

# Introduction

William Hergès  
Sorbonne Université  
william@herges.fr

## Table des matières

1. Introduction .....	2
1.1. Programmation orientée objet .....	2
1.2. Java .....	2
2. Premier programme .....	3
3. Concept de base de l'OOP .....	4
4. Syntaxe de Java .....	5

# **1. Introduction**

Partiel le lundi 9 mars. TP solo durant la séance TP9. Mini-projet, rendu & soutenance lors du dernier TP. On a le droit à une feuille A4 écrite à la main lors des examens.

## **1.1. Programmation orientée objet**

Cherche à développer des systèmes complexes. Divise le système en plusieurs objets tout en sécurisant les données sensibles.

Permet de travailler à plusieurs : toujours penser son programme pour les autres, donc on sécurise, on simplifie et on compartimente.

## **1.2. Java**

Java est une application classique de l'OOP. Langage moderne, robuste, sûr et fortement typé.

Propose une architecture dynamique avec un compilateur pour créer des applications performantes. Repose sur la JVM pour exécuter le code compilé : permet de créer des applications indépendantes de la machine physique.

## 2. Premier programme

En Java, tout le code doit être écrit dans une classe.

```
// contenu de MonPremierProgramme.java
public class MonPremierProgramme {
    // code
}
```

Un fichier `.java` contient forcément un fichier. Il doit avoir le même nom de la classe.

Un programme est un ensemble de classes. Le point d'entrée est la méthode `main`.

La signature de la fonction `main` est :

```
public class MonPremierProgramme {
    public static void main(String[] args) {
        System.out.println("Hello world :D");
    }
}
```

Besoin du JDK pour développer en Java. Pour compiler, on écrit dans le terminal :

```
javac MonPremierProgramme.java
```

Cela crée un fichier `MonPremierProgramme.class` qui contient du bytecode JVM. Pour l'exécuter, on écrit

```
java MonPremierProgramme
```

Pour compiler plusieurs classes, on fait :

```
javac *.java
```

### **3. Concept de base de l'OOP**

En OOP, tout est encapsulé dans un objet. Un objet est autonome : il est réutilisable partout. Il est aussi sécurisé : il garantit le bon usage. Il intègre des données et des méthodes pour le manipuler proprement.

## **4. Syntaxe de Java**

...

Les objets sont représentés par deux valeurs en mémoire : le type et ses attributs.