

Java

Pourquoi JAVA ?

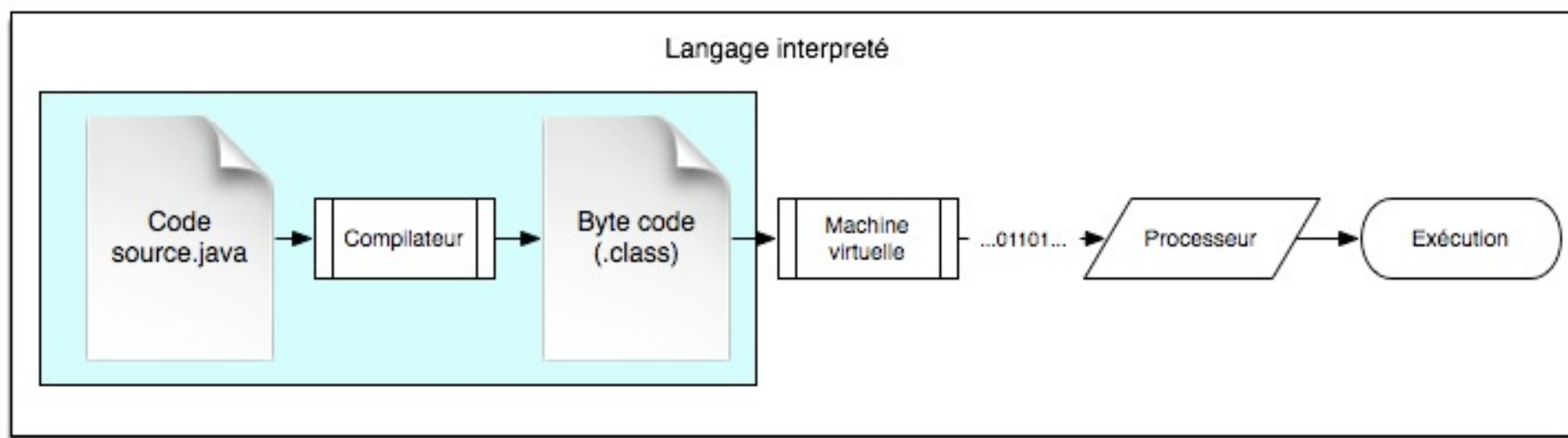
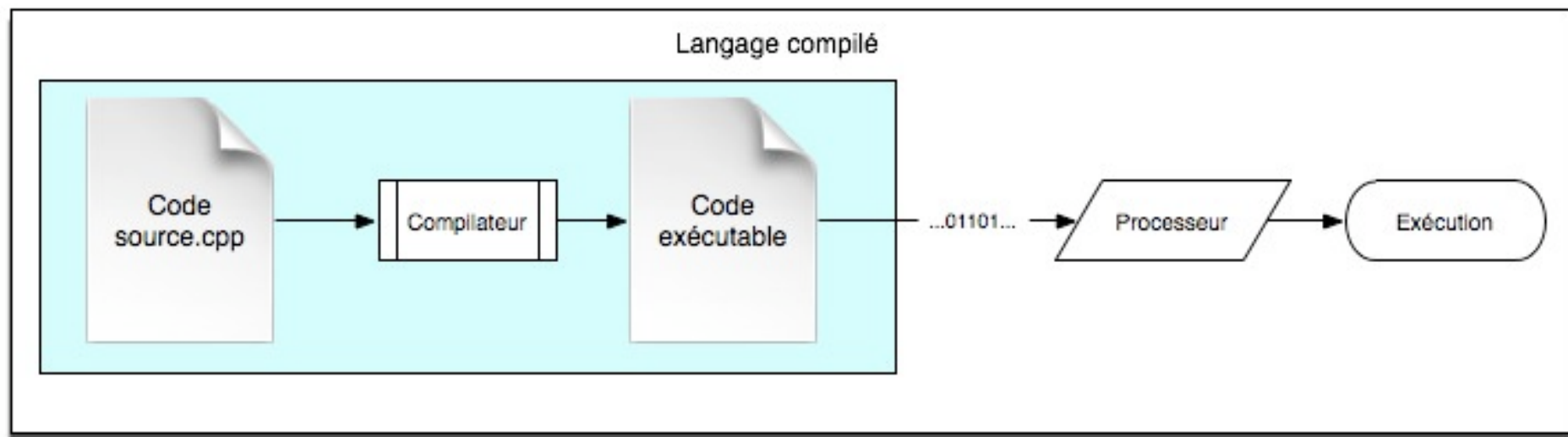
- 1994
- Premièrement développé par Sun, racheté par Oracle
- Intérêt:
 - Plate-forme portable
 - Applet / Web / Desktop / Mobile / Console / Serveur

JVM

- Couche d'abstraction entre le bytecode et la machine
- 3 notions de JVM:
 - Les spécifications
 - L'implémentation
 - L'instance

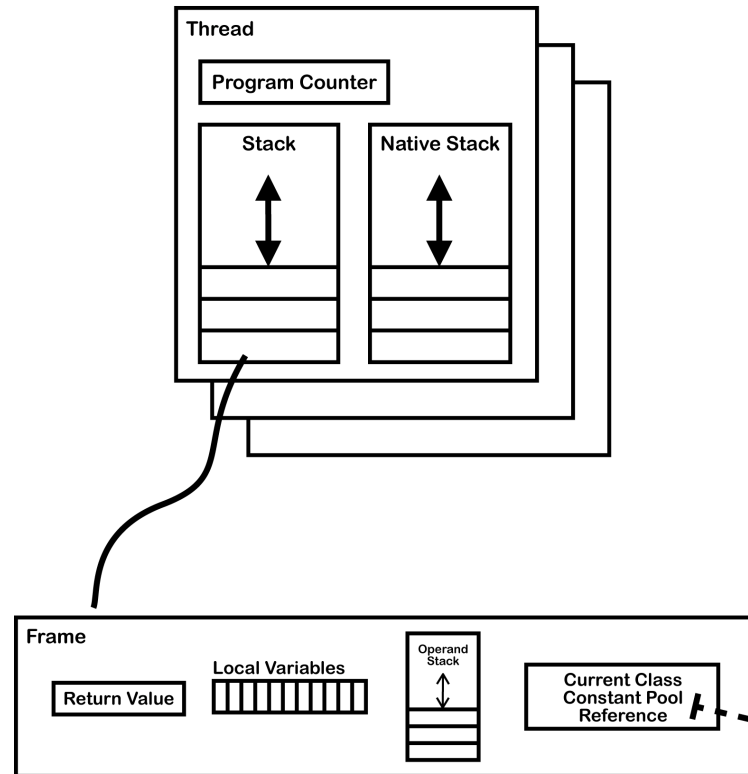


Etapes effectuées à l'avance

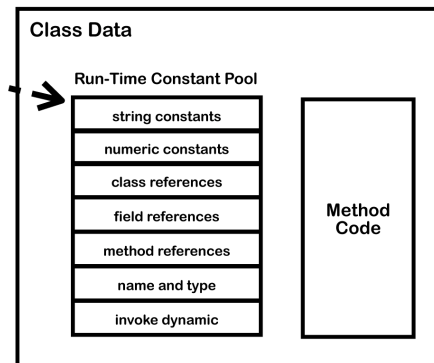
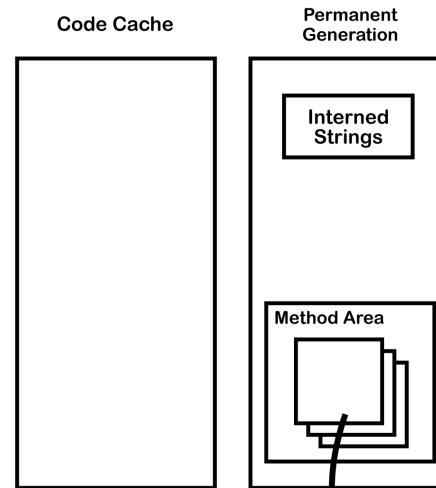


JVM

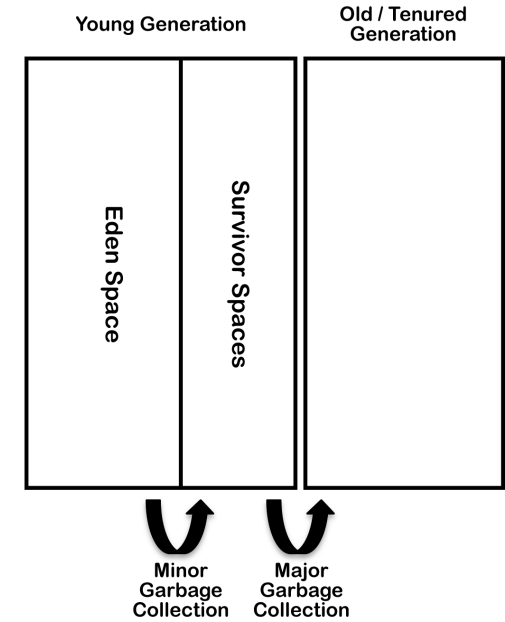
Stack



Non Heap



Heap

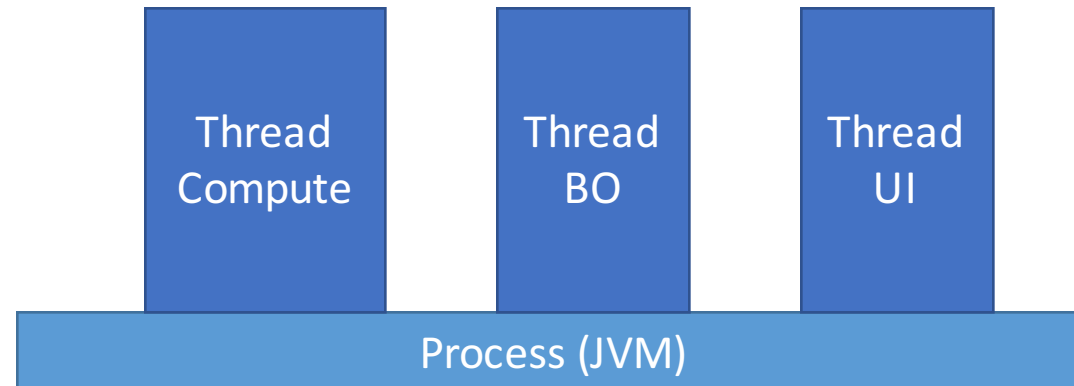


Classe compilée

- `ClassFile {`
 - `u4 magic;`
 - `u2 minor_version;`
 - `u2 major_version;`
 - `u2 constant_pool_count;`
 - `cp_info constant_pool[constant_pool_count - 1];`
 - `u2 access_flags;`
 - `u2 this_class;`
 - `u2 super_class;`
 - `u2 interfaces_count;`
 - `u2 interfaces[interfaces_count];`
 - `u2 fields_count;`
 - `field_info fields[fields_count];`
 - `u2 methods_count;`
 - `method_info methods[methods_count];`
 - `u2 attributes_count;`
 - `attribute_info attributes[attributes_count];``}`

Process Vs Thread

- Process :
 - Environnement d'exécution complet
 - Espace mémoire dédié
- Thread :
 - « Lightweight process »
 - Existe dans un process
 - Partage des ressources avec les autres Threads



JConsole

- Permet de visualiser les informations d'une JVM

Les variables 1 (type primitifs)

| | Type | Taille | De | à | Notation |
|-----------|--------|----------|------------------------|--------------------|------------|
| Entier | byte | 1 octet | -127 | 128 | |
| | short | 2 octets | -32768 | 32767 | |
| | int | 4 octets | -2×10^9 | 2×10^9 | |
| | long | 8 octets | -9×10^{18} | 9×10^{18} | 100...00L |
| Virgule | float | 4 octets | -1.4×10^{-45} | $3.4 * 10^{38}$ | 2.0f |
| | double | 8 octets | $4.9 * 10^{-324}$ | $1.7 * 10^{308}$ | 0.33...33d |
| Booléen | bool | 1 octet | false | true | |
| Caractère | char | 2 octets | | | 'a' |

Les variables 2 (type objet)

| | Type | Taille | De | à | Notation |
|-----------|----------------|----------|------------------------|--------------------|------------|
| Entier | Byte | 1 octet | -127 | 128 | |
| | Short | 2 octets | -32768 | 32767 | |
| | Integer | 4 octets | -2×10^9 | 2×10^9 | |
| | Long | 8 octets | -9×10^{18} | 9×10^{18} | 100...00L |
| Virgule | Float | 4 octets | -1.4×10^{-45} | $3.4 * 10^{38}$ | 2.0f |
| | Double | 8 octets | $4.9 * 10^{-324}$ | $1.7 * 10^{308}$ | 0.33...33d |
| Booléen | Boolean | 1 octet | false | true | |
| Caractère | Char | 2 octets | | | 'a' |
| Chaîne | String | variable | | | "Hello" |

Opérateurs 1

- Calcul arithmétique : +, -, *, /, %
- Assignment : =, +=, -=, *=, /=, %=
- Incrémentation : ++, -- en post ou préfixe
- Comparaison : ==, <, <=, >, >=, !=
- Logiques : &&, &&&, ||, |||, !

```
1 double i = 1.23;  
2 double j = 2.9999999;  
3 int k = (int)i;           //k vaut 1  
4 k = (int)j;               //k vaut 2
```

```
int i = 0;  
System.out.println(i++);  
System.out.println(i);  
System.out.println(++i);  
System.out.println(i);
```

```
0  
1  
2  
2
```

Opérateurs 2

- Bit : et &, ou inclusif |, ou exclusif ^
- Rotation : <<, >> (conservation du signe), >>>
- Test ternaire condition ? True : False
 - $a > b$? Instruction1 : Instruction2;

```
int i = 4;
System.out.println(Integer.toBinaryString(i));
i = i >> 1;
System.out.println(Integer.toBinaryString(i));
i = i << 1;
System.out.println(Integer.toBinaryString(i));
int j = -4;
System.out.println(Integer.toBinaryString(j));
j = j >> 1;
System.out.println(Integer.toBinaryString(j));
j = j << 1;
System.out.println(Integer.toBinaryString(j));
j = j >>> 1;
System.out.println(Integer.toBinaryString(j));
```

[illegible]

Convention de nommage

| Entité | Règle | Exemple |
|-----------------------|---|-------------------------------|
| Packages | Minuscules | com.company.project |
| Classes et Interfaces | CamelCase Une implémentation d'interface: nom + 'Impl' | MyClass() |
| Méthodes | Contient un verbe idéalement Première lettre minuscule, puis chaque lettre en début de mot avec une majuscule. | Public int getTotal() |
| Variables | Première lettre minuscule, puis chaque lettre en début de mot avec une majuscule. | String myVar; |
| Constantes | Majuscules | static final int VAL_MIN = 0; |

Condition

- if / else if / else

```
if (i < 0) {  
    System.out.println("Ce nombre est négatif !");  
} else if (i > 0) {  
    System.out.println("Ce nombre est positif !");  
} else {  
    System.out.println("Ce nombre est nul.");  
}
```

Switch

- switch, case, default

```
1 int note = 10; //On imagine que la note maximale est 20
2
3 switch (note)
4 {
5     case 0:
6         System.out.println("Ouch !");
7         break;
8     case 10:
9         System.out.println("Vous avez juste la moyenne.");
10        break;
11    case 20:
12        System.out.println("Parfait !");
13        break;
14    default:
15        System.out.println("Il faut davantage travailler.");
16 }
```


Boucles 1

- while, do / while

```
int b = 15;
while (b-- > 0) {
    System.out.println("Décollage dans " + b + " secondes");
}
```

- continue: passe à l'itération suivante
- break: sort de la boucle prématurément

Boucles 2

- for

```
1 for(int i = 1; i <= 10; i++)  
2 {  
3     System.out.println("Voici la ligne "+i);  
4 }
```

- for each

```
for (Suit suit : suits)  
    for (Rank rank : ranks)  
        sortedDeck.add(new Card(suit, rank));  
}
```

Installation du JDK

- Télécharger la dernière version stable du JDK sur le site d'oracle
- Pour Windows, ajouter le chemin du JDK à la variable d'environnement %PATH%

Hello World!

```
2 public class HelloWorld {  
3  
4     public static void main(String[] args) {  
5         System.out.println("HelloWorld");  
6     }  
7  
8 }  
9
```

- HelloWorld.java
- Le nom du fichier java doit être celui de la classe qu'il contient
- En commande :
 - javac HelloWorld.java
 - java HelloWorld
« HelloWorld »

TP #1

- Réaliser un mini jeu qui consiste à deviner un nombre aléatoire.
- Aide:
 - *Math.random()* génère un nombre aléatoire entre 0 et 1.
 - Lecture de l'entrée console:

```
Scanner sc = new Scanner(System.in);  
String line = sc.nextLine();
```

- Base:

```
import java.lang.Math;  
import java.util.Scanner;  
  
public class GuessIt {  
    public static void main(String args[]) {  
        // Code here  
    }  
}
```

TP #1 - Notation

- 1 – compilation & affichage d'une sortie
- 1 – génération du chiffre
- 1 – Condition
- 1 – Boucle
- 1 – méthode input