

# Introduction au langage Java

... de C à Java

University of Geneva  
www.miralab.ch

## Caractéristique de Java

- Architecture neutre, portable
- Orienté objet
- Distribué
- Compilé et interprété !
- Robuste
- Dynamique
- Sécurisé  
(gestion des exceptions)

### *Pour rappel C :*

Langage structuré

*Organisation en bloc*

Langage déclaratif

*Les éléments doivent être déclarés*

Format libre

*Liberté de mise en page*

Langage modulaire

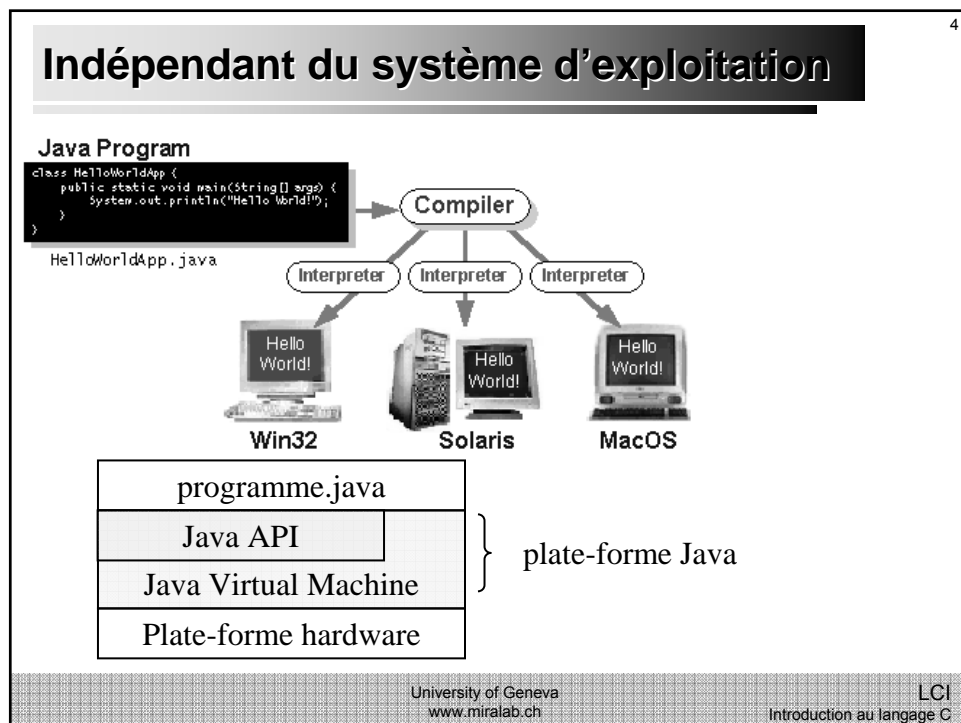
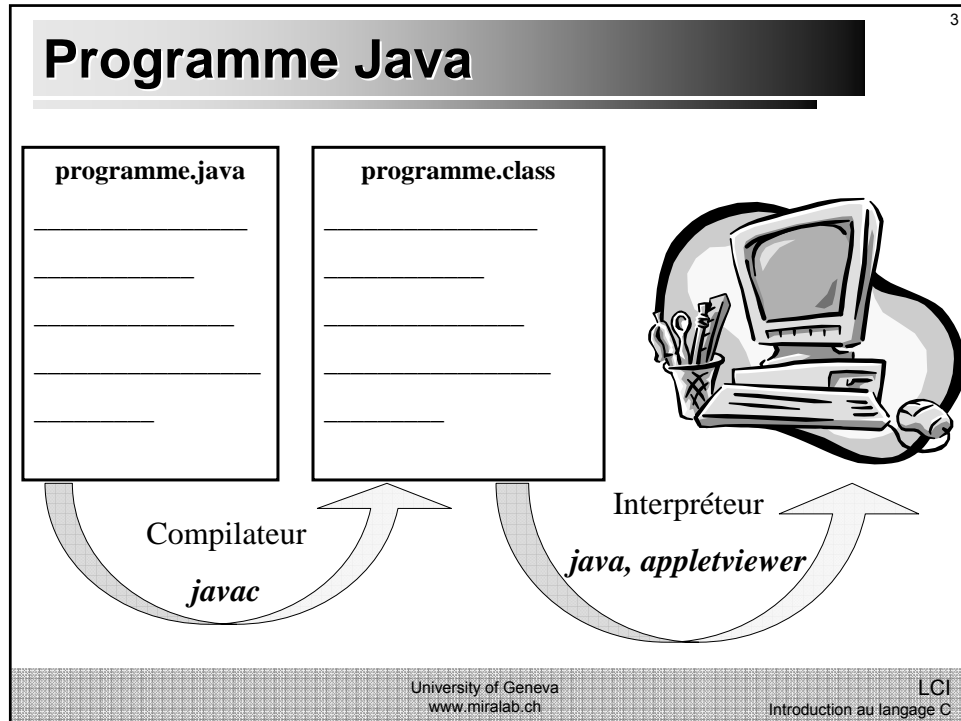
*Regroupement des fonctions par module*

Langage compilé

*Modules peuvent être compilé séparément*

University of Geneva  
www.miralab.ch

LCI  
Introduction au langage C



## Limitation de Java

5

- Moins rapide qu'un « vrai » exécutable (C)
  - L'interpréteur prend du temps pour exécuter le codes des fichiers *.class*
  
- Solutions
  - Optimiser les **Machines Virtuelles Java**
  - Compilateurs JIT (*Just In Time*)
    - Quand une classe est chargée, elle est traduite dans le code machine de la plate-forme où elle est exécutée
    - Cette opération ralentit le départ de l'application, mais ensuite l'exécution du programme est beaucoup plus rapide !

## 2 types de programmes : Application & Applet

6

### Application

Programme indépendant

Point d'entrée du programme

```
public static void main(String [] args)
```

### Applet

Programme téléchargeable

Exécuté automatiquement quand ils sont  
intégrés à l'intérieur de pages HTML

## Commentaire

7

### Identique au C++

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

### Javadoc

```
/** description pour création  
Automatique de documentation */
```

## Afficher un texte

8

### Rappel en C++

```
cout << "Hello World!" << endl;
```

### En Java

```
System.out.println("Hello World!");
```

## 1<sup>er</sup> exemple d'application

```
// fichier HelloWorld.java
class HelloWorld
{
    static public void main(String args[])
    {
        System.out.println("Hello World!");
    }
}
```

Fichier au format ASCII (notepad ou autre environnement de développement)  
⇒ HelloWorld.java

**ATTENTION : même nom que la class main (case sensitive)**

University of Geneva  
www.miralab.ch

LCI  
Introduction au langage C

## Compilation de mon exemple

Compilateur = javac

-g ou -g:none compile avec ou sans infos de debug

-classpath <dir>

<dir> répertoires d'accès aux classes importées  
séparés par ; sous Windows, : sous UNIX

-d <dir>

<dir> spécifie le répertoire de destination des fichiers de  
classes .class

La variable d'environnement CLASSPATH est utilisée  
comme équivalent de l'option -classpath

javac HelloWorld.java

↙ Compilation et résultat dans le même repertoire

University of Geneva  
www.miralab.ch

LCI  
Introduction au langage C

## Exécution de mon exemple

11

Exécution = java

-classpath <dir>

<dir> répertoires d'accès aux classes importées

-help

donne la liste des options

-version

écrit la version de la Machine Virtuelle que vous utilisez

La variable d'environnement CLASSPATH est utilisée  
comme équivalent de l'option -classpath

java HelloWorld

```
D:\StefWork_vc7\Java\LCI>javac helloworld.java
D:\StefWork_vc7\Java\LCI>java helloworld
Hello world!
D:\StefWork_vc7\Java\LCI>
```

University of Geneva  
www.miralab.ch

LCI  
Introduction au langage C

## Les types natifs Java (1)

12

### Entiers

<i>byte</i>	signé 8 bits [-128.. 127]	0
<i>short</i>	signé 16 bits [-32768.. 32767]	0
<i>int</i>	signé 32 bits [2147483648.. 2147483647]	0
<i>long</i>	signé 64 bits [-9223372036854775808.. 9223372036854775807]	0L

### Nombres réels

<i>float</i>	32 bits	0.0f
<i>double</i>	64 bits	0.0d

University of Geneva  
www.miralab.ch

LCI  
Introduction au langage C

## Les types natifs Java (2)

13

### Caractère *Unicode*

*char*      16 bits      ['\u0000'.. '\uffff']      '\u0000'

128 premiers = codes ASCII 'a', '1'

'\n' pour un saut de ligne

'\t' pour une tabulation

'\" pour le caractère '

'\"' pour le caractère "

'\\' pour le caractère \

### Booléen

*boolean*      true ou false      false

## Conversion de type

14

### Conversion automatique ou implicite

Moins nombreuse qu'en C

Erreur de compilation pour les conversions avec  
perte de données

```
double d = 0.0d;
```

```
int i = d; // erreur "possible loss of precision"
```

Nécessite une conversion explicite

```
double d = 0.0d;
```

```
int i = (int)d; // ok
```

## Références

15

En Java pas de pointeurs ☺

Une variable est

soit de type natif *byte*, ..., *double*

*char* *c*;

soit une référence désignant un objet

*String* *s*;

Impossible

de créer 1 référence vers un type natif comme  
en C++

University of Geneva  
www.miralab.ch

LCI  
Introduction au langage C

## Les Blocs

16

Bloc : identique au C, C++

```
{
}
```

On peut déclarer les variables à tout instant

Toutes les variables sont locales et temporaires

pas de variables globales

pas de variables statiques

variable automatiquement détruite en dehors de sa  
portée

University of Geneva  
www.miralab.ch

LCI  
Introduction au langage C



## Les tests

17

Test : identique au C

```

if() { }

if() { }
else { }

switch()
{
    case : break;
    default: break;
}

```

Différence

Les expressions à tester doivent être de type boolean

University of Geneva  
www.miralab.ch

LCI  
Introduction au langage C

## Les boucles

18

Identiques au C

```

while() { }

do { }
while();

for( ; ; ) { }

```

University of Geneva  
www.miralab.ch

LCI  
Introduction au langage C

19

	Primaire	Maximum
( ) [ ]		
! ~ + - ++ -- (typecast)	Unaire	
* / %	Multiplication, ...	
+ -	Addition, soustraction	
<< >> >>>	Décalages	
< <= > >=	Comparaisons	
== !=	Équivalence, différence	
&	ET binaire	
^	OU exclusif binaire	
	OU binaire	
&&	ET logique	
	OU logique	
?	Condition	
= *= /= %= += -= <<= >>= >>>= &= ^=  =	Affectation	Minimum

Priorité

University of Geneva  
www.miralab.ch

LCI  
Introduction au langage C

20

Mots clefs C inexistant en Java	
<b>typedef</b>	Remplacé par le concept de classe
<b>#define</b>	Remplacé par définitions <i>static final</i> pour les constantes l'utilisation de méthodes pour les macros
<b>enum</b>	Remplacé par classe avec constantes accessibles <i>static final public</i>
<b>extern</b>	Pas de variable globale !

University of Geneva  
www.miralab.ch

LCI  
Introduction au langage C

21

## Pas de pré compilation

Tout les mots clefs #... sont supprimés

`#include`

`#define`

`#ifdef`

`#endif`

**Le but est de simplifier la programmation  
et la lisibilité des programmes**

22

## L'opérateur new

Sert à créer des instances d'une classe

```
String s = new String();
```

Similaire au *new* de C++

mais **ne** s'applique **pas** aux types natifs en Java  
(sauf pour l'initialisation des tableaux ☺)

## Les tableaux

23

### Déclaration similaire au C

On peut mettre les `[]` avant ou après le nom de la variable

Possibilité de créer des tableaux multidimensionnels  
à dimensions changeantes en fonction de l'index précédant

L'indexage est protégé

Dépassement génère une **exception** *erreur générée à l'exécution*

Pas d'accès fortuit à de la mémoire hors du tableau

University of Geneva  
www.miralab.ch

LCI  
Introduction au langage C

## Exemple : tableaux

24

```
//Déclaration
int [] tInt;
float tFloat [];
//Création
int [] tInt2 = new int [4];
double tDouble[][] = new double[2][3];
//Création avec initialisation
long [] tLong = {3L, 2L};
short [][] tShort = {{4, 5}, {1, 2, 3},
                    null}; //Dimension
int [] vide = {};
```

University of Geneva  
www.miralab.ch

LCI  
Introduction au langage C

## Les classes de base ...

University of Geneva  
www.miralab.ch

## Les chaînes de caractères

26

Deux types (ou classes)

Pour les constantes *String*

Pour les variables *StringBuffer*

Les types natifs peuvent être convertis

```
String s= "total = " +3 + " CHF";
```

Retrouver la longueur d'une chaîne

```
int lg = s.length();
```

University of Geneva  
www.miralab.ch

LCI  
Introduction au langage C

## Les chaînes de caractères

27

### Accès en lecture (*String* ou *StringBuffer*)

à un caractère donné de la chaîne

```
.charAt(int index);
    String s = "total";
    char c = s.charAt(1); // c = 'o'
```

au tableau de caractère donné de la chaîne

```
.toArray(); //ou .toString().toArray();
    char [] ctab = s.toArray(); //ctab[]={'t',...}
.getChars(int deb, int fin,
          char [] dst, int dstdeb);
char [] ctab = new char [s.length()];
s.getChars(0, s.length(), ctab, 0); //ctab[]={'t',...}
```

## Les chaînes de caractères

28

### Accès en écriture (*StringBuffer*)

à un caractère donné de la chaîne

```
.setCharAt(int index, char c);
    StringBuffer s = new StringBuffer("total");
s.setCharAt(2, 'n'); // s = "tonal"
```

## La classe System et les E/S

29

Cette classe permet d'accéder à la console  
pour les entrées et les sorties avec  
*out*, *err* pour la sortie standard et d'erreur  
*in* pour l'entrée standard

### Rappel

```
System.out.println("Hello");
```

```
import java.io.*; //similaire #include<iostream>
public class LireLigneClavier
{
    public static void main(String[] args)
        throws java.io.IOException
    {
        BufferedReader inr = new BufferedReader(new
            InputStreamReader(System.in));
        System.out.print("tapez un nombre: ");
        String s = inr.readLine();
        int a = Integer.parseInt(s);
        System.out.println(a*a); // imprime a au carré
    }
}
```

30

## Pas de fonction en Java !!!!!

31

En Java il n'y a plus de fonction ou  
procédure

⇒ Elles sont remplacées par les classes et  
leurs méthodes (prochain cours)

## Résumé

32

Caractéristiques de Java

Les types natifs

Conversion de type

Les tableaux

Point d'entrée du programme

Différences avec le C

Références

<http://cui.unige.ch/java/>

<http://www.developer.org/club/bkostrzewa/>

<http://java.sun.com/docs/white/langenv/>

<http://java.sun.com/docs/books/tutorial/getStarted/index.html>

Les chaînes de caractères

*String & StringBuffer*



## Exercice 1 Java

33

Faire un programme qui calcule la moyenne de N notes saisies au clavier.

(la somme des notes sera calculée au fur et à mesure de la saisie)

## Exercice 1 : pas à pas

34

Déclaration des variables

N = nombre de notes à saisir  
i = compteur de 1 à N  
somme = cumul de la somme  
note = note saisie par l'utilisateur  
moyenne = résultat

Algorithme

somme  $\leftarrow$  0  
saisir N  
pour i = 1 à N  
  saisir note  
  somme  $\leftarrow$  somme + note  
moyenne  $\leftarrow$  somme / N  
afficher (moyenne)

35

```

import java.io.*;

class sommeNombre {

    public static void main(String args[])
        throws java.io.IOException
    {
        BufferedReader inr = new BufferedReader(new InputStreamReader(System.in));

        System.out.println("Nb de notes a saisir : ");
        String s = inr.readLine();
        int nbNotes = Integer.parseInt(s);

        int somme = 0;
        int i;
        for (i=0; i<nbNotes; i++) {
            s = inr.readLine();
            somme += Integer.parseInt(s);
        }
        System.out.println("La somme est = " + somme);
    }
}

```

36

## Exercice 2 (Java)

Faire un programme simulant une calculatrice 4 opérations (saisir un nombre, une opération, et un nombre, puis afficher le résultat). = pour terminer le programme.

*Exemple :*

$2 * 5 \rightarrow 10$

$4 = 6 \rightarrow \text{fin du programme}$

37

## Exercice 2 : pas à pas

1. Saisir les données (valeurs et opérateurs)
2. Faire les calculs
3. Afficher le résultat

### Déclaration des variables

float val1 et val2 = les 2 valeurs courantes de l'opération  
 float res = résultat de l'opération  
 char op = opération  
 bool fin = fin des calculs ?

38

## Exercice 2 : pas à pas (suite)

```

fin ← faux
faire {
  saisir val1, op et val2
  suivant op
  {
    si op = '*' alors    res ← val1 * val2
    si op = '/' alors    res ← val1 / val2
    si op = '+' alors    res ← val1 + val2
    si op = '-' alors    res ← val1 - val2
    si op = '=' alors    fin ← vrai
  }
  fin suivant
  afficher res
} tant que (non fin)
  
```

```

import java.io.*;

class calculette {

public static void main(String args[]) throws java.io.IOException
{
    int val1, val2, res = 0;
    char oper;
    boolean fin = false;

    BufferedReader inr = new BufferedReader(new InputStreamReader(System.in));
    String sval1, soper, sval2;

    do
    {
        System.out.println("NOMBRE OPERATEUR NOMBRE, par exemple 5 * 2");
        System.out.println("Operateur = pour finir\n");
        System.out.println("Calcul: ");
        sval1 = inr.readLine();
        soper = inr.readLine();
        sval2 = inr.readLine();

        val1 = Integer.parseInt(sval1);
        val2 = Integer.parseInt(sval2);
        oper = soper.charAt(0);

        switch (oper) {
            case '*': res=val1*val2; break;
            case '/': res=val1/val2; break;
            case '+': res=val1+val2; break;
            case '-': res=val1-val2; break;
            case '=': fin=true;
        }
        if (fin == false) {
            System.out.println(val1 + " " + oper + " " + val2 + " = " + res);
        }
    } while (fin != true);
} // main
}

```

39