

# Les langages de programmation

Un langage de programmation est une notation conventionnelle destinée à formuler des algorithmes et produire des programmes informatiques qui les appliquent.

D'une manière similaire à une langue naturelle, un langage de programmation est composé :

- d'un alphabet,
- d'un vocabulaire,
- de règles de grammaire,
- et de significations

Il existe pléthore de langage de programmation, mais ils peuvent grossièrement se classer en deux catégories :

- les langages interprétés
- les langages compilés.

## Les langages interprétés

Un langage de programmation est par définition différent du langage machine.

Il faut donc le traduire pour le rendre intelligible du point de vue du processeur.

Un programme écrit dans un langage interprété a besoin d'un programme auxiliaire (l'interpréteur) pour traduire au fur et à mesure les instructions du programme.

Un interpréteur est un langage qui exécute les lignes du code source dans l'ordre où elles apparaissent, sans faire de compilation.

## Les langages compilés

Un programme écrit dans un langage compilé est traduit par un programme annexe, appelé compilateur, afin de générer un nouveau fichier autonome, c'est-à-dire n'ayant besoin d'un programme autre pour s'exécuter; on dit que ce fichier est exécutable.

L'avantage est de ne plus avoir besoin, une fois compilé, de programme annexe pour s'exécuter. De plus, il est plus rapide à l'exécution.

Toutefois il est moins souple qu'un programme interprété car à chaque modification du fichier source il faut recompiler le programme.

D'autre part, un programme compilé garanti la sécurité du code source. D'autre part, certaines applications sécurisées nécessitent la confidentialité du code pour éviter le piratage.

## Les langages intermédiaires

Certains langages appartiennent en quelque sorte aux deux catégories Java, Python, ..) car le programme écrit avec ces langages peut dans certaines conditions subir une phase de compilation intermédiaire vers un fichier écrit dans un langage qui n'est pas intelligible (différent du fichier source) et non exécutable (nécessité d'un interpréteur).

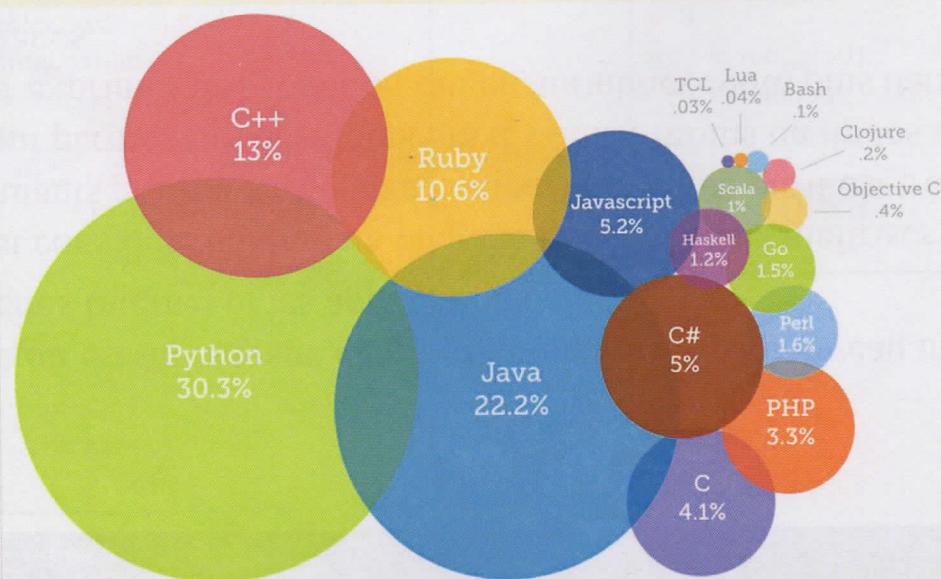
Les applets Java, petits programmes insérés parfois dans les pages Web, sont des fichiers qui sont compilés mais que l'on ne peut exécuter qu'à partir d'un navigateur internet (ce sont des fichiers dont l'extension est .class).

## Exemples de langages utilisés

Langage	Domaine d'application principal	Compilé/interprété
<b>BASIC</b>	Programmation basique à but éducatif	Langage interprété
<b>C</b>	Programmation système	Langage compilé
<b>C++</b>	Programmation système objet	Langage compilé
<b>Java</b>	Programmation orientée internet	Langage intermédiaire
<b>Python</b>	Programmation objet	Langage intermédiaire
<b>PHP</b>	Développement de sites web dynamiques	Langage interprété
<b>Perl, RuBy</b>	Traitement de chaînes de caractères	Langage interprété

## Présentation des différents langages

Les langages de programmation sont pléthore. Nous allons donc présenter les langages les plus utilisés à l'heure actuelle.



## JAVA

Le Java est un langage de programmation orienté objet qui a été développé par Sun Microsystems dans les années 90.

C'est l'un des langages de programmation les plus populaires, devenu un standard chez les programmes d'entreprise, dans les contenus et jeux Web ou encore les applications mobiles.

C'est également le langage de programmation utilisé pour le système d'exploitation mobile Android.

Le Java est fait pour travailler sur de nombreuses plateformes : un programme codé sur Mac OS X pourrait par exemple fonctionner sur Windows

Le Cnam Rhône-Alpes - 4 rue Davier - 69007 Lyon - 04 78 58 30 99

## Les caractéristiques de JAVA

Java possède des caractéristiques qui ont contribué à son succès :

- C'est un langage interprété,
- portable : indépendant de toute plateforme,
- orienté objet,
- Simple : abandon de la notion de pointeur, de la surcharge des opérateurs, ... ,
- Fortement typé,
- Assure la gestion de la mémoire : allocation de récupération automatique de la mémoire,
- Sûr,
- Économe : les bibliothèques ne sont liées qu'à l'exécution
- Multitâche : il exécute du code à plusieurs endroits de façon indépendante.

Le Cnam Rhône-Alpes - 4 rue Davier - 69007 Lyon - 04 78 58 30 99

## Les caractéristiques de JAVA

Il existe 2 types de programmes avec la version standard de Java : les applets et les applications.

Une application autonome (stand alone program) est une application qui s'exécute sous le contrôle direct du système d'exploitation.

Une applet est une application qui est chargée par un navigateur et qui est exécutée sous le contrôle d'un plug in de ce dernier.

Les principales différences entre une applet et une application sont :

- les applets n'ont pas de méthode main() : la méthode main() est appelée par la machine virtuelle pour exécuter une application.
- les applets ne peuvent pas être testées avec l'interpréteur. Elles doivent être testées avec l'applet viewer ou doivent être intégrées à une page HTML, elle-même visualisée avec un navigateur disposant d'un plug in Java, .

## Différentes éditions de Java

Sun/Oracle fournit gratuitement un ensemble d'outils et d'API pour permettre le développement de programmes avec Java.

Ce kit, nommé JDK, est librement téléchargeable sur le site web d'Oracle.

Le JRE (Java Runtime Environment) contient uniquement l'environnement d'exécution de programmes Java.

Le JDK contient lui-même le JRE.

Le JRE seul doit être installé sur les machines où des applications Java doivent être exécutées.

Trois plate-formes d'exécution Java sont définies selon les besoins des applications à développer :

- Java Standard Edition : environnement d'exécution et ensemble complet d'API pour des applications de type desktop.
- Java Enterprise Edition : Java SE pour le développement applications entreprises
- Java Micro Edition : Pour le développement applications sur appareils mobiles et embarqués

## Différentes éditions de Java

Avec différentes éditions, les types d'applications qui peuvent être développées en Java sont nombreux et variés :

- Applications desktop
- Applications web : servlets/JSP, portlets, applets
- Applications pour appareil mobile (MIDP) : midlets
- Applications pour appareil embarqué (CDC) : Xlets
- Applications pour carte à puce (Javacard) : applets Javacard
- Applications temps réel

## Les API

**Définition :** Une API a pour objet de faciliter le travail d'un programmeur en lui fournissant les outils de base nécessaires à tout travail à l'aide d'un langage donné.

Elle constitue une interface servant de fondement à un travail de programmation plus poussé.

Or, la communauté Java est très productive. C'est pourquoi, l'ensemble des API et des outils utilisables est énorme et évolue très rapidement.

## Les concepts de base

La plate-forme Java utilise quelques notions de base lors de sa mise en oeuvre, notamment :

- compilation du code source dans un langage indépendant de la plate-forme d'exécution : le bytecode
- Exécution du bytecode par une machine virtuelle nommée JVM
- la notion de package qui permet d'organiser les classes
- le classpath qui permet de préciser au compilateur et à la JVM où elle peut trouver les classes requises par l'application
- le packaging des classes compilées dans une archive de déploiement nommée jar.

## La compilation et l'exécution

Un programme Java est composé de plusieurs fichiers source.

Ces fichiers source possèdent l'extension .java.

Ils peuvent contenir une ou plusieurs classes ou interfaces mais il ne peut y avoir qu'une seule classe ou interface publique par fichier.

Le nom de ce fichier source doit correspondre au nom de cette entité publique suivi de l'extension .java

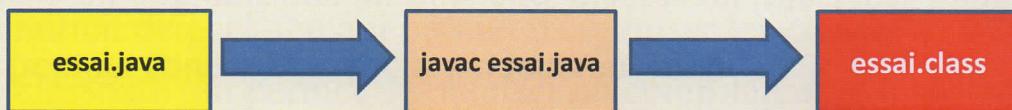
Il est nécessaire de compiler le source pour le transformer en J-code ou bytecode Java qui sera lui exécuté par la machine virtuelle.

Pour être compilé, le programme doit être enregistré au format de caractères Unicode : une conversion automatique est faite par le JDK

## La compilation et l'exécution

Un compilateur Java, par exemple l'outil javac fourni avec le JDK, est utilisé pour compiler chaque fichier source en fichier de classe possédant l'extension `.class`.

Cette compilation gère pour chaque fichier source un ou plusieurs fichiers `.class` qui contiennent du bytecode.



## Installation des outils de développement

L'un des principes phares de Java réside dans sa machine virtuelle : celle-ci assure à tous les développeurs Java qu'un programme sera utilisable avec tous les systèmes d'exploitation sur lesquels est installée une machine virtuelle Java.

Pour pouvoir créer des programmes Java, il faut donc installer les outils suivants :

- Installer le JRE en téléchargeant la dernière version stable sur le site d'Oracle <http://www.oracle.com/technetwork/java/javase/downloads/index.html>
- Installer Eclipse en téléchargeant la version Eclipse IDE for Java Developers sur <http://www.eclipse.org/downloads/>

« Eclipse IDE » est un environnement de développement libre permettant de créer des programmes en Java, C++, PHP.... C'est l'outil que nous allons utiliser pour programmer.

# Présentation d'eclipse

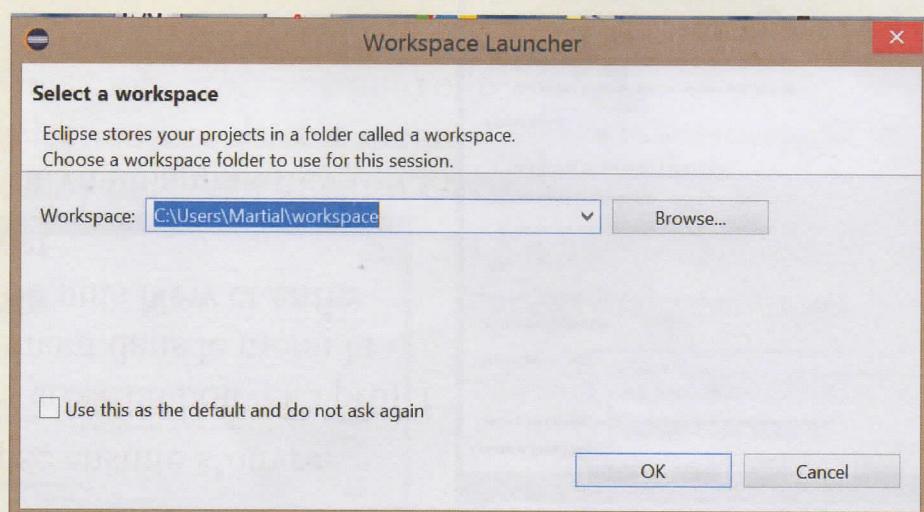
Une fois installer, lancer le programme Eclipse.exe dans le répertoire Eclipse.

configuration	15/01/2015 00:44	Dossier de fichiers
dropins	25/09/2014 20:52	Dossier de fichiers
features	15/01/2015 00:30	Dossier de fichiers
p2	15/01/2015 00:30	Dossier de fichiers
plugins	15/01/2015 00:30	Dossier de fichiers
readme	15/01/2015 00:30	Dossier de fichiers
.eclipseproduct	13/08/2014 14:06	Fichier ECLIPSEPRO... 1 Ko
<b>eclipse.exe</b>	<b>25/09/2014 20:52</b>	<b>Application 313 Ko</b>
eclipse.ini	25/09/2014 20:52	Paramètres de confi... 1 Ko
eclipsec.exe	25/09/2014 20:52	Application 26 Ko
epl-v10.html	13/08/2014 14:06	Chrome HTML Docu... 13 Ko
notice.html	13/08/2014 14:06	Chrome HTML Docu... 9 Ko

Le Cnam Rhône-Alpes - 4 rue Davier - 69007 Lyon - 04 78 58 30 99

# Présentation d'eclipse

Au démarrage apparait la boite de dialogue suivante :



Sachez que rien ne vous empêche de spécifier un autre dossier que celui proposé par défaut.

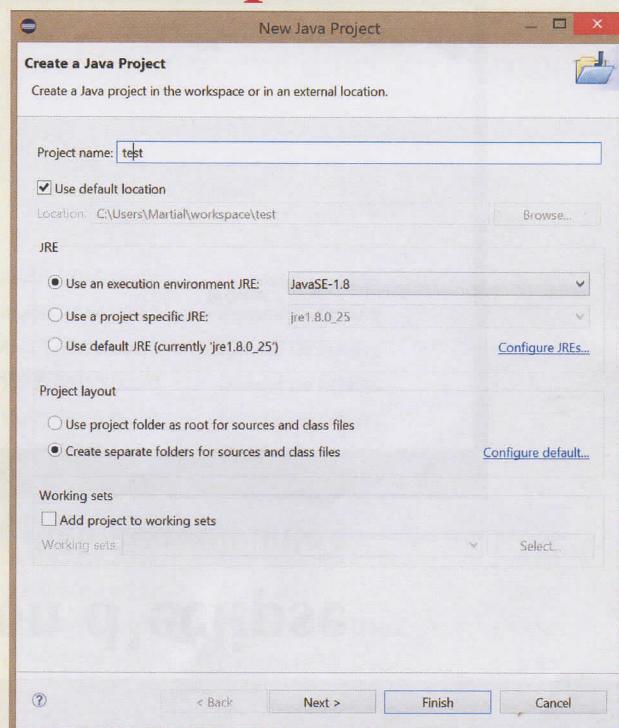
Le Cnam Rhône-Alpes - 4 rue Davier - 69007 Lyon - 04 78 58 30 99

## Présentation d'eclipse

JAVA Eclipse ensuite s'ouvre.

Il faut alors créer un nouveau projet en sélectionnant dans le menu la rubrique **File** puis **New** et enfin **Java Project**.

La fenêtre suivante apparaît :

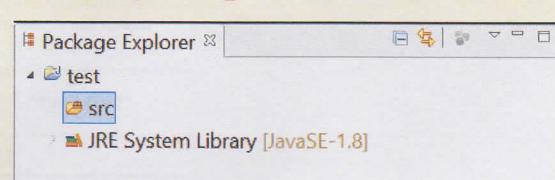
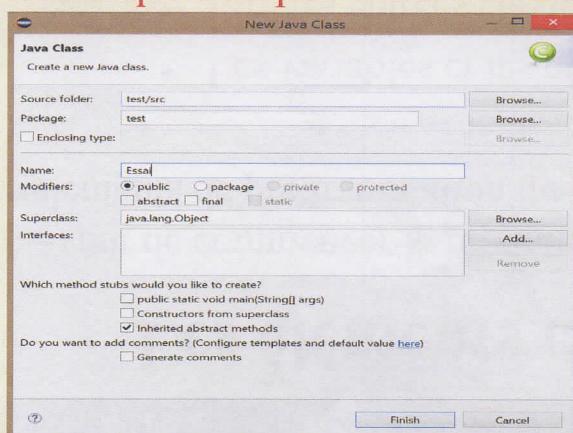


Le Cnam Rhône-Alpes - Avenue Ravier - 69007 Lyon - 04 78 58 30 99

## Présentation d'eclipse

Le projet ainsi créé, dans l'onglet intitulé **Package Explorer**, un nouveau répertoire s'affiche :

Nous pouvons alors créer une nouvelle classe en sélectionnant dans le menu la rubrique **File** puis **New** et enfin **Class**.



La nouvelle classe créée, cette dernière apparaît ainsi

```
Essai.java
1 package test;
2
3 public class Essai {
4
5 }
```

Nous pouvons alors commencer à programmer.

# Les éléments de programmation

Avant de commencer la programmation, on présente les différents éléments de programmation que sont :

- Les variables et les opérateurs
- Lire les entrées clavier
- Les conditions
- Les boucles
- Les tableaux
- Les méthodes de classe

# Les variables

Les principaux types de variables en JAVA sont :

Type	Signification	Taille (en octets)	Plage de valeurs acceptées
char	Caractère Unicode	2	'\u0000' ? '\uffff' (0 à 65535)
byte	Entier très court	1	-128 ? +127
short	Entier court	2	-32 768 ? +32 767
int	Entier	4	-2 <sup>31</sup> ?-2,147×10 <sup>9</sup> ? +2 <sup>31</sup> -1? 2,147×10 <sup>9</sup>
long	Entier long	8	-2 <sup>63</sup> ?-9,223×10 <sup>18</sup> ? +2 <sup>63</sup> -1? 9,223×10 <sup>18</sup>
float	Nombre réel simple	4	±2 <sup>-149</sup> ?1.4×10 <sup>-45</sup> ? ±2 <sup>128</sup> -2 <sup>104</sup> ? 3.4×10 <sup>38</sup>
double	Nombre réel double	8	±2 <sup>-1074</sup> ?4.9×10 <sup>-324</sup> ? ±2 <sup>1024</sup> - 2 <sup>971</sup> ?1.8×10 <sup>308</sup>
boolean	Valeur logique (booléen)	1	true (vrai), ou false (faux)

Déclaration de la variable : <Type variable> <Nom variable> ;

## Les opérateurs

Opérateur	Description	Type	Associativité
()	Appel de méthode	classes et objets	de gauche à droite
[]	Elément d'un tableau	tableaux	
.	Membre d'une classe ou d'un objet	classes et objets	
++	Incrémentation post ou pré-fixée	byte char short int long float double	de droite à gauche
--	Décrémentation post ou pré-fixée	byte char short int long float double	
+	Positif	byte char short int long float double	
-	Négation	byte short int long float double	
!	Non logique	boolean	
~	Non binaire	byte char short int long	
(type)	Conversion de type	tous	de gauche à droite
*	Multiplication	byte char short int long float double	
/	Division	byte char short int long float double	
%	Modulo (reste de la division entière)	byte char short int long	de gauche à droite
+	Addition	byte char short int long float double String (concaténation)	
-	Soustraction	byte char short int long float double	

## Les opérateurs

Opérateur	Description	Type	Associativité
<<	Décalage de bit vers la gauche	byte char short int long	de gauche à droite
>>	Décalage de bit vers la droite (signe conservé)	byte char short int long	
>>>	Décalage de bit vers la droite (signe décalé)	byte char short int long	
<	Inférieur	byte char short int long float double	de gauche à droite
<=	Inférieur ou égal	byte char short int long float double	
>	Supérieur	byte char short int long float double	
>=	Supérieur ou égal	byte char short int long float double	de gauche à droite
==	Egal	byte char short int long float double object	
!=	Different	byte char short int long float double object	
&	ET binaire	byte char short int long boolean	de gauche à droite
^	OU exclusif binaire	byte char short int long boolean	de gauche à droite
	OU binaire	byte char short int long boolean	de gauche à droite
&&	ET logique	boolean	de gauche à droite
	OU logique	boolean	de gauche à droite

## Les opérateurs

Opérateur	Description	Type	Associativité
?:	Opérateur ternaire de condition	boolean ? tous : tous	de droite à gauche
=	Affectation	tous	de droite à gauche
+=	Addition et affectation	byte char short int long float double String ( <i>concaténation</i> )	
-=	Soustraction et affectation	byte char short int long float double	
*=	Multiplication et affectation	byte char short int long float double	
/=	Division et affectation	byte char short int long float double	
%=	Modulo et affectation	byte char short int long float double	
<=>	Décaler à gauche et affectation	byte char short int long	
>=>	Décaler à droite (excepté signe) et affectation	byte char short int long	
>>>=	Décaler à droite (signe aussi) et affectation	byte char short int long	
&=	ET binaire et affectation	byte char short int long boolean	
^=	OU exclusif binaire et affectation	byte char short int long boolean	
=	OU binaire et affectation	byte char short int long boolean	
,	Enchaînement d'expressions	tous	de gauche à droite