

R201 — Développement orienté objets

Chapitre 0 — Introduction

Clément PAGÈS
`clement.pages@u-pec.fr`

Université Paris-Est Créteil
BUT Informatique 1^{re} année

2025-2026

Sommaire

- 1 Le paradigme objet
- 2 Fondamentaux de syntaxe en Java

Programmation impérative et programmation objet

Deux approches antagonistes

Le paradigme impératif

- Séparation données/procédures.
- Les fonctions *manipulent* des données.
- On raisonne en termes de fonctions à réaliser au sein du programme.

Programmation impérative et programmation objet

Deux approches antagonistes

Le paradigme impératif

- Séparation données/procédures.
- Les fonctions *manipulent* des données.
- On raisonne en termes de fonctions à réaliser au sein du programme.

Cette approche, bien que paraissant naturelle... est en fait très étrange! On ne fait pas comme ça dans la vie!

Le paradigme « objet »

- Notre programme est constitué d'entités, appelées *objets*, qui interagissent.
- Les objets sont caractérisés par leur nom, et des données qui les représentent, appelées *attributs*.
- Les interactions entre objets modifient la valeurs de leurs attributs.

Programmation impérative et programmation objet

L'approche objet

Un approche plus structurelle

- La programmation objet focalise sur *les données* manipulées.
- Les interactions entre données sont *secondaires*
- Et c'est logique : **les données sont plus stables que leurs interactions!**

Programmation impérative et programmation objet

L'approche objet

Un approche plus structurelle

- La programmation objet focalise sur *les données* manipulées.
 - Les interactions entre données sont *secondaires*
 - Et c'est logique : **les données sont plus stables que leurs interactions!**
-
- Plus de séparation entre les données (les variables en PI) et leur traitement (les fonctions en PI).
 - Code beaucoup plus modulaire et facile à maintenir...
 - ... mais souvent plus abstrait!
 - En particulier, la lecture d'un code n'est plus séquentielle.

Bref historique et langages objet

Programmation par objets

- Naissance des premiers concepts : 1970, langage [Smalltalk](#).
- Formalisation théorique : années 1980 - 1990.
- Ensuite, essor très rapide.

Bref historique et langages objet

Programmation par objets

- Naissance des premiers concepts : 1970, langage **Smalltalk**.
- Formalisation théorique : années 1980 - 1990.
- Ensuite, essor très rapide.

Les langages objet

- Langages avec objets *facultatifs* : **Python**, PHP, Javascript, etc.
- Langages intrinsèquement objet : C++, **Java**, etc.

Bref historique et langages objet

Programmation par objets

- Naissance des premiers concepts : 1970, langage **Smalltalk**.
- Formalisation théorique : années 1980 - 1990.
- Ensuite, essor très rapide.

Les langages objet

- Langages avec objets *facultatifs* : **Python**, PHP, Javascript, etc.
- Langages intrinsèquement objet : C++, **Java**, etc.
- Les concepts objets ne sont pas tous présents avec la même rigueur dans tous les langages (Python est un langage *mollement* objet).

Le langage Java

- Première version : 1995.
- Propriété de Oracle¹.
- Syntaxe héritée du langage C
 - Mêmes structures conditionnelles (**if else if else, switch**)
 - Même structures itératives (**for, while**, etc.)
 - Accolades pour délimiter les blocs
- Fonctionne dans une **machine virtuelle**, appelée JVM (*Java virtual machine*).
- Haut niveau, typage fort, statique.

1. Rien n'est parfait dans la vie...

Le langage Java

- Première version : 1995.
- Propriété de Oracle¹.
- Syntaxe héritée du langage C
 - Mêmes structures conditionnelles (**if else if else, switch**)
 - Mêmes structures itératives (**for, while**, etc.)
 - Accolades pour délimiter les blocs
- Fonctionne dans une **machine virtuelle**, appelée JVM (*Java virtual machine*).
- Haut niveau, typage fort, statique.

Environnement de développement

- En général, on développe en Java via un IDE.
- La plupart sont très lourds.
- Nous allons travailler avec **Apache Netbeans**.

1. Rien n'est parfait dans la vie...

Sommaire

- 1 Le paradigme objet
- 2 Fondamentaux de syntaxe en Java

Hello world!

```
public class HelloWorld {  
    public static void main(String[] args){  
        System.out.println("Hello World!") ;  
    }  
}
```

Main.java

- Le code Java s'écrit dans des fichier .java.
- Java est un langage **compilé**. Après compilation, on obtient un fichier .class

```
$ ls  
Main.java  
$ javac Main.java  
$ ls  
Main.java Main.class  
$ java Main  
Hello World!
```

Terminal

Types natifs de variables et opérations élémentaires

- Peu de types natifs :
`boolean byte short int long char float double.`
- Tout le reste est objet (`String`, `int[]`, etc.)!
- Syntaxe comme en C.

Types natifs de variables et opérations élémentaires

- Peu de types natifs :
boolean byte short int long char float double.
- Tout le reste est objet (String, **int**[], etc.)!
- Syntaxe comme en C.

```
public class Main {  
    public static void main(String[] args){  
        int[] tab = new int[10]; // Instance tableau de 10 int  
  
        // Initialisation du tableau  
        for(int i = 0 ; i< tab.length ; i++) {  
            if (i%2 == 0) tab[i] = 2*i;  
            else          tab[i] = i;  
        }  
        // Affichage des valeurs (il y a une meilleure solution...  
        //                                     c'est pour plus tard !)  
        for(int i = 0 ; i < tab.length ; i++)  
            System.out.println("tab["+i+"] = " + tab[i]);  
    }  
}
```

Main.java