

Le Langage Java

1^{re} année

J. Beleho (bej) C. Leruste (clr) M. Codutti (mcd)
P. Bettens (pbt) F. Servais (srv) C. Leignel (clg)
D.P. Bishop (bis) S. Drobisz (sdr)

Haute École de Bruxelles-Brabant — École Supérieure d'Informatique

Année académique 2016 / 2017

Séance 2

Développer en Java, premier survol

- La machine virtuelle
- Les outils de développement
- Structure générale
- d'un programme
- Commentaires
- Exemples
- L'erreur est humaine

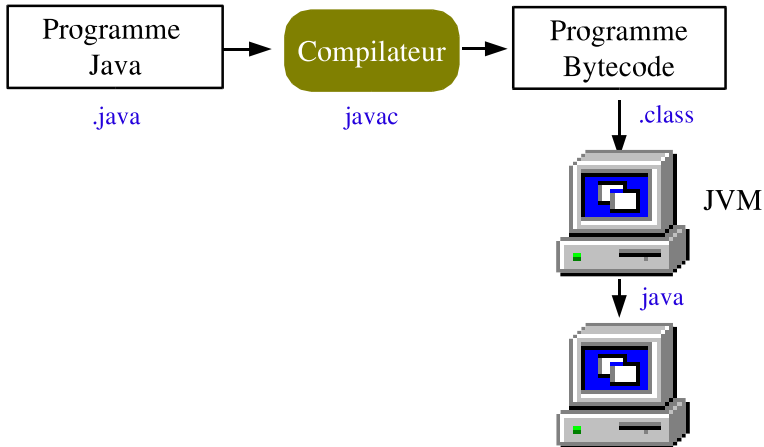
*« There are two ways to write error-free programs ;
only the third one works. » Alan J. Perlis*

Java est **compilé** puis **interprété**.

L'interpréteur Java est la machine virtuelle (JVM)

Le langage de bas niveau interprété par la JVM est le
bytecode

La machine virtuelle



Avantages et inconvénients



Exemple : premier programme

Prenons un exemple (*fichier `Hello.java`*)

```
// Mon premier programme
public class Hello {
    public static void main(String[] args) {
        System.out.println ("Bonjour!");
    }
}
```

Compilons-le

```
$ javac Hello.java
```

On obtient la version compilée, le bytecode
(`Hello.class`) qui peut être exécuté

```
$ java Hello
```

Bonjour !

Fourbir ses armes



Les outils de développement

Les éditions de Java

- ▶ **Java SE** (édition standard)
- ▶ **Java ME** (édition mobile - plus léger)
- ▶ **Java EE** (édition entreprise - plus complet)

Où trouver `javac` et `java` ?

JRE (*Java Runtime Environment*)

JDK (*Java Development Kit*)



Éditer
Compiler
Exécuter

Les outils de développement

1

- ▶ Un **éditeur avec coloration syntaxique** : gVim, Notepad++, nano, ...
- ▶ Gestion manuelle des noms et emplacements des fichiers
- ▶ Compilation et exécution en ligne de commande

Les outils de développement

2

- ▶ Un *Environnement de Développement Intégré* :
Netbeans, Eclipse, ...
- ▶ Intègre tout le processus de développement



credit photo

Structure générale du programme

```
$cat NomClasse.java
```

```
public class NomClasse {  
    // insert code here  
}
```

Attention Java est sensible à la casse

La méthode principale

\$cat NomClasse.java

```
public class NomClasse {  
    public static void main(String[] args) {  
        // insert code here  
    }  
}
```

Exemple

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

Le commentaire

Plusieurs manières d'ajouter un commentaire

```
// Commentaire sur une ligne  
/* Commentaire sur  
   plusieurs lignes */
```

La méthode principale

Exemples

```
public class Exercice1 {  
    public static void main(String[] args) {  
        System.out.println (3 + 4);    // Affiche la valeur 7  
    }  
}
```

```
public class Exercice2 {  
    /* Affiche la chaine 3 + 4 */  
    public static void main(String[] args) {  
        System.out.println ("3_+_4");  
    }  
}
```


Math.random()

```
public class Exercice3 {  
    public static void main(String[] args) {  
        /*  
        * Affiche un nombre réel au hasard entre 0 inclus  
        * et 1 exclu  
        */  
        System.out.println (Math.random());  
    }  
}
```

Math.random()

```
public class Exercice4 {  
    public static void main(String[] args) {  
        // Affiche un nombre réel au hasard entre 0 inclus et 10  
        System.out.println (Math.random()*10);  
    }  
}
```

```
public class Exercice5 {  
    public static void main(String[] args) {  
        // Affiche un nombre réel au hasard entre 1 inclus et 11 exclu  
        System.out.println (Math.random()*10 + 1);  
    }  
}
```

Processus d'écriture d'un programme

Édition / compilation / exécution

... *et tout va bien*

Quels types d'erreurs ?



Crédit photo

Quels types d'erreurs ?

Quels types d'erreurs ?

- ▶ Les erreurs de **compilation**
- ▶ Les erreurs d'**exécution**

```
public Class Hello{  
    public static void main (string[] args){  
        System.out.println("Hello");  
    }  
}
```

```
$ javac Hello.java  
Hello.java:1: class, interface, or enum  
expected  
public Class Hello  
      ^
```

Les erreurs de compilation



- ▶ Compiler souvent
- ▶ Apprendre à reconnaître **rapidement** les erreurs fréquentes
- ▶ Lire / comprendre les messages du compilateur

```
public class Division{  
    ...  
}
```

```
$ javac Division.java
```

```
$ java Division
```

```
Exception in thread "main"
```

```
java.lang.ArithmeticException: / by zero  
at Division.main(Division.java:7)
```


Les erreurs d'exécution



- ▶ Apprendre à reconnaître **rapidement** les erreurs fréquentes
- ▶ **Déboguer** son code ; la méthode de l'homme pauvre et/ou le débogueur
- ▶ Mettre des tests en œuvre

Crédits

Ces slides sont le support pour la présentation orale de l'unité d'enseignement **DEV1-JAV** à HE2B-ÉSI

Crédits

Les distributions Ubuntu et/ou debian
du système d'exploitation **GNU Linux**.

LaTeX/Beamer comme système d'édition.

Git et GitHub pour la gestion des versions et le suivi.

GNU make, **rubber**, **pdfnup**, ... pour les petites tâches.

Images et icônes

deviantart, flickr, The Noun Project       

