

Fiche de Révision POO

VAN DE MERGHEL Robin

Semestre 4

Table des matières

Introduction	1
Programmation et Paradigmes	1
Des exemples de Paradigmes	2
Fonctionnel	2
Impératif	2
Orienté Objet	3
Conclusion sur les exemples	4
Java, une brève introduction	4
Rapides spécifications	4

Introduction

Cette fiche de révision est maintenue au fur et à mesure du semestre, pour permettre de rester à jour sur le cours. Il peut y avoir de gaffes ! N'hésitez pas à me le signaler sur *Discord* : **UnSavantFou#2534**.

De plus, n'hésitez pas à me DM si jamais il y a des thématiques où j'ai mal formulé / que vous ne comprenez pas !

Programmation et Paradigmes

Chaque langage de programmation a une façon de modéliser et de résoudre un problème.

Par exemple un langage fonctionnel, comme **Prolog**, va modéliser un problème en utilisant des fonctions, alors qu'un langage orienté objet, comme **Java**, va modéliser un problème en utilisant des objets.

C'est **ça**, le paradigme : **la façon de modéliser un problème**.

Des exemples de Paradigmes

Pour de la culture générale (mais il est très conseillé de le savoir), voici quelques exemples de paradigmes :

À chaque fois, on va proposer une modélisation d'un problème simple : un chat qui mange de la nourriture.

Notez que le code n'est pas important, je veux juste montrer qu'on peut facilement modéliser le même problème dans chaque langage.

Fonctionnel

On n'utilise pas de variables, mais des fonctions. On ne modifie pas les données, mais on les transforme. On ne fait pas de boucles, mais on utilise des fonctions récursives.

Exemple : Prolog

```
% On définit un chat
chat(miaou, 5, 10).

% On définit une nourriture
nourriture(poisson, 5).

% On définit une fonction qui dit si un chat peut manger une
nourriture
peutManger(Chat, Nourriture) :-
    chat(_, _, PoidsChat),
    nourriture(_, PoidsNourriture),
    PoidsChat > PoidsNourriture.
```

Impératif

On utilise des variables, on modifie les données, on utilise des boucles.

Exemple : python

```
# On définit un chat
chat = {
    "nom": "miaou",
    "age": 5,
    "poids": 10
}

# On définit une nourriture
nourriture = {
    "nom": "poisson",
    "poids": 5
}
```

```

}

# On définit une fonction qui dit si un chat peut manger une
nourriture
def peutManger(chat, nourriture):
    if chat["poids"] > nourriture["poids"]:
        return True
    else:
        return False

```

Orienté Objet

On utilise des objets, qui sont des variables qui contiennent des fonctions. On modifie les données, on utilise des boucles.

Exemple : Java

```

// On définit un chat
class Chat {
    String nom;
    int age;
    int poids;

    public Chat(String nom, int age, int poids) {
        this.nom = nom;
        this.age = age;
        this.poids = poids;
    }
}

// On définit une nourriture
class Nourriture {
    String nom;
    int poids;

    public Nourriture(String nom, int poids) {
        this.nom = nom;
        this.poids = poids;
    }
}

// On définit une fonction qui dit si un chat peut manger une
nourriture
public boolean peutManger(Chat chat, Nourriture nourriture) {
    if (chat.poids > nourriture.poids) {
        return true;
    } else {

```

```
        return false;
    }
}
```

Conclusion sur les exemples

On a à chaque fois le même problème : un chat qui mange de la nourriture. Mais on a utilisé des paradigmes différents. Si je reformule, des façons différentes de modéliser le problème.

Chaque paradigme a un avantage (pour ça qu'il existe).

- Le fonctionnel par exemple est très bon pour pouvoir trouver des bugs, on a une équation mathématique, on peut donc prouver facilement qu'un programme est correct.
- L'impératif est très proche de la machine, et très naturel pour les humains.
- L'orienté objet est très bon pour la modélisation de problèmes complexes, on peut diviser le problème en plusieurs objets.

Java, une brève introduction

Java est un langage créé à la base pour pouvoir faire de la programmation orientée objet.

Rapides spécifications

- Un langage de programmation orienté objet.
- Un langage compilé.
 - Il est d'abord précompilé, puis compilé.
- Il est exécuté sur une machine virtuelle.
 - C'est une machine virtuelle qui est installée sur votre ordinateur, et qui va exécuter le code compilé.
 - Cela permet d'être utilisable sur téléphones, tablettes, ordinateurs, etc.
- Il vient avec le JRE (Java Runtime Environment).
 - C'est un ensemble de bibliothèques qui permettent de faire des choses plus complexes.
 - C'est un peu comme le `stdlib` de C.