

Formation Java



Emmanuel Fernandez
Avril 2022

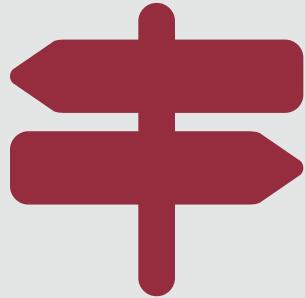
Les langages de programmation

Un langage de programmation c'est quoi ?

- Une notation conventionnelle
 - destinée à formuler des algorithmes et produire des programmes informatiques qui les appliquent
- Un moyen de communication
 - par lequel le développeur communique avec l'ordinateur

Les langages de programmation

Choisir son langage



Paradigme

Procédural POO

Evènementiel

Nature

Langage compilé

Langage interprété

Hybride



Code source



Compilateur



Fichier binaire

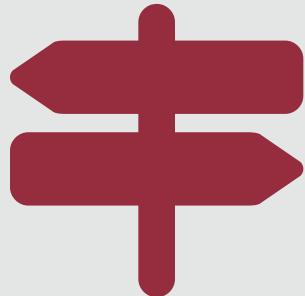


salut

```
Print ("Salut");
```

Les langages de programmation

Choisir son langage



Paradigme

Procédural POO

Evènementiel

Nature

Langage compilé

Langage interprété

Hybride



Code source



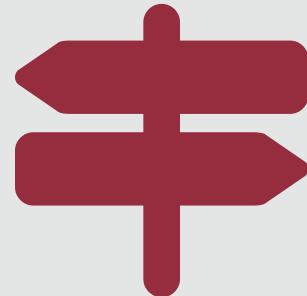
Interpréteur



```
Print("Salut");
```

Les langages de programmation

Choisir son langage



Paradigme

Procédural

POO

Evènementiel

Nature

Langage compilé

Langage interprété

Hybride



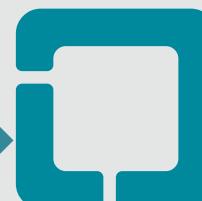
Code source



Compilateur



Fichier bytecode



interpréteur



```
Print ("Salut");
```

Les langages de programmation

Classification des langages

Langage
haut niveau



Langage
bas niveau



Le langage
assembleur

Les langages de programmation

Popularité des langages



TIOBE

Langage	2017	2012	2007	2002	1997
Java	1	2	1	1	15
C	2	1	2	2	1
C++	3	3	3	3	2
C#	4	5	7	12	-
Python	5	7	6	11	27
Visual Basic .Net	6	17	-	-	-

La plateforme Java Historique

Write once run anywhere

1990

1995

2009

2014



Naissance

Officialisation

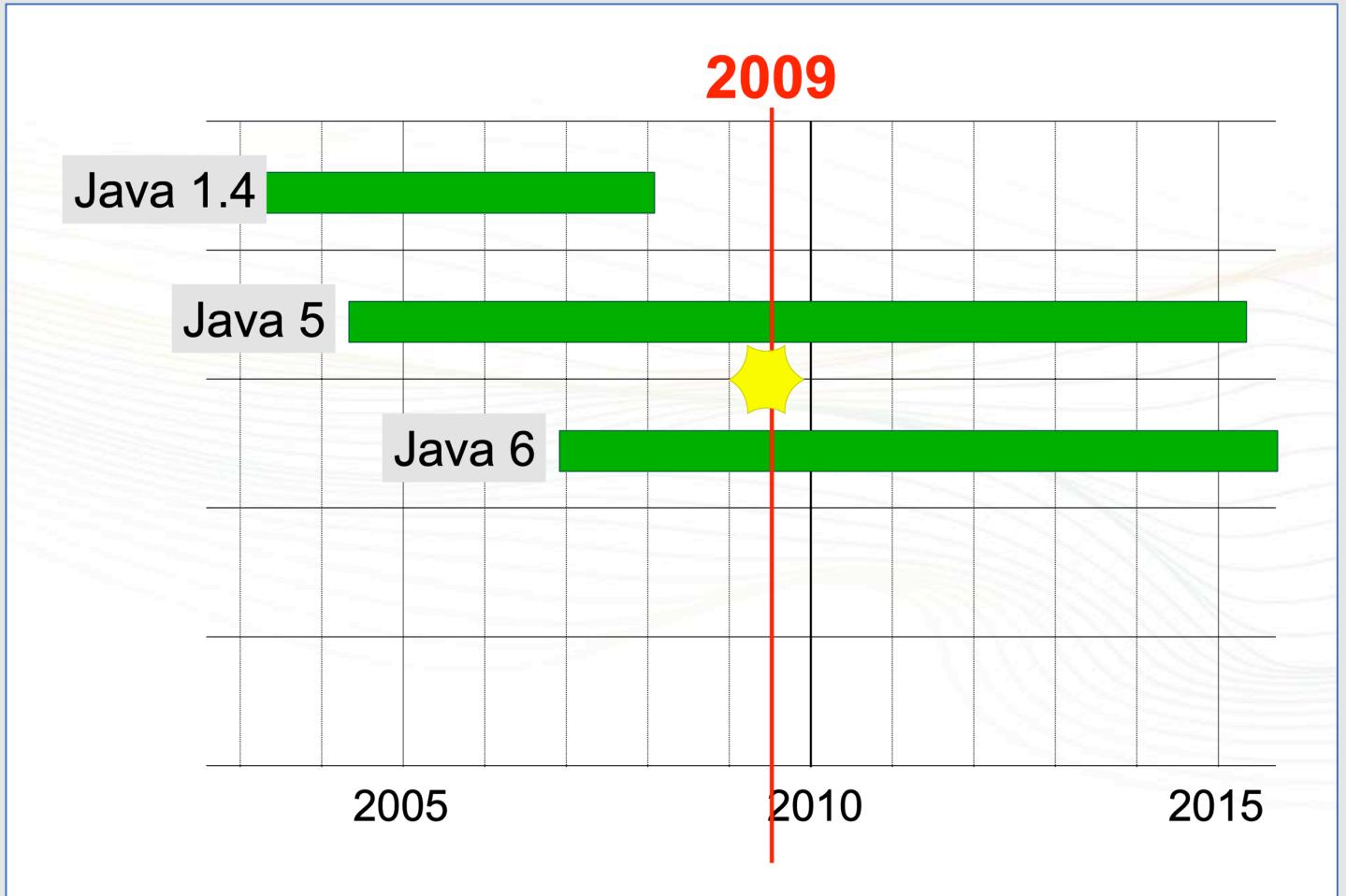
Oracle

Java 8

La plateforme Java

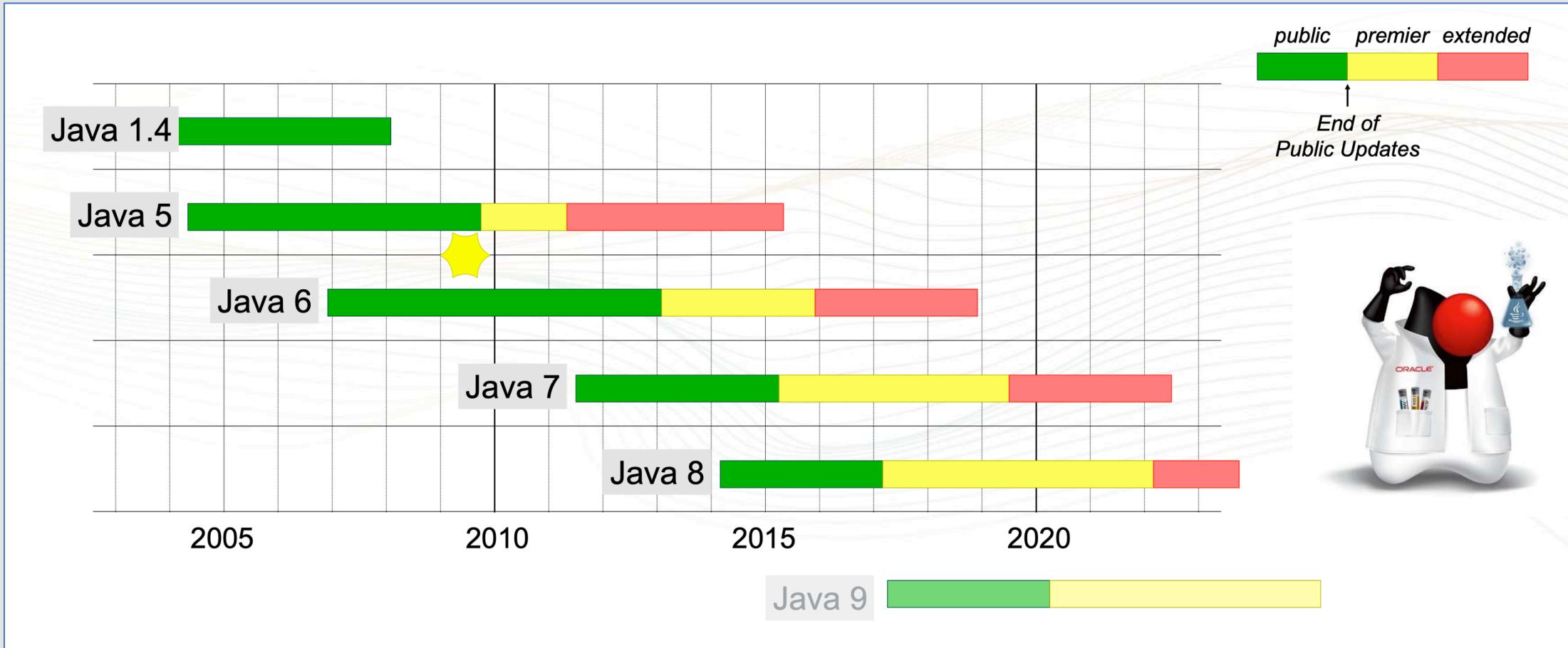
<http://java.sun.com>

- Support gratuit pendant 10 ans
- Java devient open source à partir de 2007 grâce à la création par Sun du projet [OpenJDK](#)
- Sun racheté en 2009 par Oracle



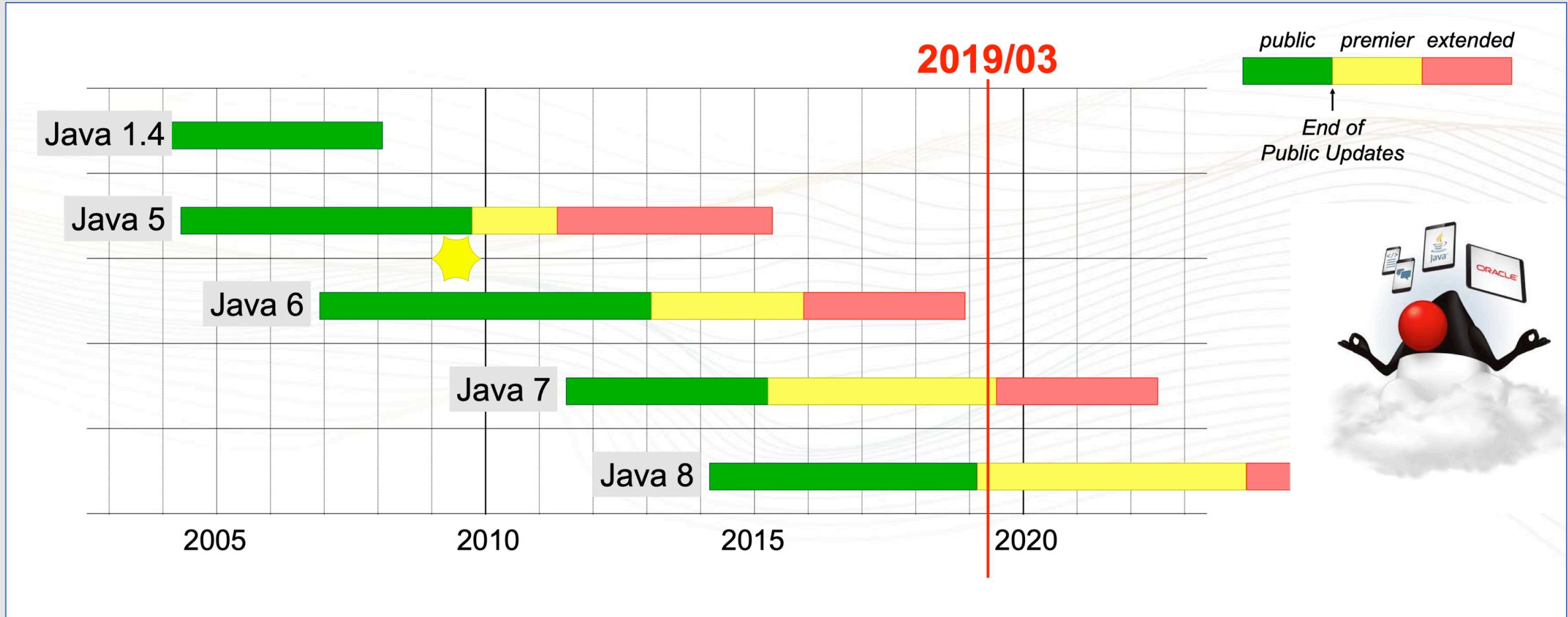
La plateforme Java

Oracle Java



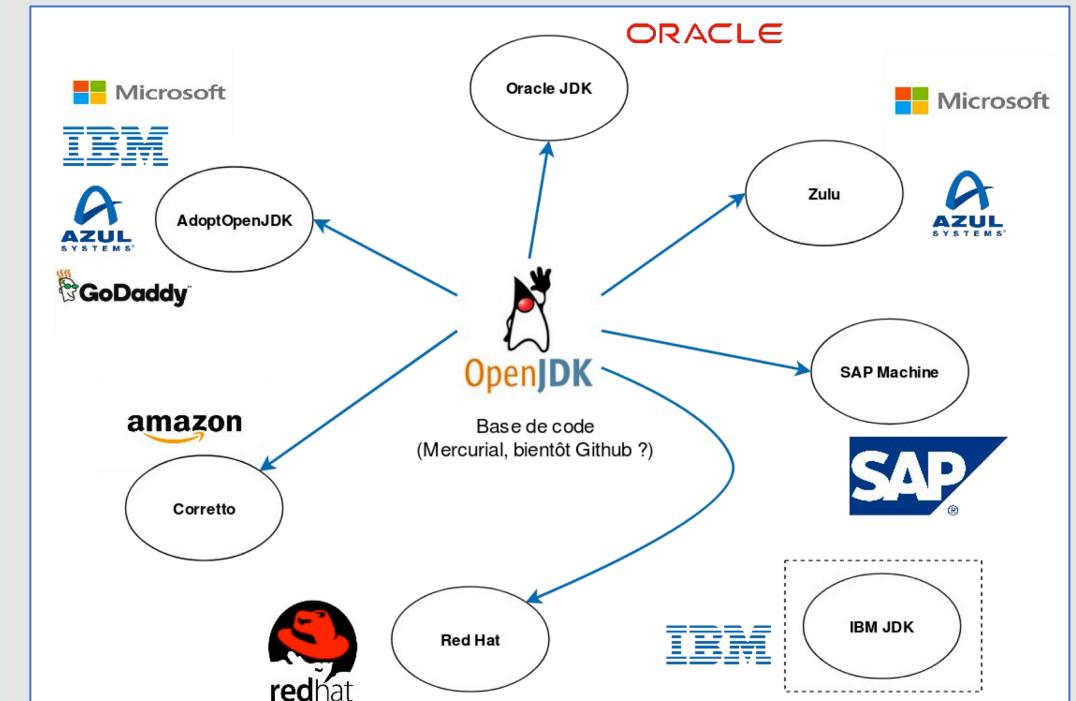
La plateforme Java

Oracle Java

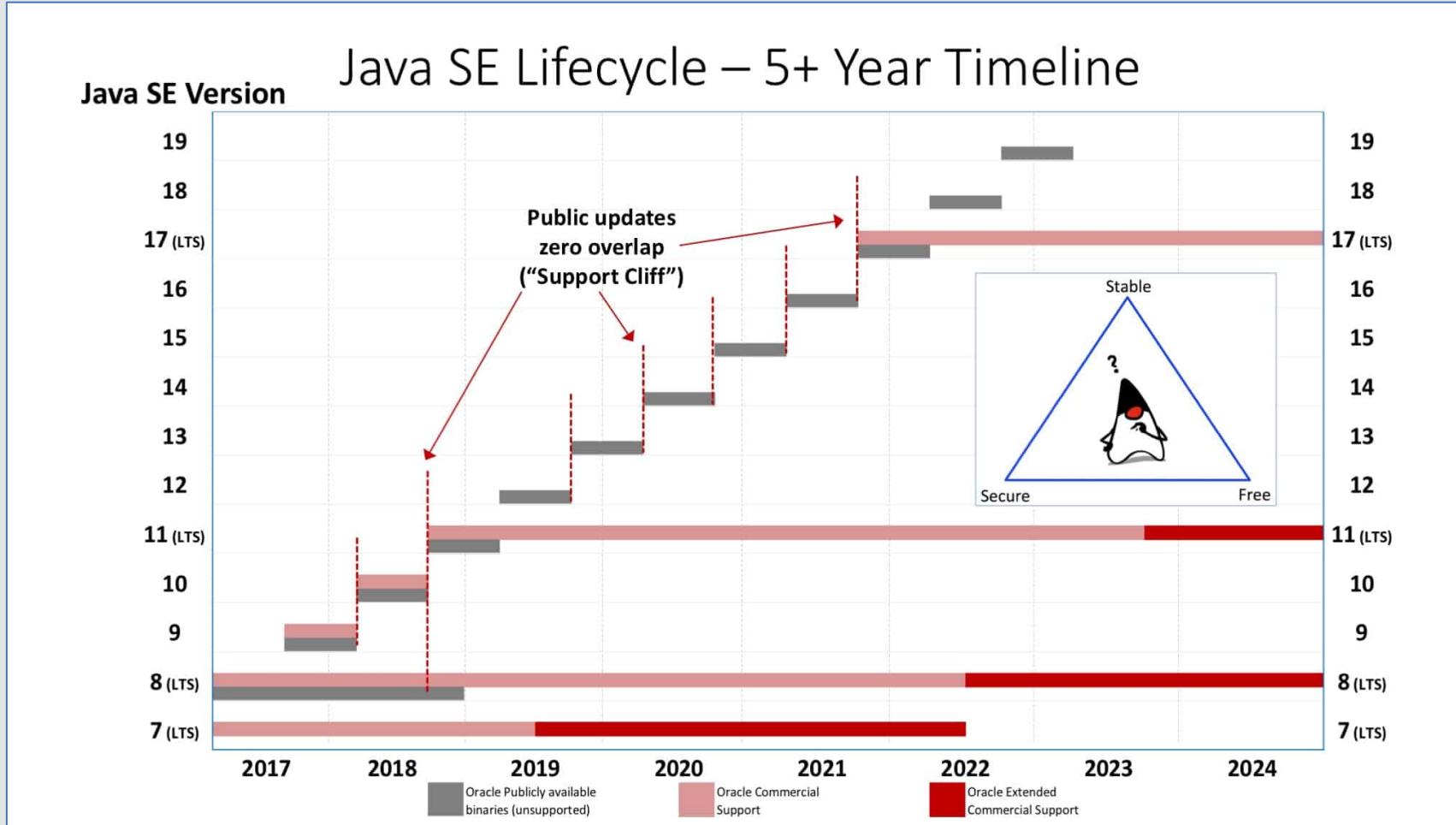


Pourquoi Oracle Java ?

- Java est Open Source
 - Il est sous licence GPL avec ClassPath Exception (chaque contributeur doit partager ses sources modifiés de la JVM)
- Il existe d'autres fournisseurs de binaires (avec possibilité de support)
 - Fin de support gratuit d'Oracle ne veut pas dire fin de vie d'OpenJDK



Nouveau cycle de release Java



Téléchargement

- Dernière version de Java
 - (au 22/07/2020, il s'agit du JDK 14.0.2)
 - Version Oracle JDK : <https://www.oracle.com/java/technologies/javase-downloads.html>
 - Version Oracle OpenJDK : <http://jdk.java.net/14/>
 - Version AdoptOpenJDK : <https://adoptopenjdk.net/index.html>
- Version plus ancienne (test avec la dernière version de Java 8)
 - Version Oracle JDK (8u261) : <https://www.oracle.com/java/technologies/javase-downloads.html>
 - Version Oracle OpenJDK : Plus fournie (plus ancienne version : 9 GA)
 - Version AdoptOpenJDK (8u262) : <https://adoptopenjdk.net/index.html>

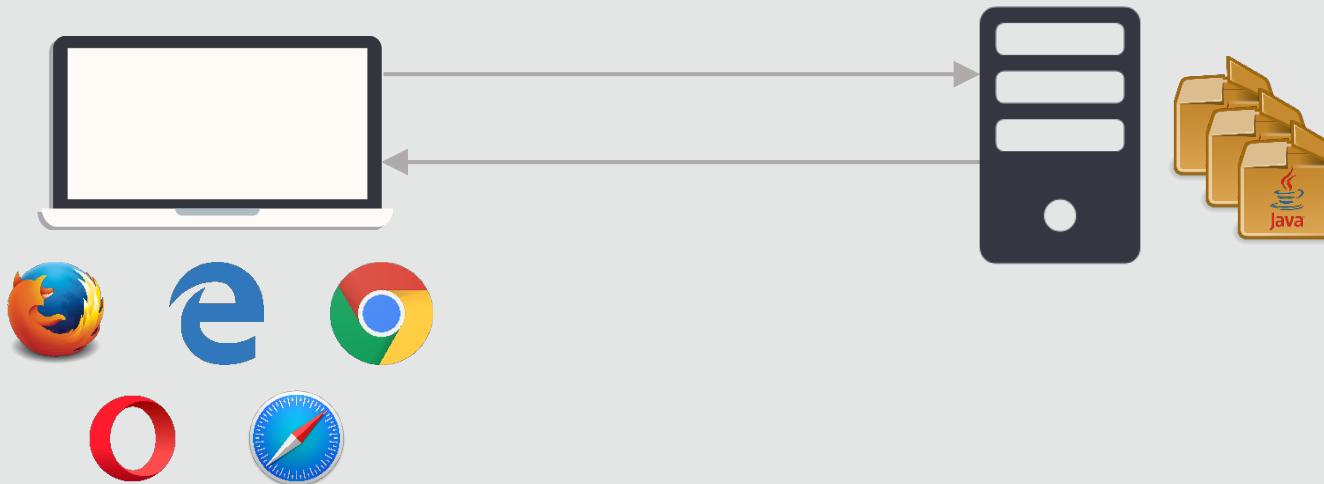
La plateforme Java

Plateforme Java SE



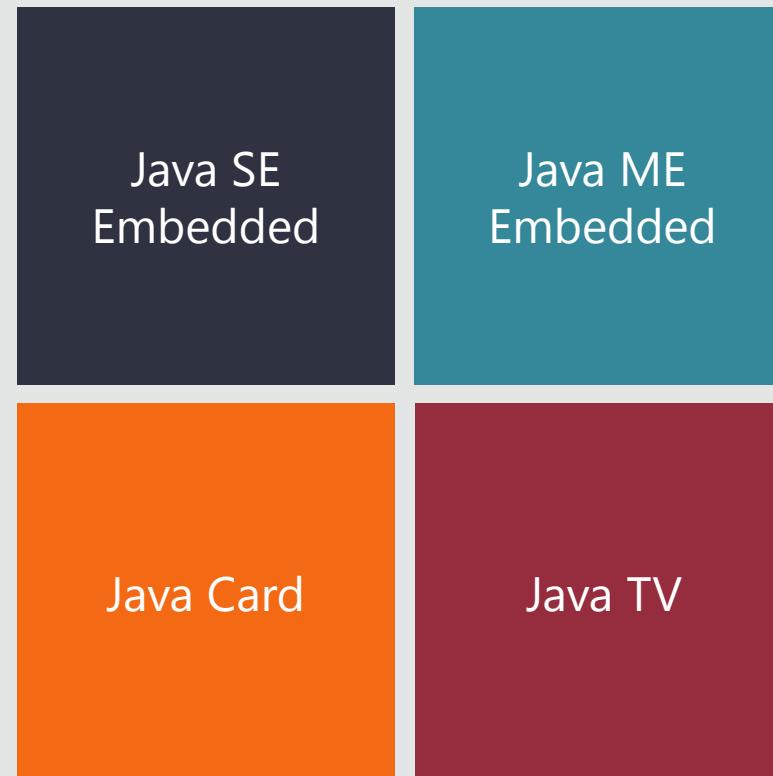
La plateforme Java

Plateforme Java EE



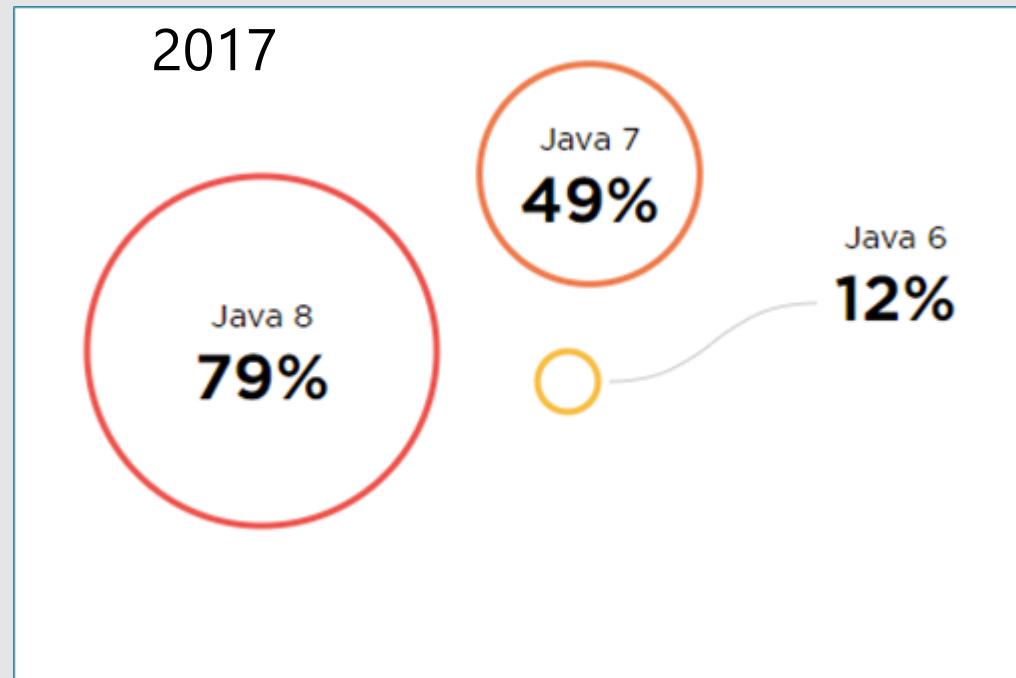
La plateforme Java

Plateformes Java Embedded



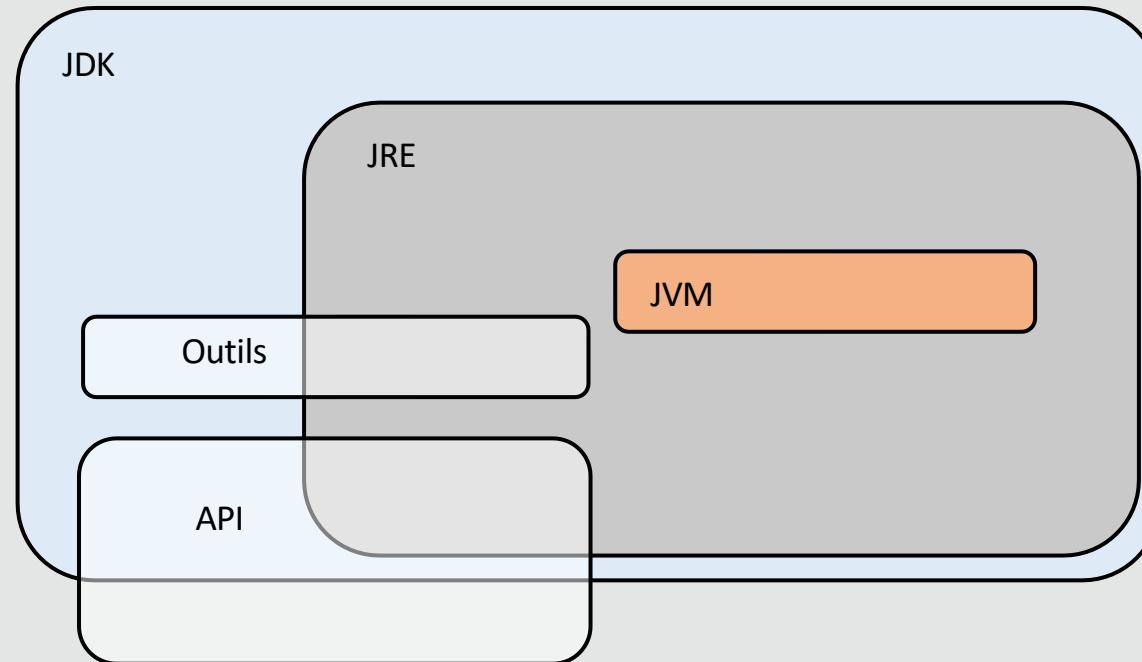
La plateforme Java

Version populaire de Java SE



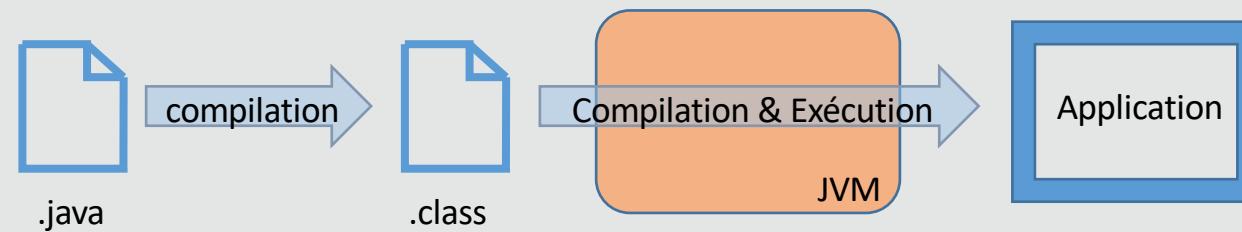
La plateforme Java

La plateforme Java SE



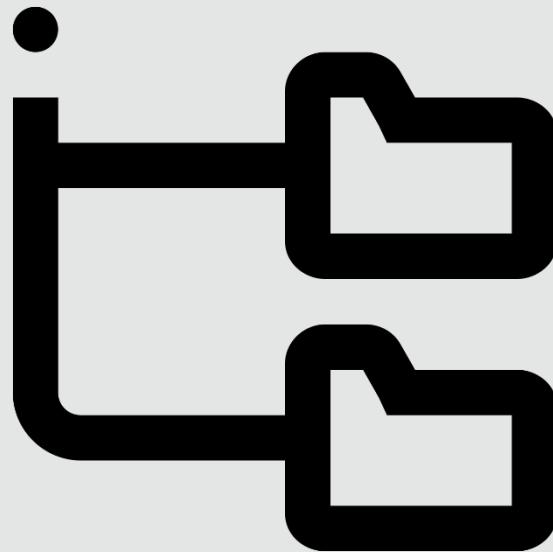
La plateforme Java

La machine virtuelle



La plateforme Java

Le classpath



La plateforme Java

Les principaux outils

javac

java

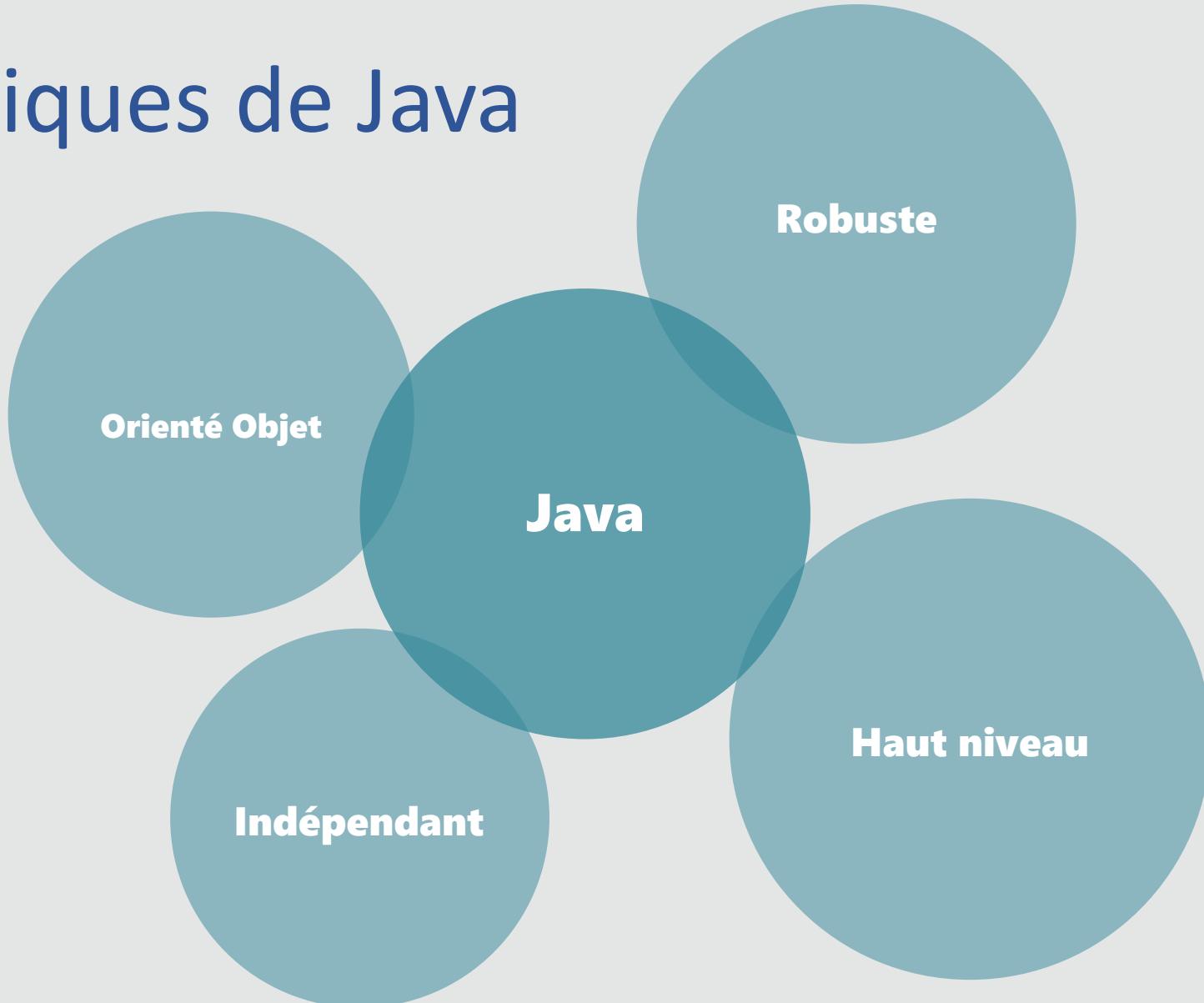
javaw

javadoc

jdb

La plateforme Java

Caractéristiques de Java



Premier programme

```
/**  
 * Description de la classe.  
 *  
 * @author EFE  
 */  
public class Test {  
    /**  
     * Description de la méthode.  
     */  
    public static void main(String[] args)  
    {  
        System.out.println("Hello world !!!");  
    }  
}
```

Compilation et exécution

- Compiler

```
javac HelloWorld.java
```

- Exécuter

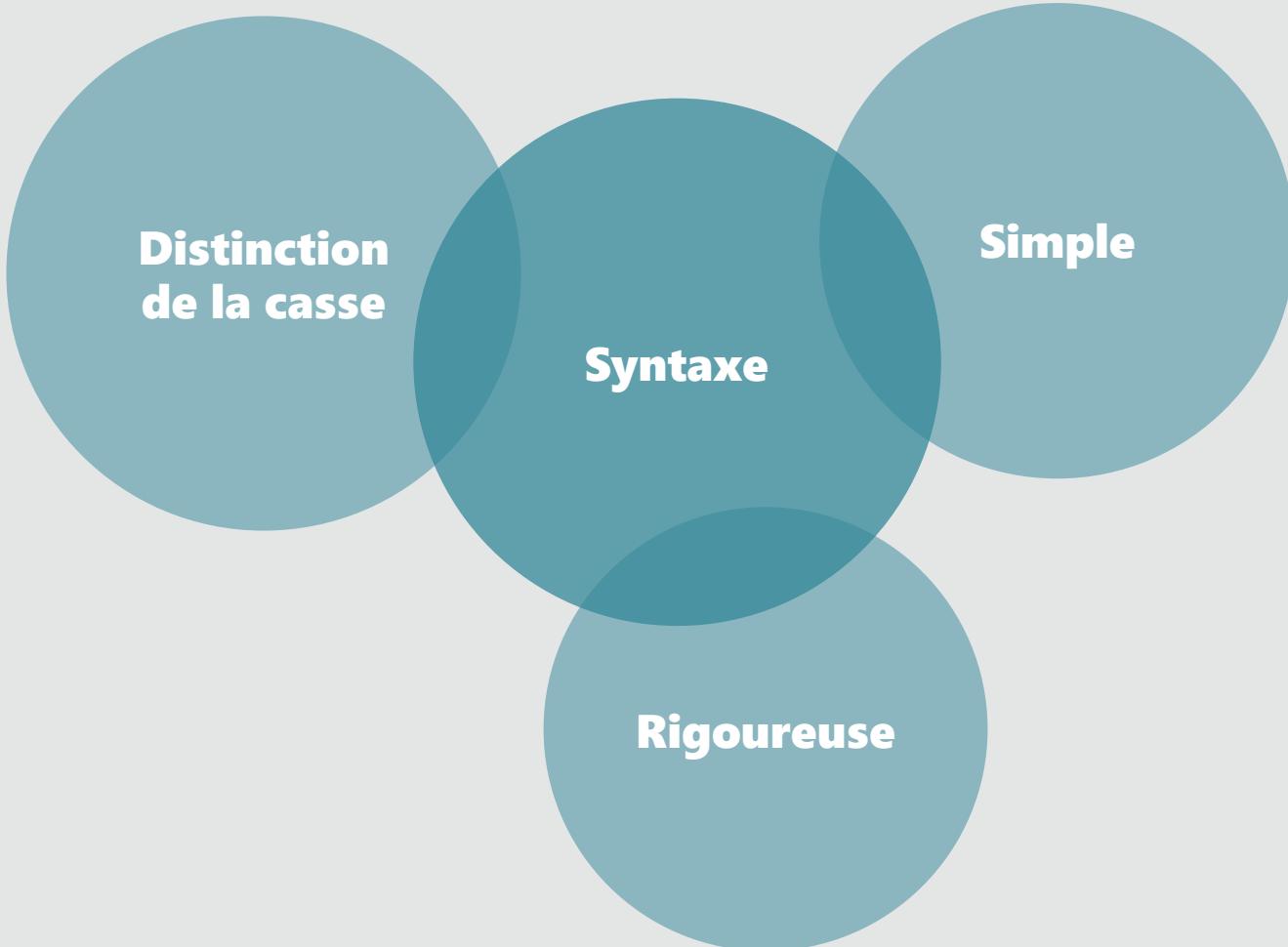
```
java HelloWorld
```

- Documenter

```
javadoc -version -author -d HelloWorld.java
```

Les fondamentaux du langage Java

Syntaxe



Les fondamentaux du langage Java

Généralités

- Bloc d'instructions { }
- Fin d'instruction ;
- Commentaires
 - // ceci est un commentaire sur une ligne
 - /* ceci est un commentaire sur plusieurs lignes */
 - /** ceci est un commentaire Javadoc */

Variables

- Nom d'une variable : composé de **lettres, chiffres, _**
- Ne commence pas par un chiffre
- Pas un mot réservé
- **lowerCamelCase**

Déclaration d'une variable

type nomVariable [=valeur] ;

Portée et durée de vie d'une variable

```
public class HelloWorld {  
  
    public static void main(String[] args)  
    {  
        int nombre = 10;  
  
        if(nombre == 10)  
        {  
            char lettre = 'A';  
            System.out.println("Valeur du nombre = " + nombre);  
            System.out.println("Valeur du caractère = " + lettre);  
        }  
        System.out.println("Valeur du nombre = " + nombre);  
    }  
}
```

Les fondamentaux du langage Java

Les types primitifs

Contenu	Type	Valeur par défaut (variables membres)	Taille en bit
entier	byte	0	8
	short	0	16
	int	0	32
	long	0	64
décimaux	float	0,0	32
	double	0,0	64
caractère	char	'\u0000'	16
boolean	boolean	false	8

Les fondamentaux du langage Java

Les chaînes de caractères

Contenu	Type	Valeur par défaut
Chaîne caractères	String	null

Les fondamentaux du langage Java

Exemples

```
public class DemosTypes {  
  
    public static void main(String[] args) {  
        boolean estPremier = true;  
  
        int index = 0;  
  
        long nombreDeGrains = 500l;  
  
        float taillePaquet = 51.20f;  
  
        double tailleTour = 105.9;  
  
        char belleLettre = 'A';  
  
        String bellePhrase = "On ne voit bien qu'avec le coeur";  
    }  
}
```

Les conversions entre types primitifs

Implicite

```
int unEntier = 10;  
  
long unLong = unEntier;  
  
System.out.println(unLong);
```

Explicite

```
double unDouble = 10.23;  
  
float unFloat = (float)unDouble;  
  
System.out.println("unFloat : " + unFloat);
```

Les fondamentaux du langage Java

Les constantes

```
final String NOM = "Legrand";
```

Les fondamentaux du langage Java

Les principaux opérateurs d'affectation

int a, b; **a=15;** **b=7;**

Opérateur	Utilisation	Equivalent à
<code>+=</code>	<code>a+=b</code>	<code>a = a + b</code>
<code>-=</code>	<code>a-=b</code>	<code>a = a - b</code>

Les structures de contrôle

- Séquences
 - Expressions de calcul
 - Lire et écrire sur la console
- Conditionnelles
- Répétitives

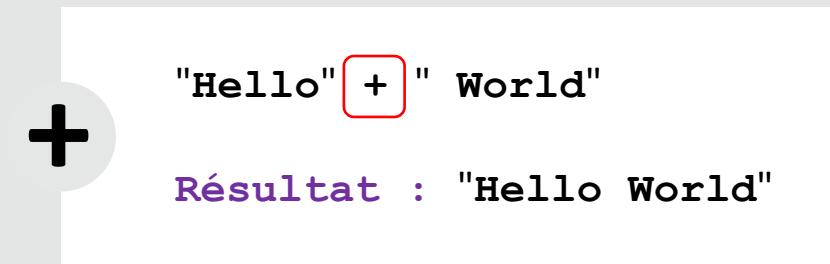
Les expressions de calcul

- Opérateurs arithmétiques

Opérateur	Exemple	Résultat
+	$6 + 4$	10
-	$10 - 4$	6
*	$6 * 6$	36
/	$36 / 6$	6
%	$6 \% 5$	1
++	<code>int i = 2; i++;</code>	3
--	<code>int i = 2; i--;</code>	1

Les expressions de calcul

- Opérateur de concaténation



Lire et écrire sur la console

```
public static void main(String[] args)
{
    System.out.println("Veuillez saisir une valeur :");
    Scanner scan = new Scanner(System.in);
    String saisie = scan.nextLine();

    //scan.nextInt();
    //scan.nextLong();
    //scan.nextFloat();
    //...
}
```

Les structures de contrôle

- Séquences
 - Expressions de calcul
 - Lire et écrire sur la console
- Conditionnelles
- Répétitives

Les fondamentaux du langage Java

Les conditionnelles

- Opérateurs de comparaison

Opérateur	Exemple	Résultat
>	$2 > 5$	false
\geq	$2 \geq 5$	false
<	$2 < 5$	true
\leq	$2 \leq 5$	true
\equiv	$2 \equiv 5$	false
\neq	$2 \neq 5$	true

Les fondamentaux du langage Java

Les conditionnelles

- Opérateurs logiques

Opérateur	Exemple	Résultat
&	true & true	true
	true false	true
^	true ^ true	false
!	!false	true
&&	false && true	false
	true false	true

Si alors sinon

```
public static void main(String[] args) {
    int age = 18;

    boolean majeur = false;

    if(age > 18)
    {
        majeur = true;
    }
    else
    {
        majeur = false;
    }
}
```

Les fondamentaux du langage Java

Si alors sinon

```
public static void main(String[] args) {
    int numero = 8;

    if(numero == 1)
    {
        System.out.println("numéro est égal à 1");
    }
    else if(numero == 2)
    {
        System.out.println("numéro est égal à 2");
    }
    else{
        System.out.println("numéro est différent de 1 et 2");
    }
}
```

Branchement conditionnel

```
public static void main(String[] args) {  
    int jour = 5;  
  
    switch(jour) {  
        case 1:System.out.println("Lundi");break;  
        case 2:System.out.println("Mardi");break;  
        case 3:System.out.println("Mercredi");break;  
        case 4:System.out.println("Jeudi");break;  
        case 5:System.out.println("Vendredi");break;  
        case 6:System.out.println("Samedi");break;  
        case 7:System.out.println("Dimanche");break;  
        default:System.out.println("Erreur");  
    }  
}
```

Les structures de contrôle

- Séquences
 - Expressions de calcul
 - Lire et écrire sur la console
- Conditionnelles
- Répétitives

Les fondamentaux du langage Java

Boucle Tant que

```
public static void main(String[] args) {  
    int i = 0;  
  
    while(i<10){  
        i++;  
        System.out.println("i = " + i);  
    }  
}
```

Boucle Faire ...Tant que

```
public static void main(String[] args) {  
    int i = 0;  
  
    do {  
        i++;  
        System.out.println("i = " + i);  
    }while(i<10);  
}
```

Les fondamentaux du langage Java

Boucle Pour

```
public static void main(String[] args) {  
    for(int i = 0;i<5;i++)  
    {  
        System.out.println("i = " + i);  
    }  
}
```

La procédure

- Bloc d'instructions nommé ne renvoyant aucune valeur
- Définition
 - `void nomProcedure() { bloc d'instructions }`
- Appel
 - `nomProcedure();`

La fonction

- Bloc d'instructions nommé renvoyant une valeur
- Définition
 - `type nomFonction() { bloc d'instructions }`
- Retour de valeur
 - `return valeur;`
- Appel
 - `type nomVariable = nomFonction();`



Les méthodes et paramètres

Le passage de paramètres

- Définition

- **void** nomProcédure(**type** paramName [, ...**type** paramName])
 { bloc d'instructions }
- **type** nomFonction(**type** paramName [, ...**type** paramName])
 { bloc d'instructions }

- Appel

- nomProcédure(valeur1 [, ...valeur2]);
- **type** nomVariable = nomFonction(valeur1 [, ...valeur2]);

Les outils de débogage

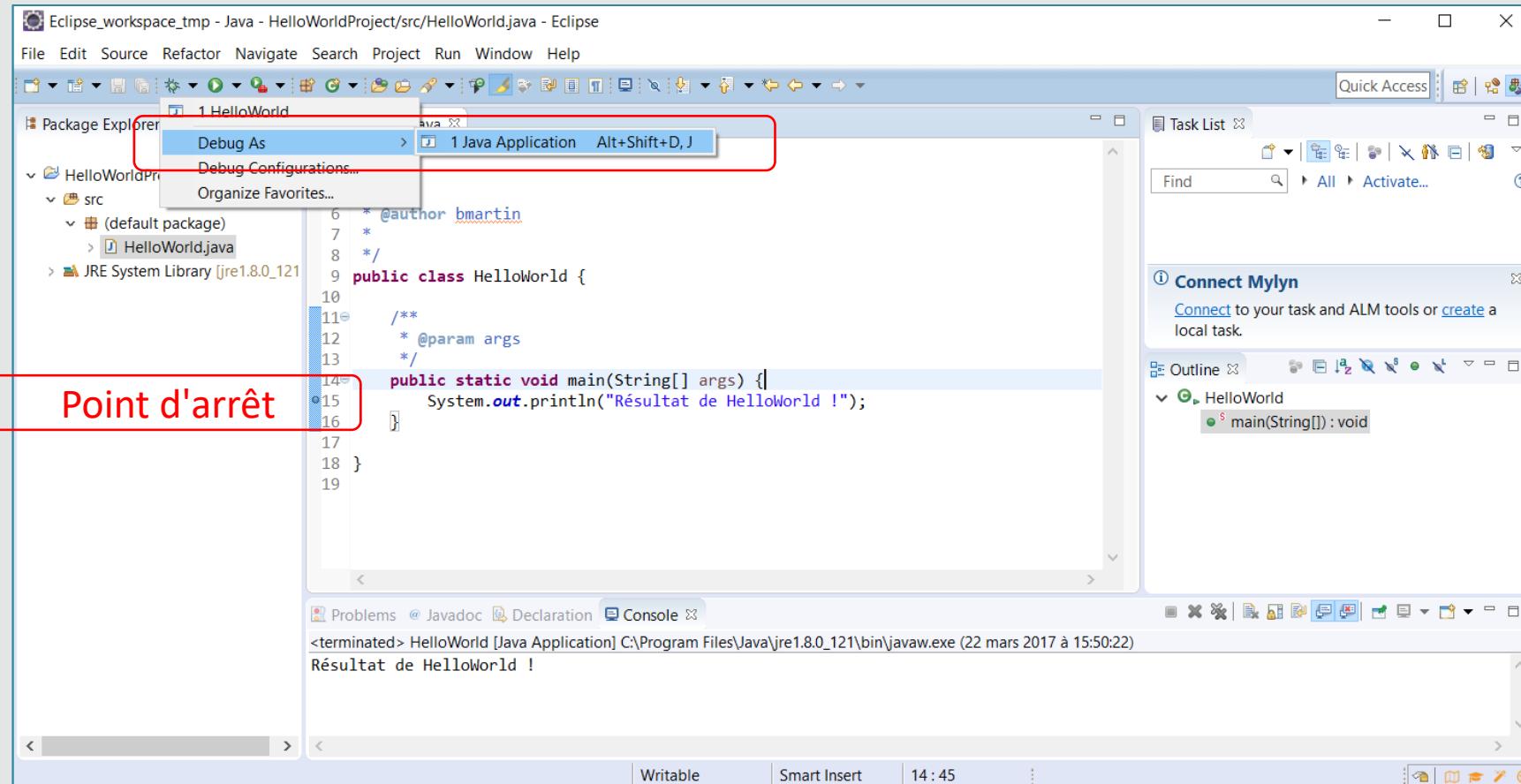
Stack trace

```
Exception in thread "main" java.lang.ArithmetricException: / by zero
    at fr.testormlite.ihm.DemonstrationErreur.fonctionAvecUneDivisionParZero(DemonstrationErreur.java:15)
    at fr.testormlite.ihm.DemonstrationErreur.main(DemonstrationErreur.java:6)
```

- Première ligne : Type d'erreur et message de l'erreur
- Lignes suivantes : L'emplacement de l'erreur

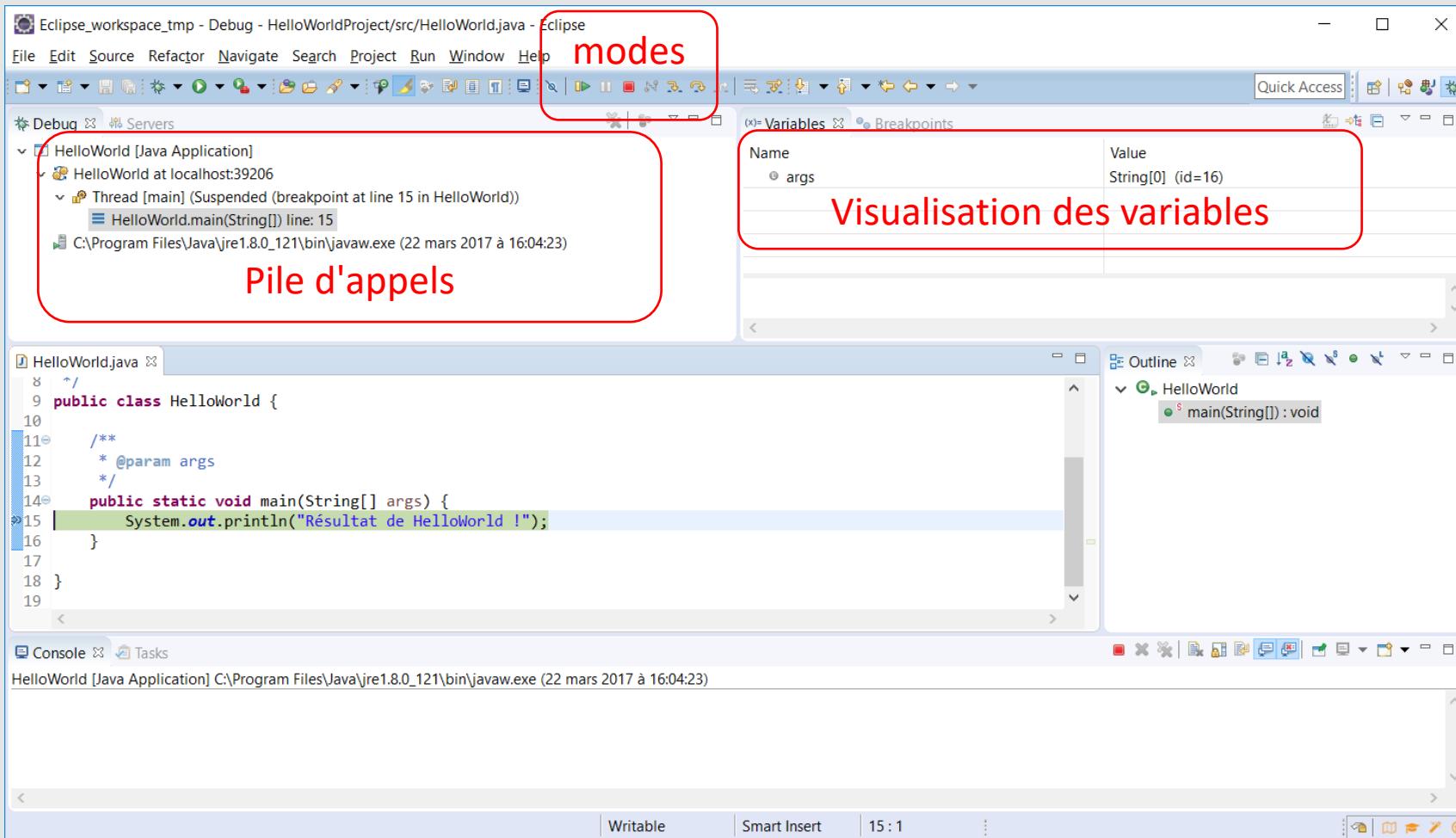
Les outils de débogage

Le debugger



Les outils de débogage

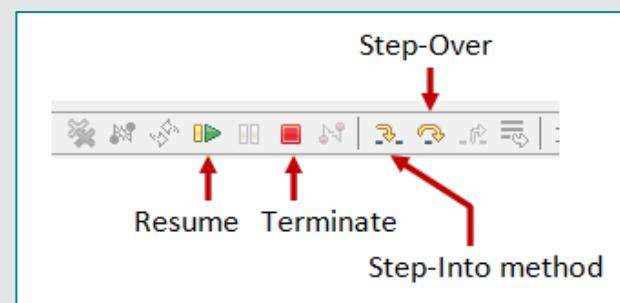
Le debugger



Les outils de débogage

Le debugger

- Resume (F8) : exécution jusqu'au prochain point d'arrêt
- Step over (F6) : pas à pas principal



Les tableaux

- Déclaration

```
type[] nomTableau = new type [TAILLE];
```

- Exemple

```
int[] tableauEntiers = new int [10];
```

- Index débute à 0
- *nomTableau.length* : fournit la taille du tableau

Les boucles Pour (foreach)

- Parcourir le tableau, élément par élément, du début jusqu'à la fin

```
int[] tableauDeInt = {1,2,3,5,8,13};  
  
for(int element : tableauDeInt)  
{  
    System.out.println(element);  
}
```

Les chaînes de caractères

- Type référence : String
- Principales fonctionnalités sur une chaîne (accessible par variable.)
 - charAt
 - equals
 - length
 - replace
 - substring
 - toCharArray
 - toLowerCase
 - trim

Les chaînes de caractères

- Comparaison entre deux chaînes de caractères :
 - `chaine1.equals(chaine2)`
 - **Bannir l'opérateur ==**

Les principaux caractères spéciaux

- \' apostrophe
- \" guillemet
- \\ anti slash
- \t tabulation
- \b retour arrière
- \r retour chariot

Les conversions de types

- Implicite

```
int unEntier = 10;  
long unLong = unEntier;  
System.out.println("unLong : " + unLong);
```

- Explicite

```
double undouble = 10.23;  
int unEntier = (int)undouble;  
System.out.println("unEntier : " + unEntier);
```

Les variables complexes

Les conversions de types

- String.valueOf()

```
String chaîne = String.valueOf("10.23");
System.out.println("chaîne : " + chaîne);
```

- Wrappers et méthodes parseXxxx(String value)

```
int unEntier = Integer.parseInt("10");
System.out.println("unEntier : " + unEntier);
```

Les conversions de types

- Autoboxing

```
Character ch = 'a';
```

- Unboxing

```
Double[] ld = new Double[1];  
ld[1] = 3.1416; //Autoboxing
```

```
double pi = ld[1]; //Unboxing
```

Les variables complexes

Les énumérations

- Définition

```
public enum Jours
{
    LUNDI,
    MARDI,
    MERCREDI,
    JEUDI,
    VENDREDI,
    SAMEDI,
    DIMANCHE
}
```

- Déclaration

```
Jours jour = Jours.JEUDI;
```

Les variables complexes

Les énumérations

- Utilisation

```
for(Jours j : Jours.values())
{
    System.out.println(j.name() + " " + j.ordinal());
}

/*
     LUNDI 0
     MARDI 1
     MERCREDI 2
     JEUDI 3
     VENDREDI 4
     SAMEDI 5
     DIMANCHE 6
*/
```