**Buổi 1 – System.out.print, Comment, Variables and Types**

# **System.out.print**

Hiển thị ra màn hình dòng chữ Hello, World! là một trong những chương trình cơ bản nhất khi bắt đầu làm quen với một ngôn ngữ lập trình. Trong Java để hiển thị ra màn hình một dòng chữ nào đó bạn có thể sử dụng câu lệnh System.out.println(). Ví dụ để hiển thị ra màn hình dòng chữ "Hello Java" bạn có thể làm như sau:

public class HelloWorld {

public static void main(String[]args) {

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

}

}

Kết quả khi chạy chương trình:

Hello Java

Nếu bạn là người mới bắt đầu thì trong đoạn code trên có khá nhiều câu lệnh mà bạn không biết như public class HelloWorld và public static void main(String[]args) nhưng bạn đừng quan tâm tới các câu lệnh này vội, hãy coi như những câu lệnh này là bắt buộc đối với 1 chương trình Java.

Tương tự như trên, bạn có thể viết chương trình hiển thị ra màn hình tên của mình. Ví dụ nếu bạn tên là Pham Tuan Tai thì bạn có thể làm như sau:

public class HelloWorld{

public static void main(String[]args){

System.out.println("Pham Tuan Tai");

}

}

## ***In nhiều dòng ra màn hình***

Bạn có thể sử dụng 2 câu lệnh System.out.println() để hiển thị ra thông tin trên 2 dòng giống như sau:

public class HelloWorld{

public static void main(String[]args){

System.out.println("Line1");

System.out.println("Line2");

}

}

Kết quả khi chạy chương trình:

Line1

Line2

Ngoài ra bạn cũng có thể sử dụng ký tự \n (in xuống dòng) để hiển thị thông tin trên nhiều dòng. Ví dụ:

public class HelloWorld{

public static void main(String[]args){

System.out.println("Line1\nLine2\nLine3");

}

}

Kết quả khi chạy chương trình:

Line1

Line2

Line3

## ***In kết quả phép tính***

Ngoài việc sử dụng câu lệnh System.out.println() để hiển thị ra màn hình thông tin dưới dạng các ký tự thì bạn cũng có thể sử dụng câu lệnh này để hiển thị ra màn hình giá trị của các phép tính như phép cộng, phép trừ, phép nhân và phép chia giống như chương trình sau:

public class HelloWorld{

public static void main(String[] args){

System.out.println(8 + 4);

System.out.println(8 - 4);

System.out.println(8 \* 4);

System.out.println(8 / 4);

}

}

Kết quả khi chạy chương trình:

12

4

32

2

Qua ví dụ trên bạn hoàn toàn có thể tạo ra được một chương trình tính toán đơn giản. Ví dụ bạn có thể viết chương trình tính tích của 17 và 18 giống như sau:

public class HelloWorld{

public static void main(String[] args){

System.out.println(17 \* 18);

}

}

Kết quả khi chạy chương trình:

306

Quay trở lại phần bài tập, bạn hãy viết chương trình hiển thị lên màn hình tổng của 313 và 122.

## ***In ra phép tính cùng kết quả***

Để hiển thị ra màn hình 125 + 206 = bạn có thể làm như sau:

public class HelloWorld {

public static void main(String[] args) {

System.out.println("125 + 206 = ");

}

}

Kết quả khi chạy chương trình:

125 + 206 =

Để hiển thị ra màn hình tổng của 125 và 206 bạn có thể làm như sau:

public class HelloWorld {

public static void main(String[] args) {

System.out.println(125 + 206);

}

}

Kết quả khi chạy chương trình:

331

Để làm được bài này bạn cần kết hợp được 2 câu lệnh trên giống như sau:

public class HelloWorld {

public static void main(String[] args) {

System.out.println("125 + 206 = " + (125 + 206));

}

}

Lưu ý: trong đoạn code trên nếu không có dấu đóng ngoặc ở biểu thức 125 + 206 giống như sau:

public class HelloWorld {

public static void main(String[] args) {

System.out.println("125 + 206 = " + 125 + 206);

}

}

Thì chương trình sẽ hiển thị ra:

125 + 206 = 125206

Do khi thực hiện các toán tử có cùng độ ưu tiên thì chương trình sẽ thực hiện từ trái qua phải nên biểu thức 125 + 206 sẽ không được thực hiện, thay vào đó là chương trình sẽ thực hiện cộng các xâu ký tự.

# **Comment**

Comment hay còn gọi là chú thích được dùng để giải thích các dòng code, giúp cho người đọc hay chính người viết code sau này hiểu được source code dễ dàng hơn. Các ký tự và đoạn code bên trong comment sẽ được chương trình bỏ qua khi biên dịch.

Trong Java có 2 loại comment là comment trên một dòng và comment trên nhiều dòng.

Comment trên một dòng sẽ được bắt đầu với //. Ví dụ đơn giản về comment một dòng:

public class HelloWorld{

public static void main(String[] args){

// Dùng phương thức System.out.println() để hiển thị ra Hello, World!

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

}

}

Kết quả khi chạy chương trình:

Hello World!

Bạn cũng có thể đặt comment ngay sau câu lệnh:

public class HelloWorld{

public static void main(String[] args){

System.out.println("Hello, World!"); // Dùng phương thức System.out.println() để hiển thị ra Hello, World!

}

}

Comment trên nhiều dòng trong Java sẽ bắt đầu bằng dấu /\* và kết thúc bằng dấu \*/. Ví dụ:

/\*

Chương trình Java hiển thị ra màn hình dòng chữ Hello, World!

sử dụng câu lệnh System.out.println()

\*/

public class HelloWorld{

public static void main(String[] args){

System.out.println("Hello, World!"); // Dùng phương thức System.out.println() để hiển thị ra Hello, World!

}

}

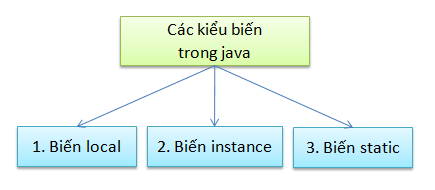
Kết quả khi chạy chương trình:

Hello World!

# **Variables (Biến)**

## ***Các kiểu biến***

Trong java, biến là tên của vùng nhớ. **Có 3 kiểu biến trong java**, bao gồm biến local (biến địa phương), biến instance (biến toàn cục) và biến static.



## **Khai báo biến trong java**

**Cú pháp khai báo biến:**

DataType varName [ = value] [, varName2] [ = value2]...;

Trong đó, DataType là kiểu dữ liệu của biến, varName là tên biến.

**Quy tắc đặt tên biến trong java:**

* Chỉ được bắt đầu bằng một ký tự(chữ), hoặc một dấu gạch dưới(\_), hoặc một ký tự dollar($)
* Tên biến không được chứa khoảng trắng
* Bắt đầu từ ký tự thứ hai, có thể dùng ký tự(chữ), dấu gạch dưới(\_), hoặc ký tự dollar($)
* Không được trùng với các từ khóa
* Có phân biệt chữ hoa và chữ thường

**Ví dụ về khai báo biến trong java:**

**public class Bien {**

**public static float PI = 3.14f;   // Đây là biến static**

**int n;                            // Đây là biến instance**

**public Bien () {**

**char c = 'c';                 // Đây là biến local**

**}**

**}**

## ***Biến kiểu nguyên***

Khái niệm biến trong lập trình cũng giống khái niệm biến trong toán học, biến được dùng để đại diện cho một giá trị nào đó.

Để khai báo biến kiểu số nguyên trong Java bạn sử dụng từ khóa int (int và viết tắt của integer) giống như sau:

int tên\_biến;

Ví dụ về chương trình tạo và sử dụng biến kiểu số nguyên:

public class Variable {

public static void main(String[] args) {

// Khai báo biến a kiểu số nguyên

int a;

// Gán giá trị cho a = 5

a = 5;

// Hiển thị ra màn hình giá trị của biến a

System.out.println("a = " + a);

}

}

Kết quả khi chạy chương trình:

a = 5

Bạn còn có thể vừa khai báo vừa gán giá trị cho biến trong 1 câu lệnh giống như chương trình sau:

public class Variable {

public static void main(String[] args) {

// Khai báo biến a kiểu số nguyên và gán giá trị cho a = 438

int a = 438;

// Khai báo biến b kiểu số nguyên và gán giá trị cho b = 238

int b = 238;

// Hiển thị ra màn hình hiệu của a và b

System.out.println("a - b = " + (a - b));

}

}

Kết quả khi chạy chương trình:

a - b = 200

Ngoài ra, bạn cũng có thể khai báo và gán giá trị cho nhiều biến trong 1 câu lệnh như giống như chương trình sau:

public class Variable {

public static void main(String[] args) {

int a = 438, b = 238;

System.out.println("a - b = " + (a - b));

}

}

Khi đặt tên cho biến bạn cần tuân theo 1 số nguyên tắc sau:

* Tên biến có thể có chữ cái, chữ số và dấu gạch dưới nhưng ký tự đầu tiên của tên biến bắt buộc phải là dấu gạch dưới hoặc chữ cái.
* Tên biến không được có khoảng trắng.
* Tên biến không được trùng với các từ khóa như int, float, double, class, ...

Một số tên biến hợp lệ:

int a;

int \_a;

int a10\_;

Một số tên biến không hợp lệ:

int 10a;

int a 10;

int int;

## ***Biến kiểu chuỗi ký tự***

Ở đoạn code này int là viết tắt của **integer** có nghĩa là biến a sẽ chỉ lưu được các giá trị là số nguyên. Vì vậy bạn không thể gán:

int a = "Codelearn";

Để tạo ra biến lưu được các ký tự như tên, địa chỉ, bạn cần khai báo biến với kiểu dữ liệu là String thay vì int giống như chương trình sau:

public class Variable {

public static void main(String[] args) {

// Khai báo biến name kiểu chuỗi các ký tự và gán giá trị cho name = "Codelearn"

String name = "Codelearn";

System.out.println(name);

}

}

Kết quả khi chạy chương trình:

Codelearn

## ***Biến kiểu số thực***

Để khai báo và gán giá trị cho biến kiểu số thực bạn dùng từ khóa double:

// Khai báo biến a kiểu số thực và gán giá trị cho a = 10.5

double a = 10.5;

Ví dụ chương trình tạo và hiển thị biến kiểu số thực ra màn hình:

public class Variable {

public static void main(String[] args) {

// Khai báo biến a kiểu số thực và gán giá trị cho a = 1.5

double a = 1.5;

System.out.println("a = " + a);

}

}

Kết quả khi chạy chương trình:

a = 1.5

Ngoài double ra thì trong Java còn một kiểu dữ liệu nữa cũng được dùng để lưu trữ số thực nữa là float nhưng trong hầu hết mọi trường hợp bạn nên sử dụng kiểu double.

**Lưu ý:** Bạn không thể dùng biến kiểu int để lưu trữ số thực và kết quả của phép toán giữa 2 số nguyên là một số nguyên. Ví dụ:

public class Variable {

public static void main(String[] args) {

int a = 4;

int b = 3;

System.out.println(a / b);

System.out.println(4 / 3);

}

}

Kết quả khi chạy chương trình:

1

1

Kết quả sẽ là 1 mà không phải 1.(3) do cả a và b đều là biến kiểu số nguyên nên kết quả sẽ là một số nguyên. Do đó khi thực hiện các phép toán trong Java bạn cần lưu ý tới kiêu dữ liệu. Để chương trình trên hiển thị đúng kết quả bạn có thể làm như sau:

public class Variable {

public static void main(String[] args) {

double a = 4;

double b = 3;

System.out.println(a / b);

System.out.println(4.0 / 3);

}

}

Kết quả khi chạy chương trình:

1.3333333333333333

1.3333333333333333

## ***Biến kiểu ký tự***

Kiểu ký tự trong Java là kiểu dữ liệu chỉ lưu trữ được 1 ký tự trong bảng mã UTF-16, ký tự này có thể là một chữ cái (a, b, c, ..., x, y, z), một chữ số  (0, 1, 2, ..., 9), một phép toán (+, -, \*, /) hay một ký tự bất kỳ khác (!, &, ...).

Biến kiểu ký tự được khai báo bằng từ khóa char (char là viết tắt của **character**)**.**Ví dụ chương trình tạo và hiển thị biến kiểu ký tự:

public class Variable {

public static void main(String[] args) {

// Khai báo biến ký tự c và gán cho c ký tự x

char c = 'x';

System.out.println(c);

}

}

Kết quả khi chạy chương trình:

x

Kiểu ký tự bản chất là một kiểu số nguyên, bạn có thể hiện thị giá trị nguyên của một ký tự bằng câu lệnh (int)ký\_tự; giống như chương trình sau:

public class Variable {

public static void main(String[] args) {

System.out.println("a = " + (int) 'a');

System.out.println("b = " + (int) 'b');

System.out.println("c = " + (int) 'c');

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

System.out.println("x = " + (int) 'x');

System.out.println("y = " + (int) 'y');

System.out.println("z = " + (int) 'z');

}

}

Kết quả khi chạy chương trình:

a = 97

b = 98

c = 99

...

x = 120

y = 121

z = 122

Từ kết quả trên có thể thấy, giá trị số của các chữ cái từ 'a' đến 'z' là tăng dần, mỗi giá trị liên tiếp cách nhau 1 đơn vị.

Do kiểu ký tự là một kiểu số nguyên nên bạn có thể áp dụng các phép toán +, -, \*, /. Từ đoạn code bên trên có thể thấy để hiển thị ký tự 'd' bạn có thể lấy ký tự 'a' + 3, để hiển thị ký tự 'e' bạn có thể lấy ký tự 'd' + 1 như chương trình sau:

public class Variable {

public static void main(String[] args) {

char d = 'a' + 3;

char e = 'd' + 1;

System.out.println(d);

System.out.println(e);

}

}

Màn hình sẽ hiển thị ra:

d

e

# **DataTypes (Các kiểu dữ liệu)**

Trong ngôn ngữ lập trình Java có các kiểu dữ liệu lưu trữ số nguyên như: byte, short, int, long, ...

Sự khác biệt giữa các kiểu dữ liệu này là miền giá trị:

Miền giá trị của kiểu byte là từ -128 tới  127.

Miền giá trị của kiểu short là từ -32768 tới 32767.

Miền giá trị của kiểu int là từ -2147483648 tới 2147483647.

Miền giá trị của kiểu long là từ -9223372036854775808 tới 9223372036854775807.

Có thể thấy miền giá trị của kiểu long là lớn nhất, bạn có thể dùng kiểu dữ liệu này để thay cho các kiểu dữ liệu khác nhưng bù lại dùng kiểu dữ liệu này sẽ tốn bộ nhớ hơn (kiểu dữ liệu có miền giá trị càng lớn sẽ càng tốn bộ nhớ). Do đó bạn cần sử dụng các kiểu dữ liệu một cách hợp lý, ví dụ biến để lưu trữ số học sinh trong một lớp học nên là kiểu short vì thường một lớp chỉ có vài chục học sinh.

Quay lại bài này, bạn không thể dùng kiểu int để lưu giá trị cho biến a và b do kết quả của phép nhân 2 số kiểu int sẽ là 1 số kiểu int mà miền giá trị của kiểu int là không đủ để lưu trữ kết quả của phép nhân 384847522 \* 988347273. Ví dụ:

public class Variable {

public static void main(String[] args) {

int a = 384847522;

int b = 988347273;

System.out.println(a \* b);

}

}

Kết quả khi chạy chương trình:

-628954446

Có thể thấy kết quả này là sai hoàn toàn, bạn cần dùng kiểu dữ liệu có miền giá trị lớn hơn để lưu trữ biến a và biến b, đó là kiểu long:

public class Variable {

public static void main(String[] args) {

long a = 384847522;

long b = 988347273;

System.out.println(a \* b);

}

}

Kết quả khi chạy chương trình:

380362998889507506

Ngoài các kiểu dữ liệu bạn đã được học như char, String, short, int, long, ... thì trong Java còn một kiểu dữ liệu căn bản nữa là kiểu boolean (kiểu luận lý). Kiểu dữ liệu này chỉ nhận 2 giá trị là true và false (tương ứng với đúng và sai). Ví dụ về chương trình sử dụng kiểu boolean:

public class Variable {

public static void main(String[] args) {

// Khai báo biến a kiểu boolean và gán giá trị cho a = true

boolean a = true;

// Khai báo biến b kiểu boolean và gán giá trị cho a = true

boolean b = false;

System.out.println(a);

System.out.println(b);

}

}

Kết quả khi chạy chương trình:

true

false

Kiểu boolean thường được sử dụng để lưu trữ kết quả của các biểu thức điều kiện như a > b, a < b, a == b giống như sau:

public class Variable {

public static void main(String[] args) {

int a = 5;

int b = 4;

boolean c = a > b;

System.out.println(c);

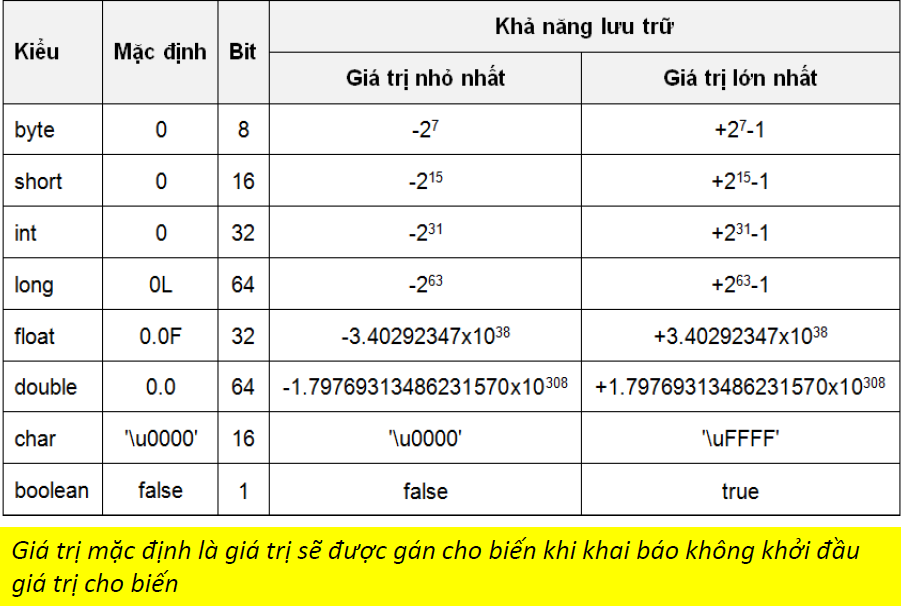
}

}

Kết quả khi chạy chương trình:

true

Bảng mô tả miền giá trị và kích thước của các kiểu dữ liệu trên:



Ở trong bảng trên có một cột là cột kích thước, cột này thể hiện số byte mà chương trình cần để lưu trữ biến của một kiểu dữ liệu. Ví dụ nếu bạn khai báo biến a kiểu số nguyên thì chương trình sẽ cần sử dụng 4 bytes trên ram để lưu trữ biến a.

Bảng trên không có kiểu String do kích thước của biến kiểu String phụ thuộc vào số ký tự của nó.

Có thể thấy kiểu dữ liệu có miền giá trị càng lớn thì càng tốn bộ nhớ nên bạn cần chọn kiểu dữ liệu phù hợp cho các biến. Ví dụ bạn không nên dùng biến kiểu long chỉ để lưu trữ số học sinh của một lớp học hay số nhân viên của một công ty, thay vào đó bạn nên dùng kiểu dữ liệu short hoặc int