



C. Maréchal

Définitions

- Classe
- Objet, instance
- Champs
- Méthodes

Mécanismes

- Encapsulation
- Héritage
- Polymorphisme

Questions

Programmation Orientée Objet (POO)

Terminologie

Beamer_02_POO_terminologie.tex V6241

C. Maréchal

Centre de Formation Professionnelle Technique
École d'Informatique
Genève

04/09/2015



Ce document est publié par le DIP Genève sous licence Creative Commons - utilisation et adaptation autorisées sous conditions. www.ge.ch/sem/cc/by-nc-sa

Auteur : C. Maréchal



Sommaire

C. Maréchal

Définitions

Classe
Objet, instance
Champs
Méthodes

Mécanismes

Encapsulation
Héritage
Polymorphisme

Questions

- 1 Définitions
 - Classe
 - Objet, instance
 - Champs
 - Méthodes

- 2 Mécanismes
 - Encapsulation
 - Héritage
 - Polymorphisme



Sommaire

C. Maréchal

Définitions

Classe
Objet, instance
Champs
Méthodes

Mécanismes

Encapsulation
Héritage
Polymorphisme

Questions

- 1 Définitions
 - Classe
 - Objet, instance
 - Champs
 - Méthodes

- 2 Mécanismes
 - Encapsulation
 - Héritage
 - Polymorphisme



Sommaire

C. Maréchal

Définitions

Classe
Objet, instance
Champs
Méthodes

Mécanismes

Encapsulation
Héritage
Polymorphisme

Questions

1 Définitions

- Classe
- Objet, instance
 - Champs
 - Méthodes

2 Mécanismes

- Encapsulation
- Héritage
- Polymorphisme



Définitions

Classe

C. Maréchal

Définitions

Classe

Objet, instance

Champs

Méthodes

Mécanismes

Encapsulation

Héritage

Polymorphisme

Questions

Classe

Type structuré pouvant contenir :

- des données
- du code

C'est un modèle, un moule, un tampon.





Définitions

Objet, instance

C. Maréchal

Définitions

Classe

Objet, instance

Champs

Méthodes

Mécanismes

Encapsulation

Héritage

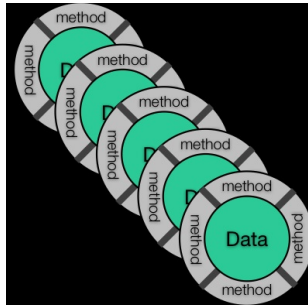
Polymorphisme

Questions

Objet, instance

Variable de type « *classe* »

C'est une trace laissée par le coup de tampon.





Définitions

Champs

C. Maréchal

Définitions

Classe

Objet, instance

Champs

Méthodes

Mécanismes

Encapsulation

Héritage

Polymorphisme

Questions

Champs

Variables locales à l'objet :

- constituent la « *mémoire* » de l'objet
- sont associées à un objet spécifique
- ne sont pas visibles depuis l'extérieur de l'objet
- sont également appelées :
 - attributs
 - propriétés
 - variables d'instance (*ivar : instance variables*)



Définitions

Méthodes

C. Maréchal

Définitions

Classe
Objet, instance
Champs
Méthodes

Mécanismes

Encapsulation
Héritage
Polymorphisme

Questions

Méthodes

Fonctions de manipulation des champs de l'objet :

- constituent le « *cerveau* » de l'objet
- sont communes à **tous** les objets d'une classe
- sont visibles depuis l'extérieur de l'objet



Sommaire

C. Maréchal

Définitions

Classe
Objet, instance
Champs
Méthodes

Mécanismes

Encapsulation
Héritage
Polymorphisme

Questions

1 Définitions

- Classe
- Objet, instance
 - Champs
 - Méthodes

2 Mécanismes

- Encapsulation
- Héritage
- Polymorphisme



Mécanismes

Encapsulation I

C. Maréchal

Définitions

Classe
Objet, instance
Champs
Méthodes

Mécanismes

Encapsulation
Héritage
Polymorphisme

Questions

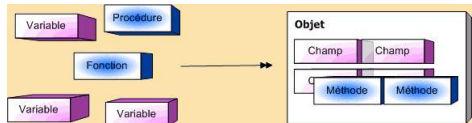
Encapsulation

Action de réunir sous la même entité :

- les données
- les moyens de les gérer

à savoir :

- les champs
- les méthodes





Mécanismes

Encapsulation II

C. Maréchal

Définitions

Classe
Objet, instance
Champs
Méthodes

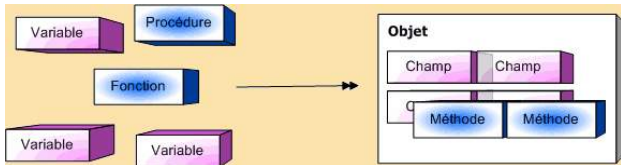
Mécanismes

Encapsulation
Héritage
Polymorphisme

Questions

L'encapsulation permet de faire voir l'objet à l'extérieur comme une « *boîte noire* » ayant :

- un certaine mémoire
- un comportement spécifié





Mécanismes

Héritage I

C. Maréchal

Définitions

Classe
Objet, instance
Champs
Méthodes

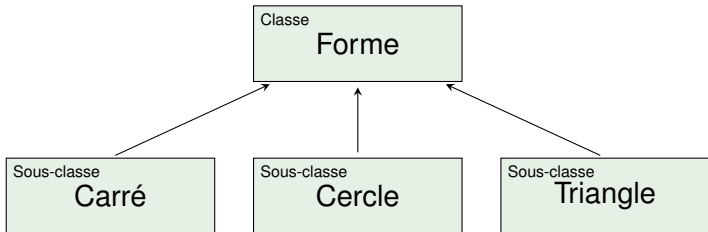
Mécanismes

Encapsulation
Héritage
Polymorphisme

Questions

Héritage

Principe permettant de créer une nouvelle classe à partir d'une classe existante



→ "est une"
"hérite de"



Mécanismes

Héritage II

C. Maréchal

Définitions

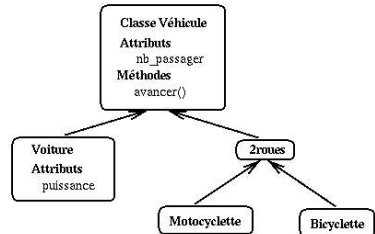
Classe
Objet, instance
Champs
Méthodes

Mécanismes

Encapsulation
Héritage
Polymorphisme

Questions

- Les objets possèdent les champs et les méthodes de leur classe, qui elle même hérite ses champs et méthodes d'une classe de niveau supérieur
- Cela définit un arbre d'héritage, appelé « *hiérarchie de classes* »



→ relation d'héritage



Mécanismes

Polymorphisme I

C. Maréchal

Définitions

Classe
Objet, instance
Champs
Méthodes

Mécanismes

Encapsulation
Héritage
Polymorphisme

Questions

Polymorphisme

Possibilité pour une entité de prendre plusieurs formes.
En programmation : caractérise une entité qui fait référence au moment de l'exécution à des occurrences de différentes classes.

En pratique, on utilise les mots :

- **surcharge** (*overloading*) pour le polymorphisme à la compilation
- **redéfinition** (*overriding*) pour le polymorphisme à l'exécution



Mécanismes

Polymorphisme I

C. Maréchal

Définitions

Classe
Objet, instance
Champs
Méthodes

Mécanismes

Encapsulation
Héritage
Polymorphisme

Questions

Polymorphisme

Possibilité pour une entité de prendre plusieurs formes.
En programmation : caractérise une entité qui fait référence au moment de l'exécution à des occurrences de différentes classes.

En pratique, on utilise les mots :

- **surcharge** (*overloading*) pour le polymorphisme à la compilation
- **redéfinition** (*overriding*) pour le polymorphisme à l'exécution



Mécanismes

Polymorphisme II : surcharge (exemple C#)

Surcharge (*overloading*)

Dans la même classe, plusieurs méthodes ont **le même nom**. Elles se différencient (à la compilation) par le nombre et/ou le type des paramètres d'entrée.

```
// C#  
  
class TA  
{  
    public int f(){...};  
    public int f(int b){...};           // overloading  
    public int f(float c){...};        // overloading  
};
```



Mécanismes

Polymorphisme III : redéfinition (exemple C#)

C. Maréchal

Définitions

- Classe
- Objet, instance
- Champs
- Méthodes

Mécanismes

- Encapsulation
- Héritage
- Polymorphisme

Questions

Redéfinition (*overriding*)

Une méthode d'une classe dérivée a **le même nom et les mêmes paramètres** que la méthode de la classe parente. Elles se différencient (à l'exécution) par l'identité de l'objet.

```
// C\#

class TA {
    public virtual int f(){return 1;}
};

class TB : TA {
    public override int f(){return 2;}    //
    overriding
};
```

```
class TA { // C#
    public virtual int f(){return 1;}
};
class TB : TA {
    public override int f(){return 2;} // overriding
};

static void Main(string[] args)
{
    TA a = new TA();
    TB b = new TB();
    TA obj;

    obj = a;
    Console.WriteLine("{0}", obj.f()); // display 1

    obj = b;
    Console.WriteLine("{0}", obj.f()); // display 2
}
```



Vos questions

C. Maréchal

Définitions

- Classe
- Objet, instance
- Champs
- Méthodes

Mécanismes

- Encapsulation
- Héritage
- Polymorphisme

Questions

