

Introduction à la programmation et à la modélisation objet en Java

Jean-François Viaud

January 6, 2020

Plan

- 1 Organisation générale
- 2 Les variables
- 3 Les structures de contrôle

The Java programming language is a general-purpose, concurrent, class-based, object-oriented language.

It is designed to be simple enough that many programmers can achieve fluency in the language.

The Java programming language is related to C and C++ but is organized rather differently, with a number of aspects of C and C++ omitted and a few ideas from other languages included.

It is intended to be a production language, not a research language, and so, as C. A. R. Hoare suggested in his classic paper on language design, the design has avoided including new and untested features.

Exemples

Ont été, au moins en partie, codés en Java :

- LibreOffice
- MineCraft
- Matlab

La programmation Android se fait en Java.

- Programme, Entrées, Sorties.
- Algorithme, Langages de programmation
- Paradigme : Impératif, Objet, Fonctionnel, etc.
- Compilation, Interprétation, Machine Virtuelle.

Exemple

Éléments de Java :

- `System.out.println("Youpi !");`
- `// import java.util.Scanner`
`Scanner s = new Scanner(System.in);`
- `int n = s.nextInt();`
- `import java.util.Scanner;`
`Scanner s = new Scanner(System.in);`
`int n = s.nextInt();`
- Java est sensible à la casse.

Code Source : First.java

Commandes : `javac, java`

Plan

- 1 Organisation générale
- 2 Les variables
- 3 Les structures de contrôle

The Java programming language is a statically typed language, which means that every variable and every expression has a type that is known at compile time.

The Java programming language is also a strongly typed language, because types limit the values that a variable can hold or that an expression can produce, limit the operations supported on those values, and determine the meaning of the operations.

Strong static typing helps detect errors at compile time.

Notions :

- Variable : Identifiant, Type, Valeur, Référence.
- Type primitifs : boolean, byte, short, int, long, char, float, double.
- Objets : String, Scanner, etc.
- Déclaration
- Opérations : affectation, égalité, division, concaténation, opérateurs logiques, etc. Les opérations dépendent du type.
- Conventions de nommage : contraintes syntaxiques, lisibilité, mémorisation.

Code Source : Resolution.java

Moodle CM-01 : Exercice

Exemples supplémentaires

- Logique : & ou &&, etc.
- Entier : / et %. Dépassement. Cast.
- Flottants : Arrondis. Conversion.
- Chaines : Concaténation.

- 1 Ecrivez un programme permettant la conversion euro - livre.
(Utilisation du mot clé `final`)
- 2 Ecrivez un programme convertissant une vitesse en m/s en une vitesse en km/h ; puis un programme convertissant une vitesse en km/h en une allure en min/km .

Plan

- 1 Organisation générale
- 2 Les variables
- 3 Les structures de contrôle

Branchement conditionnel

Exemple :

```
if (x > 0) {  
    x = x - 1;  
} else {  
    x = x + 1;  
}
```

- Noter les opérateurs logiques & ou &&.
- Attention à l'égalité des chaînes de caractère.

Code source : Condition.java

Exercice : Catégories d'IMC

Boucle itérative

Exemple :

```
int n = 11;
int fact = 1
for (int i = 1; i < n; i++) {
    fact = fact * i;
}
```

Répétition inconditionnelle.

Boucle conditionnelle

Exemple :

```
double x = 1000;
double y = 1
while (x > 0 & y < 5) {
    x -= y;
    y += 1 / x;
}
```

- `do{ ... } ...`
- `switch ... case ...`

Code source : Repetition.java, Visible.java

Exercice : Ecrivez un programme calculant une valeur approchée de $\ln 2$ à une précision saisie par l'utilisateur. Testez votre programme avec Moodle.

