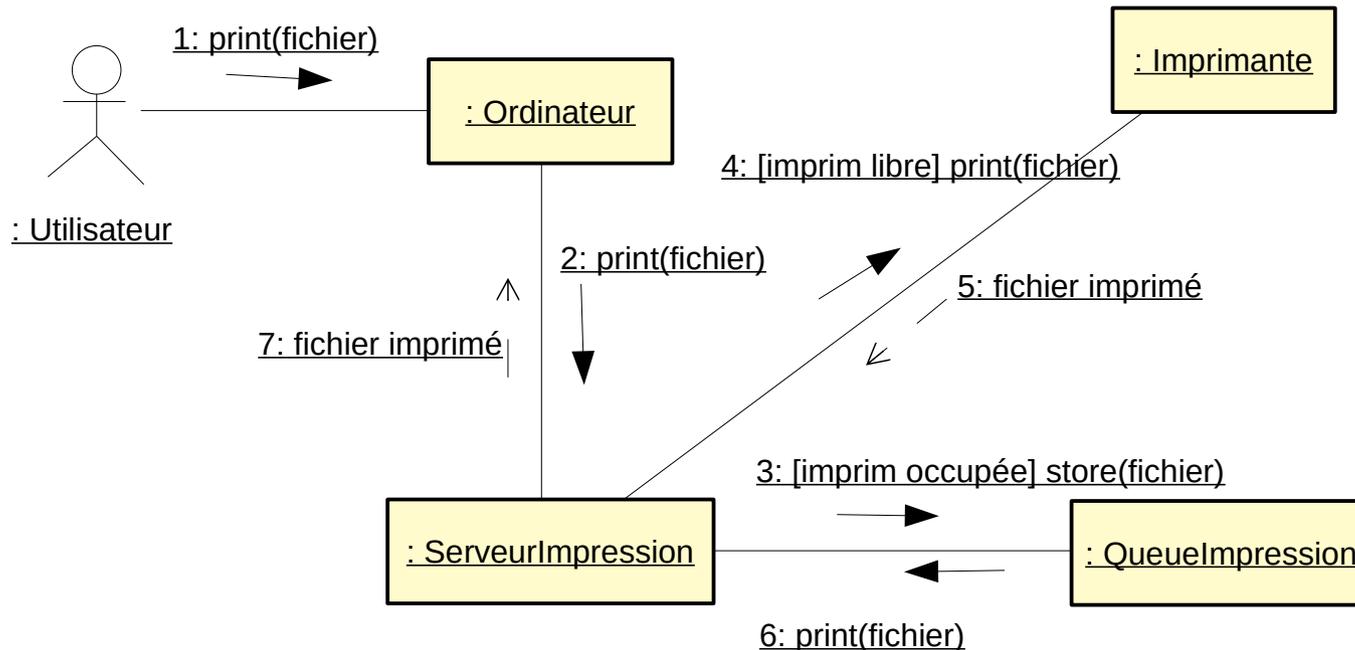
21

- Les attributs ne sont pas une préoccupation de la conception mais de la programmation
- Un attribut n'existe que parce qu'un service en a besoin

**Ne me parlez plus d'attributs
(sauf à ma demande)**

Association

- Un objet ne doit pas être omniscient mais au contraire spécialisé
 - Sinon cela revient à faire de la conception procédurale
- Il doit donc faire appel aux services d'autres objets qu'il connaît par la liste de ses associations



Plan du chapitre

1
Le paradigme
objet

2
Les objets

3
Les classes

Classe

- Représente un concept du domaine
- Génératrice d'objets
- Nom
- Casse*
- Représentation UML

Les attributs / propriétés

- Nom
- Type : primitifs ou assimilés
- Casse
- Représentation UML

Les méthodes

- Nom
- Casse
- Représentation UML

Implémentation en Java

27

Exemple

Cas de Voiture