

LẬP TRÌNH JAVA

Bài 2: Ngôn ngữ lập trình Java

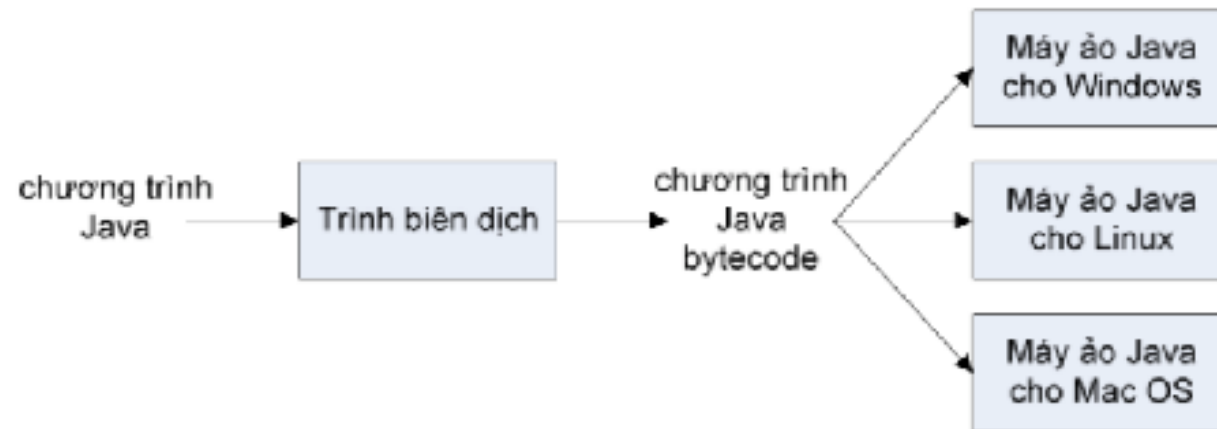
Nguồn gốc – Khái niệm

- ❑ Java được hãng Sun Microsystems thiết kế năm 1991.
- ❑ Java là ngôn ngữ lập trình hướng đối tượng
- ❑ “Write once, run anywhere” - Java có tính độc lập nền tảng, chạy trên máy ảo Java, có khả năng chạy trên nhiều hệ điều hành khác nhau mà không cần biên dịch lại.
- ❑ Java là ngôn ngữ lập trình phân tán.
- ❑ Java là một ngôn ngữ an toàn, phần mềm và mã độc không thể chèn thêm vào chương trình java sau khi đã biên dịch.

Máy ảo Java – Java Virtual Machine (JVM)

Các chương trình viết bằng ngôn ngữ Java được biên dịch thành mã máy (Java bytecode)

Để chạy được chương trình Java, cần một trình thông dịch dành cho Java bytecode, trình thông dịch này giả lập máy ảo Java theo kiểu Virtual PC, gọi là Java Virtual Machine.



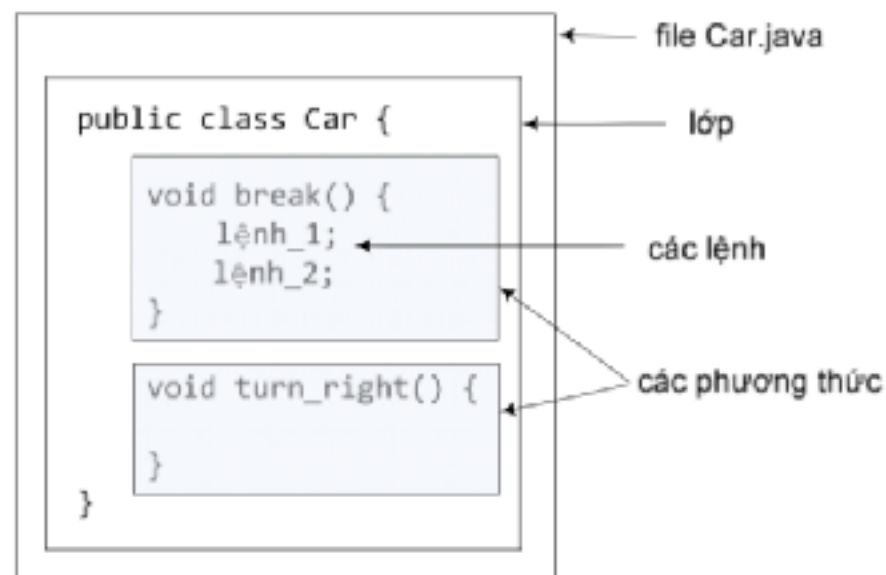
Cài đặt

<https://freetuts.net/huong-dan-cau-hinh-va-cai-dat-java-1208.html>

Cấu trúc mã nguồn Java

Tên file có phần mở rộng là .java, chứa một định nghĩa lớp (class), mỗi lớp là một phần của chương trình. Nhiều lớp hợp thành một chương trình Java.

Mỗi lớp chứa các thuộc tính (attribute) và phương thức (method) của đối tượng.



Chương trình Java đầu tiên

```
// A program to display "Hello World" message
public class HelloWorld {
    public static void main(String[] args) {
        System.out.println("Hello World");
    }
} // End of class HelloWorld
```

Các khái niệm cơ bản

- ❑ Biến
- ❑ Các phép toán cơ bản
- ❑ Cấu trúc điều khiển

Biến

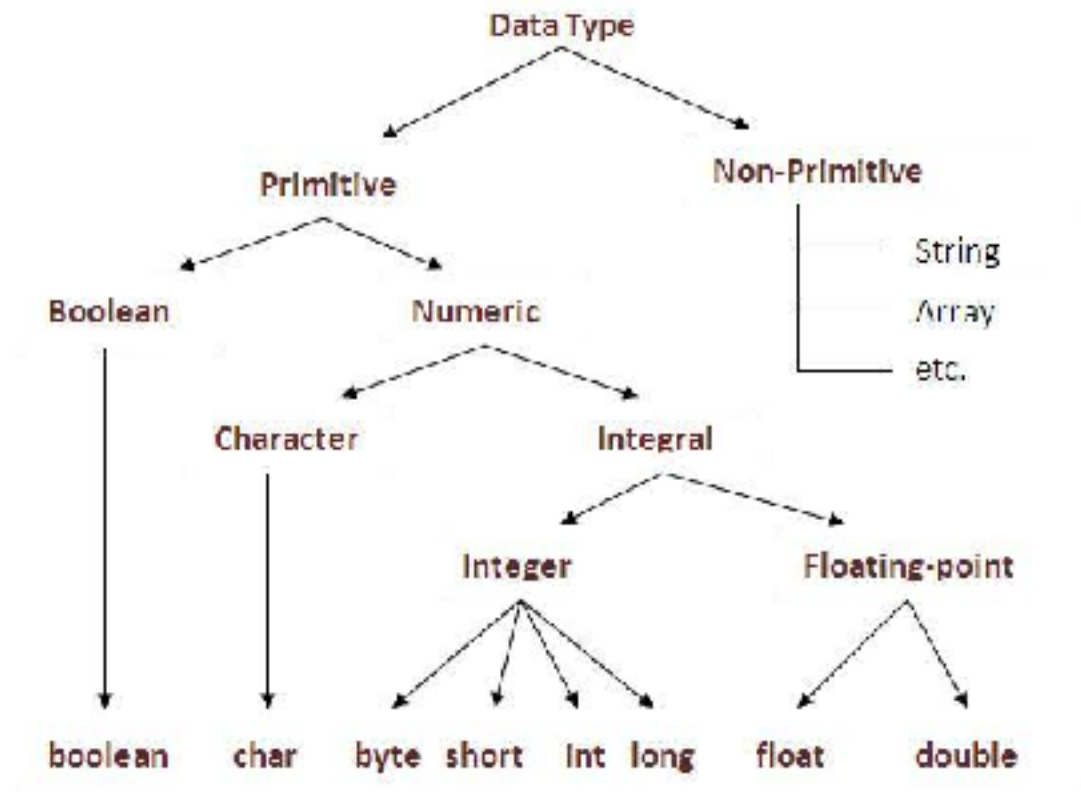
- ❑ Java yêu cầu khai báo biến trước khi sử dụng

Khai báo: `int x;`

Sử dụng: `x = 10;`

- ❑ Biến nằm trong hàm là biến địa phương. Phạm vi hiệu lực từ dấu `{` đến dấu `}`
- ❑ Biến nằm trong đối tượng là thuộc tính của đối tượng.
- ❑ Biến tĩnh (static) là biến nằm trong đối tượng, khởi tạo một lần duy nhất, và được chia sẻ giữa các đối tượng của lớp đó.

Kiểu dữ liệu



Kiểu dữ liệu

Số nguyên: int, long, short, byte, char (số nguyên không âm)

Ký tự: char

Kiểu số thực: float, double

Trạng thái đúng sai: boolean

Kiểu chuỗi: String

Kiểu mảng: int[], String[]

Khai báo biến

Kiểu dữ liệu nguyên thủy:

```
int a, b, c;
```

```
float a = 3.5, b, c = 0;
```

Kiểu mảng một chiều:

```
int[] a, b, c; hoặc int a[],b[],c[];
```

```
int[] a = new int[20];
```

Kiểu mảng hai chiều:

```
int[][] a; hoặc int [][]a; hoặc int a[][]; hoặc int[] a[];
```

Các phép toán cơ bản

- ❑ Phép gán: `biến = biểu thức;`
- ❑ Phép toán số học: `+, -, *, /, %, ++, --,`
- ❑ Phép gán kèm theo biểu thức: `+=, -=, *=, /=, %=, >>=, <<=, &=, ^=, |=`
- ❑ Phép toán so sánh: `>, <, >=, <=, ==, !=`
- ❑ Phép toán logic: `&&, ||, !`
- ❑ Độ ưu tiên các phép toán:
 - Toán tử đơn (`+, -, !, ++, --`) -> Toán tử đơn (`*, /, %`) -> Toán tử đôi (`&&, ||`) -> Phép toán so sánh
 - Có thể dùng cặp ngoặc `()` để xác định thứ tự ưu tiên trong biểu thức

Cấu trúc điều khiển

Cấu trúc rẽ nhánh if-else

```
if (score >= 8.5)
    grade = 'A';
else if (score >= 7.0)
    grade = 'B';
else if (score >= 5.5)
    grade = 'C';
else if (score >= 4.0)
    grade = 'D';
else
    grade = 'F';
```

Cấu trúc điều khiển

Cấu trúc rẽ nhánh switch-case

```
switch (biểu_thức) {  
    case hằng_1:  
        tập_lệnh_1; break;  
    case hằng_2:  
        tập_lệnh_2; break;  
    ...  
    default:  
        tập_lệnh_mặc_định;  
}
```

```
switch (grade) {  
    case 'A':  
    case 'B':  
    case 'C':  
        cout << "Grade is A, B or C."; break;  
    default:  
        cout << "Grade is not A, B or C.";  
}
```

Cấu trúc điều khiển

Cấu trúc lặp while

```
while (điều_kiện_lặp)  
    thân_vòng_lặp
```

```
count = 1;  
while (count <= number) {  
    System.out.print(count + ", ");  
    count++;  
}
```

Cấu trúc điều khiển

Cấu trúc lặp do-while

```
do  
    thân_vòng_lặp  
while (điều_kiện_lặp);
```

```
count = 1;  
do {  
    System.out.print(  
        count + ", ");  
    count++;  
} while (count <= number);
```


Cấu trúc điều khiển

So sánh cấu trúc while & do-while

```
count = 1;
while (count <= number) {
    System.out.print(
        count + ", ");
    count++;
}
```

```
count = 1;
do {
    System.out.print(
        count + ", ");
    count++;
} while (count <= number);
```

Cấu trúc điều khiển

Cấu trúc lặp for

*for (khởi_tạo; điều_kiện_lặp; cập_nhật)
thân_vòng_lặp*

```
for (int count = 0; count < subjects; count++) {  
    float mark;  
    mark = input.nextFloat();  
    sum += mark;  
}
```

Cấu trúc điều khiển

Lệnh **break** & **continue** trong cấu trúc lặp

- break: chấm dứt cấu trúc lặp
- continue: kết thúc lần lặp và bắt đầu lần lặp kế tiếp

```
for (count = 0; count < subjects; count++) {  
    float mark;  
    mark = input.nextFloat();  
    if (mark < 0) break;  
    sum += mark;  
}
```

```
while (count < subjects) {  
    System.out.print("#" + (count+1) + ": ");  
    float mark = input.nextFloat();  
    if (mark < 0) {  
        System.out.println(mark + " ignored");  
        continue;  
    }  
    sum += mark;  
    count++;  
}
```

Bài tập

Viết chương trình Java thực hiện các công việc sau đây:

1. Viết chương trình tìm ước số chung lớn nhất, bội số chung nhỏ nhất của hai số tự nhiên a và b.
2. Viết chương trình chuyển đổi một số tự nhiên ở hệ cơ số 10 thành số ở hệ cơ số b bất kì ($1 < b \leq 36$).
3. Hãy viết chương trình tính tổng các chữ số của một số nguyên bất kỳ. Ví dụ: Số 8545604 có tổng các chữ số là: $8+5+4+5+6+0+4= 32$.
4. Viết chương trình liệt kê tất cả các số nguyên tố nhỏ hơn n cho trước.
5. Dãy số Fibonacci được định nghĩa như sau: $F_0 = 1$, $F_1 = 1$; $F_n = F_{n-1} + F_{n-2}$ với $n \geq 2$. Hãy viết chương trình tìm số Fibonacci thứ n.

Bài tập

6. Viết chương trình nhập một số nguyên dương n và thực hiện các chức năng sau: a) Liệt kê các ước số của n . Có bao nhiêu ước số. b) Liệt kê các ước số là nguyên tố của n .
7. Viết chương trình nhập một số nguyên dương n và thực hiện các chức năng sau: a) Tính tổng các chữ số của n . b) Phân tích n thành các thừa số nguyên tố.
8. Nhập một chuỗi ký tự. Đếm số từ của chuỗi ký tự đó. Ví dụ "Truong hoc" có 2 từ.
9. Nhập một chuỗi ký tự. Đếm số lần xuất hiện của từng từ trong chuỗi đó. Ví dụ: "ngon ngu lap trinh java la ngon ngu huong doi tuong" => ngon: 2, ngu: 2, lap: 1, trinh: 1, java: 1, la: 1, huong: 1, doi: 1, tuong: 1.
10. Viết chương trình thực hiện nhập một chuỗi họ tên theo cấu trúc: họ...đệm...tên; chuyển chuỗi đó sang biểu diễn theo cấu trúc tên...họ...đệm.

Bài tập

11. Nhập một mảng số nguyên $a_0, a_1, a_2, \dots, a_{n-1}$. In ra màn hình tổng của tất cả các phần tử trong mảng đó.
12. Nhập một mảng số nguyên $a_0, a_1, a_2, \dots, a_{n-1}$. In ra màn hình các phần tử xuất hiện trong mảng đúng 2 lần.
13. Nhập một mảng số nguyên $a_0, a_1, a_2, \dots, a_{n-1}$. Hãy sắp xếp mảng theo thứ tự tăng dần, sau đó chèn phần tử x vào mà vẫn đảm bảo mảng là tăng dần.
14. Nhập vào 2 ma trận 3×3 , tính tổng, hiệu, tích của 2 ma trận đó.
15. Viết chương trình nhập vào ma trận A có n dòng, m cột, các phần tử là số nguyên lớn hơn 0 và nhỏ hơn 100. Thực hiện các chức năng sau:
 - a) Tìm và in ra phần tử lớn thứ nhất với chỉ số của nó (chỉ số đầu tiên tìm được).
 - b) Tìm và in ra các phần tử là số nguyên tố của ma trận.
 - c) Tìm cột trong ma trận có nhiều số nguyên tố nhất.