

Conception Orientée Objet & Programmation JAVA

Chapitre 1: Introduction



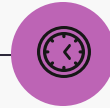
Présentation générale du module



UP Java



COOPJ



42H

3A, 3B



APP

Approche par Projet



Objectifs du chapitre



- Comprendre le paradigme OO
- Découvrir Java
- Décrire la syntaxe du langage Java
- Identifier les types des variables
- Créer une application Java

{ ..

P00 vs PP

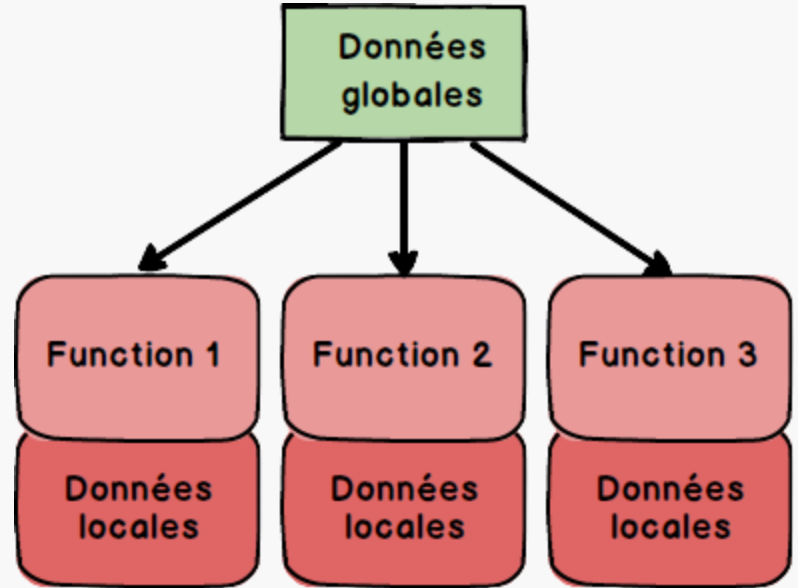


} ..

Programmation procédurale

La programmation classique telle qu'étudiée à travers les langages **C**, **Pascal**, ... définit un programme comme étant:

- Un ensemble de données sur lesquelles agissent des procédures et des fonctions.
- Les données sont **la partie passive** du programme. Les procédures et les fonctions sont **la partie active**.

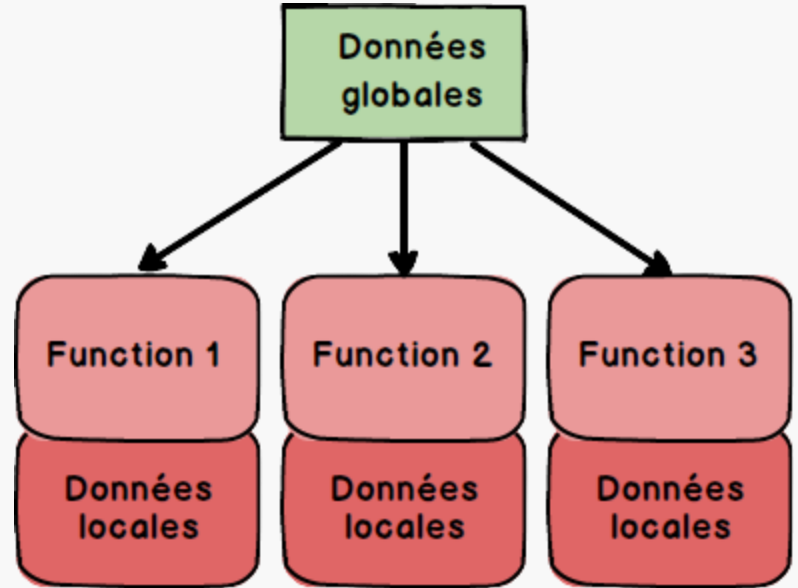


"Que doit faire mon programme ? "

Programmation procédurale

Programmer dans ce cas consiste à :

- Définir un certain nombre de variables (structures, tableaux, etc.).
- Écrire des procédures pour les manipuler sans les associer explicitement les unes aux autres.

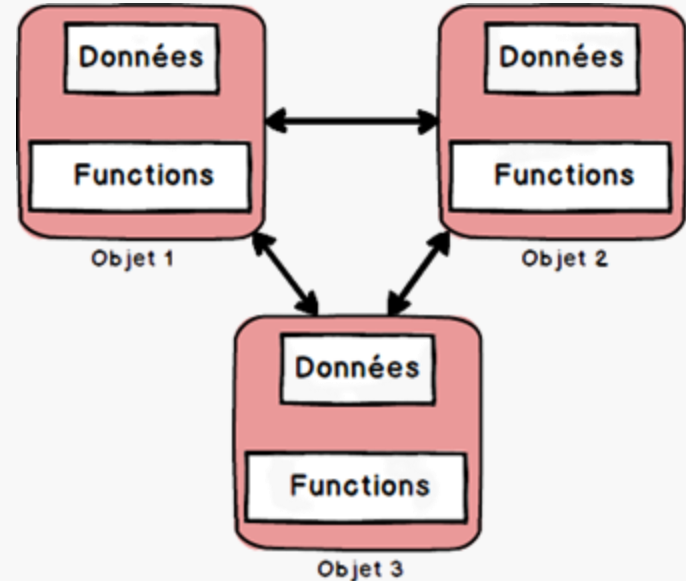


"Que doit faire mon programme ? "

P.O.O (Programmation Orientée Objet)

La programmation orientée objet est basée sur une approche de conception et de développement de logiciels

Représenter les éléments du monde réel par des entités informatiques appelées **"objets"** en adoptant un haut niveau d'abstraction

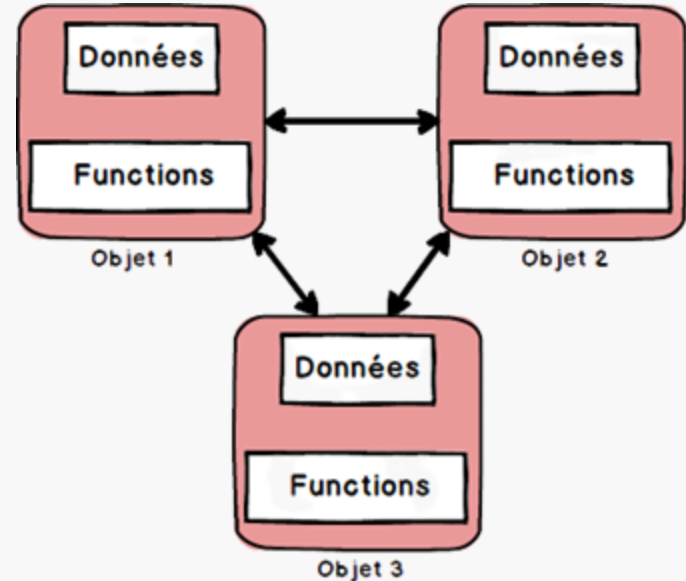


"De quoi doit être composé mon programme ?"

P.O.O (Programmation Orientée Objet)

Un objet est une entité logicielle:

- Ayant une **identité**
- Capable de sauvegarder **un état**, c'est-à-dire un ensemble d'information dans des variables internes (**attributs***).
- Répondant à des messages précis en déclenchant des activations internes appropriés qui changent l'état de l'objet (**comportement****).



"De quoi doit être composé mon programme ?"

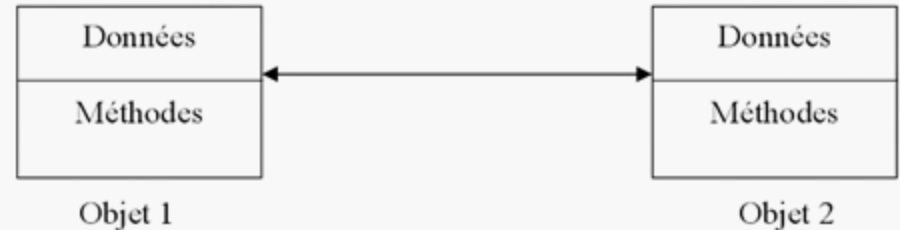
P.O.O (Programmation Orientée Objet)

*Les attributs d'un objet sont l'ensemble des informations se présentant sous forme de variable et permettant de représenter l'état de l'objet.

Ces opérations sont appelées **méthodes. Ce sont des fonctions liées à des objets et qui précisent le comportement de ces objets.

Un objet peut recevoir un message qui déclenche:

- Une méthode qui modifie son état ou
- Une méthode qui envoie un msg à un autre objet



Langages Procédurales & Orientée Objet

{

Langages procéduraux	Langages Orientée Objet
Pascal	PHP \geq 5
C	C++
PHP $<$ 5	Java

}

{ ..

C'est quoi JAVA?



} ..

Java le langage

Java est un langage de programmation orienté objet, ce qui signifie qu'il permet de créer des programmes en utilisant des objets et des classes.

Java est un langage **compilé** et **interprété**.



Compilation



Interprétation

Java le langage

- Java est conçu pour être sûr et fiable, avec des fonctionnalités de sécurité intégrées pour protéger contre les erreurs de programmation et les attaques externes.
- Java est multidisciplinaire:



Java SE



Java ME



Java EE



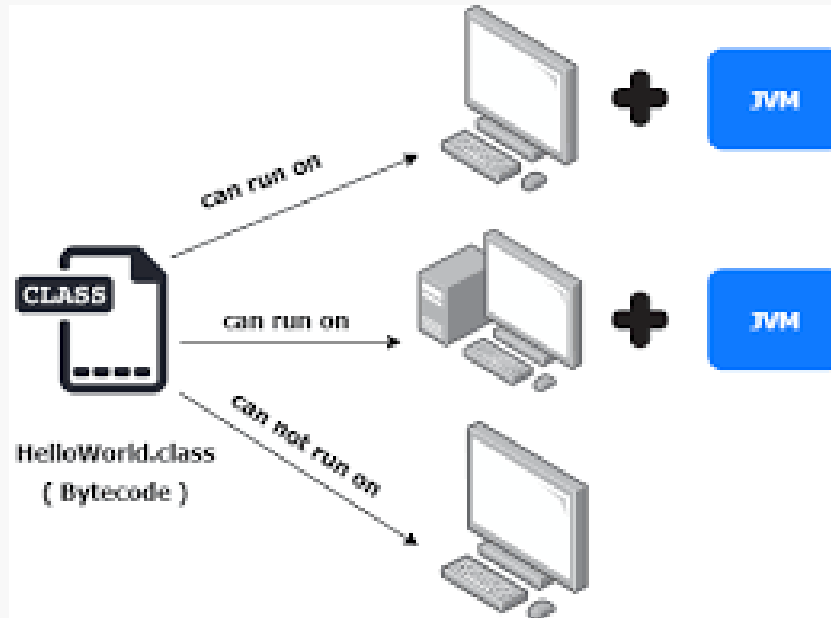
Java
Embedded



Java
Card

Java le langage

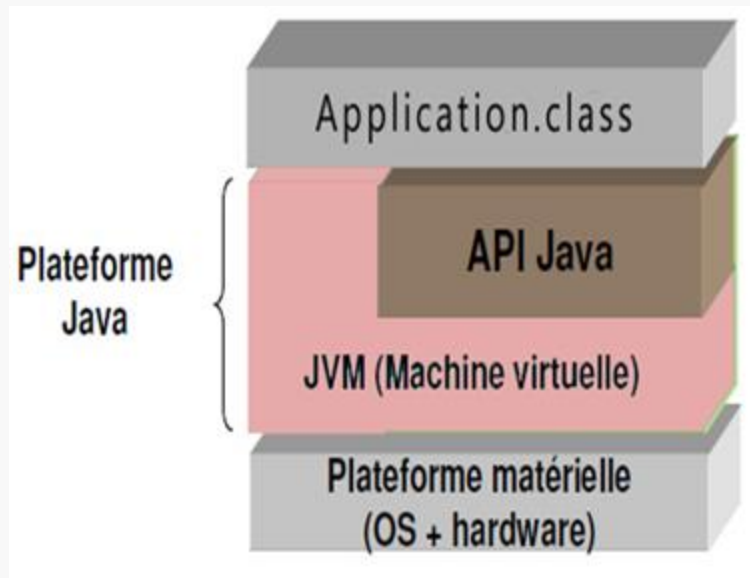
- Java est portable, il est indépendant de toute plate-forme

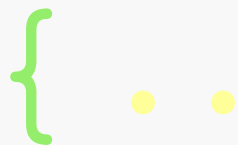


Java la plateforme

La plateforme Java est constituée de plusieurs éléments qui travaillent ensemble pour permettre l'exécution de programmes Java sur différentes plateformes. Les principaux éléments de la plateforme Java sont :

- Les bibliothèques de classes (API) :
un ensemble de classes prédéfinies qui fournissent des fonctionnalités de base pour les programmes Java
- La JVM (Java Virtual Machine) :
Une machine virtuelle est un ordinateur fictif s'exécutant sur un ordinateur réel qui traduit le bytecode vers le langage natif de l'ordinateur





Les notions fondamentales



Classe et Objet

En Java, une **classe** est **un modèle** pour créer **des objets** qui ont des propriétés (attributs) et des comportements (méthodes) similaires.

Une classe est le moule qui nous permettra de créer des objets à son image, CAD qu'un **objet** est issu d'une classe.

Classe et Objet

Partie statique

les données (*Propriétés*)

LES ATTRIBUTS :

Les attributs représentent la description des données propres à chaque classe d'objets

Partie dynamique

le code les manipulant
(*Méthodes*)

LES METHODES :

Les méthodes représentent l'ensemble des actions, procédures, fonctions ou opérations que l'on peut associer à une classe

Classe

Variables

En Java, une **variable** est un **espace mémoire** qui peut stocker une **valeur** ou une **référence à un objet**.

Pour déclarer une variable en Java, vous devez spécifier son type, suivi de son nom, comme ceci:

```
int age; //ici "int" est le type de variable et "age" est son nom
float price;
char grade;
```

Il est également possible d'initialiser une variable lors de sa déclaration en lui donnant une valeur:

```
int age = 25; //ici la variable "age" est de type "int" et crée avec la valeur 25
float price = 19.99;
char grade = 'A';
String name = "John";
```

Les types de donnée en java

En java, il existe 2 catégories de types de données :

- **Primitifs** : Ces types de données contiennent la valeur réelle de la variable en mémoire.
- **Référence** : Contient l'adresse mémoire où l'information relative à l'objet est réellement stockée

Les types primitifs

- **Valeur logique**

- boolean (true/false)

- **Nombres entiers**

- byte (1 octet)
- short (2 octets)
- int (4 octets)
- long (8 octets)

- **Nombres non entiers (à virgule flottante)**

- float (4 octets)
- double (8 octets)

- **Caractère** (un seul)

- char (2 octets)

Primitive type	int i = 1	i	1
----------------	-----------	---	---

La valeur est stockée
directement dans la
case mémoire

Les types primitifs

Type	Bits	Bytes	Minimum Range	Maximum Range
byte	8	1	-2^7	2^7-1
short	16	2	-2^{15}	$2^{15}-1$
int	32	4	-2^{31}	$2^{31}-1$
long	64	8	-2^{63}	$2^{63}-1$
float	32	4	n/a	n/a
double	64	8	n/a	n/a

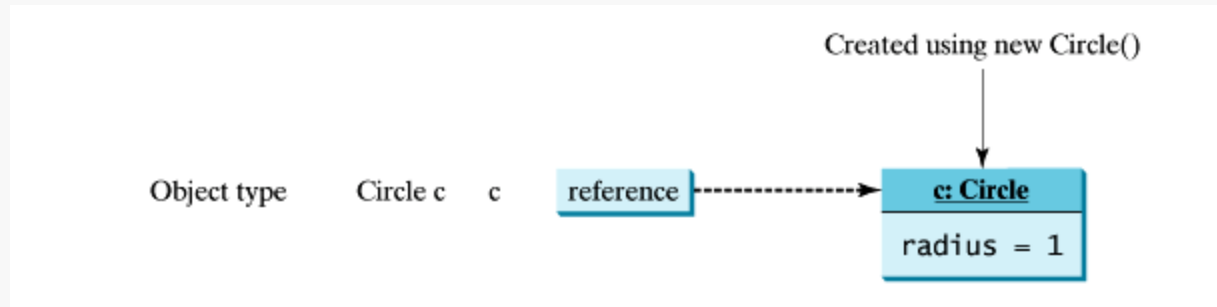
Les types primitifs

Type	Valeur par défaut
byte	0
short	0
int	0
long	0L
float	0.0f
double	0.0d
char	'\u0000'
boolean	false

Les types de référence

En Java, une référence est un type de variable qui stocke **l'adresse** d'un objet en mémoire.

Les références sont utilisées pour accéder aux objets et pour effectuer des opérations sur eux.



Les enveloppeurs (Wrappers)

Une classe d'enveloppe (ou Wrapper en Anglais) est une classe qui encapsule un type de données primitif et lui permet d'être traité comme un objet.

Les enveloppeurs fournissent des **méthodes** utiles pour travailler avec des types de données **primitifs**, tels que la conversion d'une chaîne en double ou la comparaison de deux valeurs int.

Exemple:

```
String s = "123";  
int x = Integer.parseInt(s); // 123
```

Les enveloppeurs (Wrappers)

Primitive	Wrapper Class	Constructor Arguments
<code>boolean</code>	<code>Boolean</code>	<code>boolean</code> or <code>String</code>
<code>byte</code>	<code>Byte</code>	<code>byte</code> or <code>String</code>
<code>char</code>	<code>Character</code>	<code>char</code>
<code>double</code>	<code>Double</code>	<code>double</code> or <code>String</code>
<code>float</code>	<code>Float</code>	<code>float</code> , <code>double</code> , or <code>String</code>
<code>int</code>	<code>Integer</code>	<code>int</code> or <code>String</code>
<code>long</code>	<code>Long</code>	<code>long</code> or <code>String</code>
<code>short</code>	<code>Short</code>	<code>short</code> or <code>String</code>

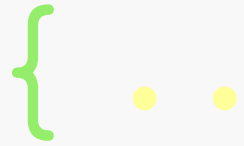
Structures conditionnelles/itératives

Schémas conditionnels : exécuter une série d'instructions dans le cas où une condition est vraie, et d'exécuter une autre série d'instructions dans le cas contraire

- if (même syntaxe qu'en C/C++)
- switch-case (**presque** la même syntaxe qu'en C/C++)

Schémas itératifs : Le traitement itératif est utilisé pour exécuter une ou plusieurs instructions plusieurs fois

- for (même syntaxe qu'en C/C++)
- while (même syntaxe qu'en C/C++)
- do-while (même syntaxe qu'en C/C++)



Outils de développement



Outils de développement

Pour développer un programme JAVA on a besoin d'un :

- **IDE**(Integrated Development Environment) : Un programme regroupant un ensemble d'outils pour le développement de logiciels. (IntelliJ, NetBeans, Eclipse)
- **JDK**(Java Development Kit) : un ensemble d'outils de développement pour Java qui comprend un compilateur (**javac**), un débogueur (**jdb**), des bibliothèques et d'autres outils nécessaires pour créer et exécuter des applications Java.
- **JRE**(Java Runtime Environment) : un ensemble d'outils nécessaires pour exécuter des applications Java sur un ordinateur, y compris la JVM.

{ ..

Premier programme



} ..

1/Lancer un éditeur de texte
et créer un fichier intitulé
«HelloWorld.java »

2/Copier le code ci-joint

3/Compiler avec la commande:

```
Javac HelloWorld.java
```

Remarquer qu'un fichier
«HelloWorld.class» est
générée

4/Exécuter en lançant la
machine virtuelle java et en
lui spécifiant le point
d'entrée:

```
Java HelloWorld
```

A screenshot of a Java IDE window titled "HelloWorld.java". The code is as follows:

```
public class HelloWorld {  
  
    /**  
     * @param args  
     */  
    public static void main(String[] args) {  
        // TODO Auto-generated method stub  
        System.out.println("Hello world!");  
    }  
}
```

{ • • Merci pour votre attention

