

C4 Programmation orienté objet

Programmation procédurale

Jusqu'à maintenant pour programmer nous avons utilisé une approche dite *procédurale*, c'est à dire que :

C4 Programmation orienté objet

Programmation procédurale

Jusqu'à maintenant pour programmer nous avons utilisé une approche dite *procédurale*, c'est à dire que :

- nous avons défini des variables représentant ce que l'on souhaite modéliser. Par exemple, avec le module `curse` la position d'un point sur le terminal est représenté par deux entiers (ligne et colonne).

C4 Programmation orienté objet

Programmation procédurale

Jusqu'à maintenant pour programmer nous avons utilisé une approche dite *procédurale*, c'est à dire que :

- nous avons défini des variables représentant ce que l'on souhaite modéliser. Par exemple, avec le module `curse` la position d'un point sur le terminal est représenté par deux entiers (ligne et colonne).
- nous créons ensuite des fonctions qui modifient l'état de ces variables. Par exemple, pour déplacer notre point nous créons une fonction qui va modifier la position du point.

C4 Programmation orienté objet

Programmation procédurale

Jusqu'à maintenant pour programmer nous avons utilisé une approche dite *procédurale*, c'est à dire que :

- nous avons défini des variables représentant ce que l'on souhaite modéliser. Par exemple, avec le module `curse` la position d'un point sur le terminal est représenté par deux entiers (ligne et colonne).
- nous créons ensuite des fonctions qui modifient l'état de ces variables. Par exemple, pour déplacer notre point nous créons une fonction qui va modifier la position du point.

En informatique, une façon de d'approcher un problème et d'en concevoir et modéliser une solution s'appelle un **paradigme** de programmation, et nous n'avons donc jusqu'à maintenant utilisé le **paradigme procédural**.

C4 Programmation orienté objet

Définition

- Un **objet** en informatique, est une représentation dans un langage de programmation d'un élément du monde physique (voiture, chien, film, ...), ou d'un concept (jeu, personnage, ...).

C4 Programmation orienté objet

Définition

- Un **objet** en informatique, est une représentation dans un langage de programmation d'un élément du monde physique (voiture, chien, film, ...), ou d'un concept (jeu, personnage, ...).
- Un **attribut** est une caractéristique de l'objet, par exemple un objet livre pourrait avoir comme attribut son auteur, son nombre de pages, ...

C4 Programmation orienté objet

Définition

- Un **objet** en informatique, est une représentation dans un langage de programmation d'un élément du monde physique (voiture, chien, film, ...), ou d'un concept (jeu, personnage, ...).
- Un **attribut** est une caractéristique de l'objet, par exemple un objet livre pourrait avoir comme attribut son auteur, son nombre de pages, ...
- Une **méthode** est une fonction permettant d'interagir avec l'objet et de définir son comportement. Par exemple, un objet personnage d'un jeu vidéo peut être doté d'une méthode permettant de le déplacer.

C4 Programmation orienté objet

Définition

- Un **objet** en informatique, est une représentation dans un langage de programmation d'un élément du monde physique (voiture, chien, film, ...), ou d'un concept (jeu, personnage, ...).
- Un **attribut** est une caractéristique de l'objet, par exemple un objet livre pourrait avoir comme attribut son auteur, son nombre de pages, ...
- Une **méthode** est une fonction permettant d'interagir avec l'objet et de définir son comportement. Par exemple, un objet personnage d'un jeu vidéo peut être doté d'une méthode permettant de le déplacer.
- Une **classe**, déclare l'ensemble des attributs et méthodes communes à des objets. On dit donc parfois qu'un objet est l'**instanciation** d'une classe. Par exemple, un livre (*Les misérables*, Victor Hugo) est une instanciation de la classe des livres (qui indique qu'un livre a un auteur et un titre).

C4 Programmation orienté objet

Définition

- Un **objet** en informatique, est une représentation dans un langage de programmation d'un élément du monde physique (voiture, chien, film, ...), ou d'un concept (jeu, personnage, ...).
- Un **attribut** est une caractéristique de l'objet, par exemple un objet livre pourrait avoir comme attribut son auteur, son nombre de pages, ...
- Une **méthode** est une fonction permettant d'interagir avec l'objet et de définir son comportement. Par exemple, un objet personnage d'un jeu vidéo peut être doté d'une méthode permettant de le déplacer.
- Une **classe**, déclare l'ensemble des attributs et méthodes communes à des objets. On dit donc parfois qu'un objet est l'**instanciation** d'une classe. Par exemple, un livre (*Les misérables*, Victor Hugo) est une instanciation de la classe des livres (qui indique qu'un livre a un auteur et un titre).
- En **programmation orienté objet**, le concepteur du logiciel centre sa réflexion sur les objets qu'il manipule et sur la façon dont ces objets communiquent et interagissent. Il s'agit donc d'un nouveau **paradigme de programmation**

C4 Programmation orienté objet

Visualisation et exemples

Objets « réels »

C4 Programmation orienté objet

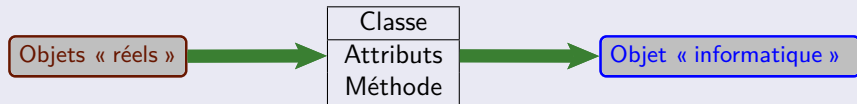
Visualisation et exemples



délisation

C4 Programmation orienté objet

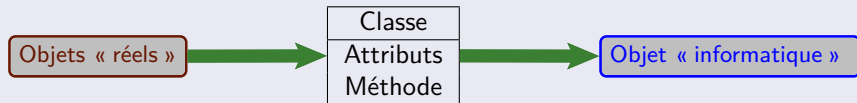
Visualisation et exemples



Exemples :

C4 Programmation orienté objet

Visualisation et exemples

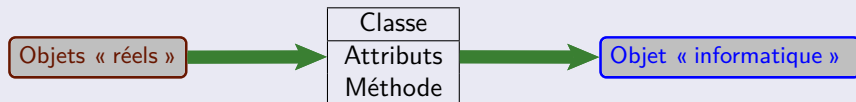


Exemples :

- une voiture (objet réel) pourrait être modélisé par des attributs marque, modèle, immatriculation, et des méthodes comme faire le plein, réparer.

C4 Programmation orienté objet

Visualisation et exemples

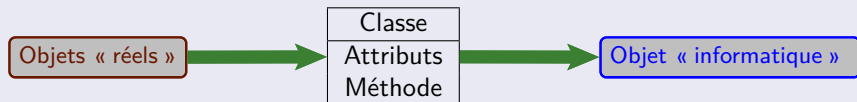


Exemples :

- une voiture (objet réel) pourrait être modélisé par des attributs marque, modèle, immatriculation, et des méthodes comme faire le plein, réparer.
- un point (objet mathématique) pourrait être modélisé par des attributs comme abscisse, ordonnée et nom et des méthodes comme déplacer.

C4 Programmation orienté objet

Visualisation et exemples



Exemples :

- une voiture (objet réel) pourrait être modélisé par des attributs marque, modèle, immatriculation, et des méthodes comme faire le plein, réparer.
- un point (objet mathématique) pourrait être modélisé par des attributs comme abscisse, ordonnée et nom et des méthodes comme déplacer.
- un personnage (objet virtuel) d'un jeu vidéo pourrait être modélisé par des attributs comme race, points de vie, arme, ... et des méthodes comme se battre, parler, ...

C4 Programmation orienté objet

Exemple

- Le module `turtle` de Python est orienté objet, ainsi, les tortues sont pensées comme des objets pouvant dessiner sur l'écran, qui est lui-même un objet (possédant par exemple une méthode pouvant modifier sa couleur).

C4 Programmation orienté objet

Exemple

- Le module `turtle` de Python est orienté objet, ainsi, les tortues sont pensées comme des objets pouvant dessiner sur l'écran, qui est lui-même un objet (possédant par exemple une méthode pouvant modifier sa couleur).
- Considérons le fragment de code suivant :

```
1  ecran = turtle.Screen()  
2  ecran.bgcolor("brown")  
3  caroline=turtle.Turtle()  
4  caroline.forward(100)
```

Repérer les objets et qui sont créés, quelle sont les méthodes appelées et avec quelle syntaxe ?

C4 Programmation orienté objet

Exemple

- Le module `turtle` de Python est orienté objet, ainsi, les tortues sont pensées comme des objets pouvant dessiner sur l'écran, qui est lui-même un objet (possédant par exemple une méthode pouvant modifier sa couleur).
- Considérons le fragment de code suivant :

```
1  ecran = turtle.Screen()  
2  ecran.bgcolor("brown")  
3  caroline=turtle.Turtle()  
4  caroline.forward(100)
```

Repérer les objets et qui sont créés, quelle sont les méthodes appelées et avec quelle syntaxe ? On crée un objet `ecran` de la classe `Screen` et un objet `caroline` de la classe `tortue`.

C4 Programmation orienté objet

Exemple

- Le module `turtle` de Python est orienté objet, ainsi, les tortues sont pensées comme des objets pouvant dessiner sur l'écran, qui est lui-même un objet (possédant par exemple une méthode pouvant modifier sa couleur).
- Considérons le fragment de code suivant :

```
1  ecran = turtle.Screen()  
2  ecran.bgcolor("brown")  
3  caroline=turtle.Turtle()  
4  caroline.forward(100)
```

Repérer les objets et qui sont créés, quelle sont les méthodes appelées et avec quelle syntaxe ? On crée un objet `ecran` de la classe `Screen` et un objet `caroline` de la classe `tortue`.

On appelle la méthode `bgcolor` sur l'objet `ecran` et la méthode `forward` sur l'objet `caroline`. La syntaxe est à chaque fois :

<objet>.<méthode>(…)

C4 Programmation orienté objet

Syntaxe de Python

C4 Programmation orienté objet

Syntaxe de Python

- En Python, la définition d'une classe commence par le mot clé `class` suivi du nom de la classe (par convention le nom d'une classe commence par une majuscule) et du caractère `:`

C4 Programmation orienté objet

Syntaxe de Python

- En Python, la définition d'une classe commence par le mot clé `class` suivi du nom de la classe (par convention le nom d'une classe commence par une majuscule) et du caractère `:`
- Une méthode spéciale appelée `constructeur` permet d'instancier la classe pour créer un objet, il s'agit de la méthode `__init__` de plus en Python les méthodes spéciales sont encadrés par des doubles soulignés `__`

C4 Programmation orienté objet

Syntaxe de Python

- En Python, la définition d'une classe commence par le mot clé `class` suivi du nom de la classe (par convention le nom d'une classe commence par une majuscule) et du caractère `:`
- Une méthode spéciale appelée `constructeur` permet d'instancier la classe pour créer un objet, il s'agit de la méthode `__init__` de plus en Python les méthodes spéciales sont encadrés par des doubles soulignés `__`
- Le mot clé `self` permet de faire référence à l'objet.

C4 Programmation orienté objet

Exemple

Il existe de nombreux type de dés à jouer, qui possèdent en commun d'avoir un nombre de faces bien définis et de pouvoir être lancé afin d'obtenir un résultat. On peut donc représenter un dé par un objet informatique

```
1 class De :  
2     def __init__(nombre_faces) :  
3         self.faces=nombre_faces
```