

Le Langage Java

1^e année

J. Beleho (bej) C. Leruste (clr) M. Codutti (mcd)
P. Bettens (pbt) F. Servais (srv) C. Leignel (clg)
D. Nabet (dna) J. Lechien (jlc)

Haute École de Bruxelles — École Supérieure d'Informatique

Année académique 2014 / 2015

Liste des séances

- 1 Objectifs, évaluations et introduction
- 2 Développer en Java, premier survol
- 3 La gestion des erreurs et le survol des alternatives

Séance 1

Objectifs, évaluations et introduction

- Objectifs
- Moyens
- Évaluations
- Concepts
- Traduction

« *I really hate this darn machine ; I wish that they would sell it.
It won't do what I want it to, but only what I tell it.* »
Anonyme

Notre objectif ...

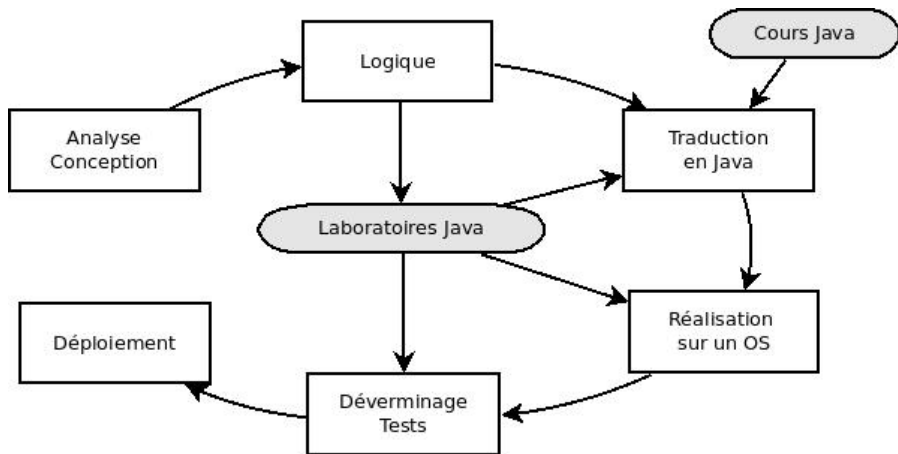
... votre objectif



Objectifs du cours

- ▶ initiation à la programmation
- ▶ apprentissage de bons comportements
- ▶ implémentation sur un OS (*operating system*)

Liens avec les autres cours



Supports et ressources

Rien

- ▶ pas de syllabus ;
- ▶ pas de livre ;

Quoique

- ▶ les slides sur github ;
- ▶ des liens, des documents, ... sur poÉSI
- ▶ un forum de discussion, (fora)




Évaluation à quelle sauce ?

Crédit photo

Évaluation

Évaluation de l'unité d'enseignement, une cote pour toutes les activités d'apprentissage :

ALG - JAV - LAJ



programmer

[proh-gram-er]

an organism that turn caffeine into software

geek.

Définitions

Définissons les concepts suivants :

- ▶ Un **programme** ?
- ▶ **Programmer** ?
- ▶ Un **langage de programmation** ?
- ▶ Différence entre **langue** et *langage* ?

Un programme

La seule chose dont est capable un ordinateur est de réaliser extrêmement rapidement des instructions élémentaires

Toute tâche qu'on veut lui confier doit donc être préalablement décrite comme une suite séquentielle d'instructions (un programme)

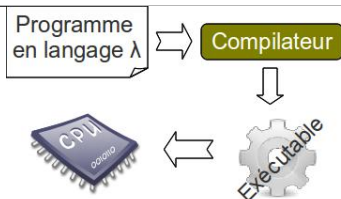
Un langage

Une classe de langages est adaptée à une classe de problèmes ... et ces problèmes évoluent dans le temps ...

Le problème de la traduction

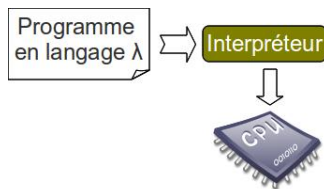
Un ordinateur ne comprend que le langage machine.
Nécessité d'une **traduction**

Compilation



*traduit d'une traite
avant l'exécution*

Interprétation



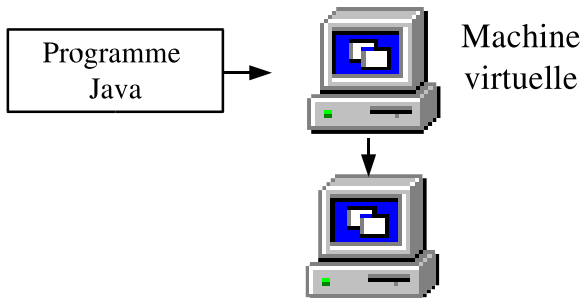
*traduit morceau par morceau
au moment de l'exécution*

Et Java ?

Compilé ou interprété ?

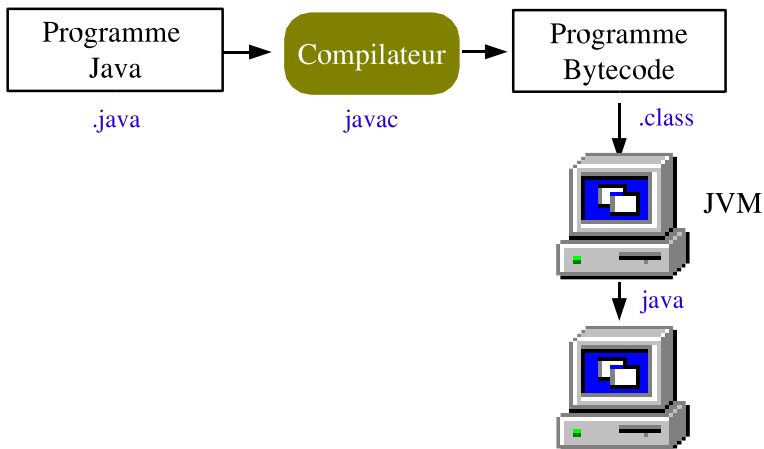
La machine virtuelle

Java a une approche mixte la **machine virtuelle** Java (JVM)



D'abord compilé
ensuite interprété

La machine virtuelle



timeline Java



Crédit photo

Historique de Java

- 92 SUN crée oak (systèmes embarqués).
Auteur : James Gosling
- 94 Adapté à Internet grâce aux **applets**.
Devient Java
- 96 Première version stable et gratuite de JDK
- 98 Sortie de Java 2
- 05 Version 1.5 de Java 2
- 09 Oracle rachète Sun (et donc Java)
- 11 Version 1.7 (Java 7, en GPL)
- 14 Version 1.8 (Java 8)



- ▶ Java est-il installé sur ma machine ?
- ▶ Puis-je commencer à écrire un programme Java ?
- ▶ Qu'ais-je pris comme note ?

Séance 2

Développer en Java, premier survol

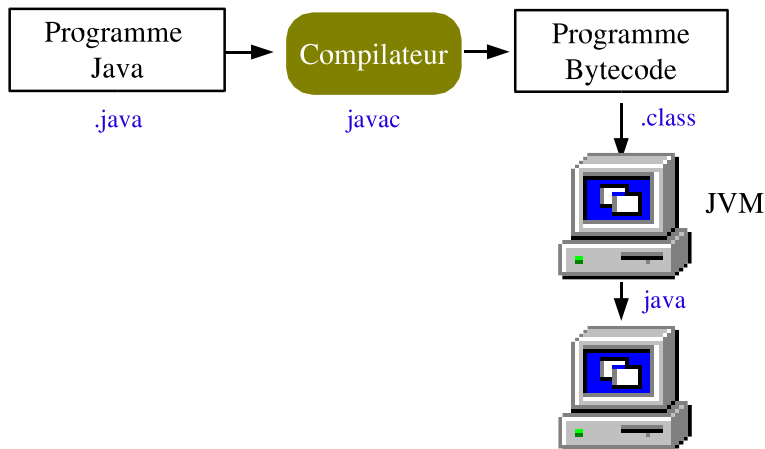
- La machine virtuelle
- Les outils de développement
- Algorithmes séquentiels (survol)
- Structure générale d'un programme
- Constantes
- Conventions
- Commentaires

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](#))

On peut l'exécuter \$ 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  
    }  
}
```

Les variables

Les types disponibles

En Logique	En Java
Entier	int
Réel	double
Chaine	String
Caractère	char
Booléen	boolean

Exemple de déclaration

```
int nb1;
```

L'assignation et les calculs

L'assignation se fait via le symbole

=

```
nb1 = 1;
```

Opérateurs :

+ - * / %

Exemple

\$cat Moyenne.java

```
public class Moyenne {  
    public static void main(String[] args) {  
  
        double nombre1;  
        double nombre2;  
        double moyenne;  
  
        nombre1 = 34345;  
        nombre2 = -3213213;  
        moyenne = (nombre1 + nombre2) / 2;  
        System.out.println(moyenne);  
    }  
}
```

Exemple

\$cat Moyenne.java

```
public class Moyenne {  
    public static void main(String[] args) {  
  
        int nombre1 = 34345;  
        int nombre2 = -321321;  
        double moyenne;  
  
        // division réelle car un des 2 opérandes est réel  
        moyenne = (nombre1 + nombre2) / 2.0;  
        System.out.println ("La moyenne est " + moyenne);  
    }  
}
```

Lire au clavier

Les applications modernes préfèrent les lectures dans des champs de saisies.

Voici une manière de faire

Exemple

```
import java.util.Scanner;  
// ...  
Scanner clavier = new Scanner(System.in);  
// ...  
nombre1 = clavier.nextInt ();
```

Lire au clavier - Exemple

```
$cat Test.java
```

```
import java.util.Scanner;

public class Test {
    public static void main(String[] args) {
        Scanner clavier = new Scanner(System.in);
        double nombre1;
        double nombre2;
        double moyenne;

        nombre1 = clavier.nextDouble();
        nombre2 = clavier.nextDouble();
        moyenne = (nombre1 + nombre2) / 2.0;
        System.out.println (moyenne);
    }
}
```


Lire au clavier

Pour lire...	on écrit...
un entier	<code>nextInt()</code>
un réel	<code>nextDouble()</code>
un booléen	<code>nextBoolean()</code>
un mot	<code>next()</code>
une ligne	<code>nextLine()</code>
un caractère	<code>next().charAt(0)</code>

Constante locale

Une **constante** s'écrit grâce à **final**

Exemple

```
final int X = 1;  
final int Y;  
Y = 2*X;  
X = 2; // Erreur : possède déjà une valeur  
Y = 3; // Idem
```

Handwritten text in a cursive script, likely a historical form of Hebrew or Aramaic, arranged in four rows. Each row consists of a main body of text, a smaller column of text to the right, and a single character 'I' further to the right.

Conventions d'écriture

Printed text in a bold, blocky font, likely a historical form of Hebrew or Aramaic, arranged in one row. It consists of a main body of text, a smaller column of text to the right, and a single character 'I' further to the right.

Handwritten text in a cursive script, likely a historical form of Hebrew or Aramaic, arranged in one row. It consists of a main body of text, a smaller column of text to the right, and a single character 'I' further to the right.

Handwritten text in a cursive script, likely a historical form of Hebrew or Aramaic, arranged in one row. It consists of a main body of text, a smaller column of text to the right, and a single character 'I' further to the right.

Le commentaire

Plusieurs manières d'ajouter un commentaire

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

Séance 3

La gestion des erreurs et le survol des alternatives

- L'erreur est humaine
- Alternatives (survol)

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

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 le débogueur
- ▶ Mettre des tests en œuvre



Instructions de choix

Le **Si**

```
if ( condition ) {  
    instructions  
}
```

Le **Si-sinon**

```
if ( condition ) {  
    instructions  
} else {  
    instructions  
}
```

Exemple

```
import java.util.Scanner;
public class Test {
    public static void main(String[] args) {
        Scanner clavier = new Scanner(System.in);
        int nombre1;

        nombre1 = clavier.nextInt();
        if (nombre1 < 0) {
            System.out.println(nombre1 + " est négatif");
        }
    }
}
```

Exemple

```
import java.util.Scanner;
public class Test {
    public static void main(String[] args) {
        Scanner clavier = new Scanner(System.in);
        int nombre1;

        nombre1 = clavier.nextInt();
        System.out.println(nombre1 + " est un nombre ");
        if (nombre1 < 0) {
            System.out.println(" négatif ");
        } else {
            System.out.println(" positif ");
        }
    }
}
```


Exercice

Comment traduire cet algorithme?

```
MODULE test ()  
  nombre1: Entier  
  LIRE nombre1  
  SI nombre1 > 0 ALORS  
    ECRIRE nombre1, "est positif"  
  SINON  
    SI nombre1 = 0 ALORS  
      ECRIRE nombre1, "est nul"  
    SINON  
      ECRIRE nombre1, "est négatif"  
    FIN SI  
  FIN SI  
FIN MODULE
```

Expressions booléennes

Les comparateurs

< > <= >= == !=

Les opérateurs booléens

&& (et) || (ou) ! (non)

Exemple

```
import java.util.Scanner;
public class Exemple {
    public static void main(String[] args) {
        Scanner clavier = new Scanner(System.in);
        int nombre1;

        nombre1 = clavier.nextInt();
        if ((nombre1 % 2) == 0) {
            System.out.println("Le nombre est pair");
        } else {
            System.out.println("Le nombre est impair");
        }
    }
}
```

Exemple

```
import java.util.Scanner;
public class Exemple {
    public static void main(String[] args) {
        Scanner clavier = new Scanner(System.in);
        int âge;

        âge = clavier.nextInt();
        if ( âge<21 || âge>=60 ) {
            System.out.println ("Tarif réduit !");
        }
    }
}
```

Le « selon-que »

Première forme

```
switch(produit) {  
    case "Coca" :  
    case "Sprite" :  
    case "Fanta" :    prixDistributeur =60; break;  
    case "IceTea" :    prixDistributeur =70; break;  
    case "Aquarius" :  
    case "Nalu" :    prixDistributeur =80; break;  
    default :    prixDistributeur =0 break;  
}
```

- ▶ Notez le **break**
- ▶ Possible avec : entiers, caractères et chaînes

Le « selon-que »

Deuxième forme : la logique suivante

Selon que

nb > 0 : Écrire " positif "

nb = 0 : Écrire "nul"

autres : Écrire " négatif "

Fin selon que

s'écrit en Java

```
if (nb>0) {  
    System.out. println ( " positif " );  
} else if (nb==0) {  
    System.out. println ( "nul" );  
} else {  
    System.out. println ( " négatif " );  
}
```

Crédits

Ce document a été produit avec les outils suivants

- ▶ Les distributions **Ubuntu** et/ou **debian** du système d'exploitation **Linux**
- ▶ **LaTeX/Beamer** comme système d'édition
- ▶ **Git** et **GitHub** pour la gestion des versions et le suivi des corrections
- ▶ Les outils **make**, **rubber**, **pdftup**, ...

Il est proposé sous licence

Creative Commons Paternité - Partage à l'Identique 2.0 Belgique
<http://creativecommons.org/licenses/by-sa/2.0/be/>