

Học lập trình Java cơ bản



Bài 1: Biến, kiểu dữ liệu, toán tử trong Java

NỘI DUNG CHÍNH (7 nội dung)



**Các kiểu dữ liệu
cơ bản**



Khai báo biến



Phép gán dữ liệu



**Các phép toán cơ
bản**



**Hướng dẫn
Debug**



**Một ví dụ phép
gán cơ bản hay**



**Một số hàm
toán học
thường dùng**

Phần 1

Các kiểu dữ liệu cơ bản



Các kiểu dữ liệu cơ bản

- Trong khoa học máy tính và lập trình máy tính, **kiểu dữ liệu (data type)** là một cách phân loại dữ liệu cho trình biên dịch hoặc thông dịch hiểu các lập trình viên muốn sử dụng dữ liệu.
- Hầu hết các ngôn ngữ hỗ trợ nhiều kiểu dữ liệu khác nhau, như số nguyên, số thực, ký tự, ...

Kiểu số nguyên

→ *short*, *int*, *uint*, *long*, *ulong*...

Kiểu số thực

→ *float*, *double*, *decimal*

Kiểu ký tự

→ *char*

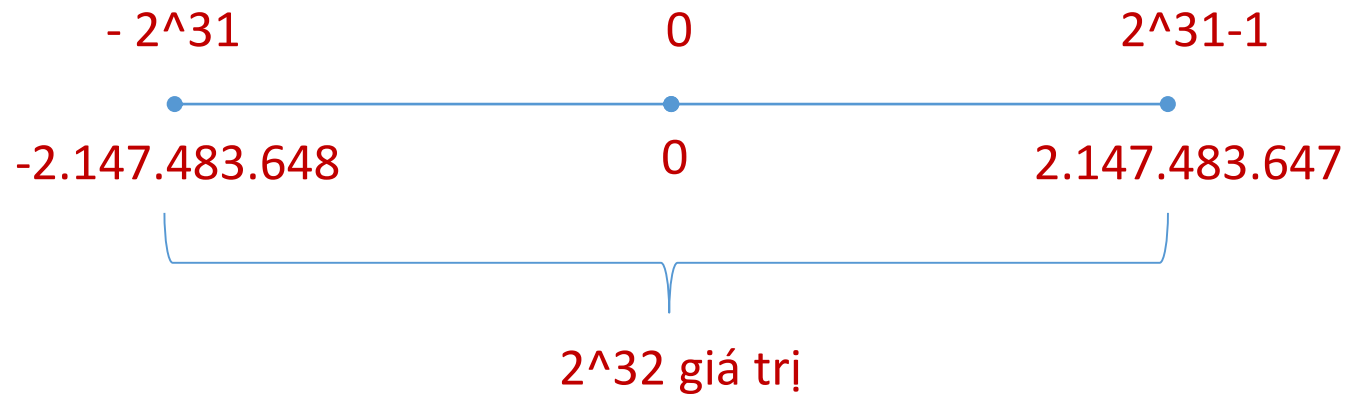
Kiểu chuỗi

→ *String*

Các kiểu dữ liệu cơ bản

Kiểu int: là kiểu số nguyên, có giá trị 4byte

4byte thì sao? Nghĩa là 32 bit, mà mỗi 1 bit thì biểu diễn được 2 số phân biệt 0,1. N bit thì biểu diễn được 2^N số phân biệt, cụ thể ở đây, sẽ có 2^{32} số phân biệt thể hiện bởi kiểu int



Nếu một số không nằm trong đoạn giá trị và cho vào tập này thì sao? Ngôn ngữ Java sẽ quản lý theo kiểu quay vòng giá trị.

Ví dụ giá trị **2.147.483.649** vượt ra ngoài 2 giá trị, vậy nếu cho vào tập int, sẽ quay vòng về đầu 2 giá trị, trở thành số **-2.147.483.647**

Các kiểu dữ liệu cơ bản

Kiểu dữ liệu nguyên thủy

Nhóm	Kiểu dữ liệu	Kích thước (bytes)	Ý nghĩa
Kiểu số nguyên	byte	1	Số nguyên dương không dấu có giá trị từ 0 đến 255
	short	2	Số nguyên có dấu có giá trị từ -32,768 đến 32,767
	int	4	Số nguyên có dấu có giá trị từ -2,147,483,647 đến 2,147,483,647
	long	8	Số nguyên có dấu có giá trị từ - 9,223,370,036,854,775,808 đến 9,223,370,036,854,775,807
Kiểu ký tự	char	2	Chứa một ký tự Unicode
Kiểu logic	boolean	1	Chứa 1 trong 2 giá trị logic là true hoặc false
Kiểu số thực	float	4	Kiểu số thực dấu chấm động có giá trị dao động từ 3.4E – 38 đến 3.4E + 38 , với 7 chữ số có nghĩa
	double	8	Kiểu số thực dấu chấm động có giá trị dao động từ 1.7E – 308 đến 1.7E + 308 , với 15, 16 chữ số có nghĩa



Các kiểu dữ liệu cơ bản

Lớp Wrapper

Lớp Wrapper trong java cung cấp cơ chế để chuyển đổi kiểu dữ liệu nguyên thủy thành kiểu đối tượng và ngược lại từ đối tượng thành kiểu dữ liệu nguyên thủy.

```
public class Main {
    public static void main(String[] args) {
        int num1 = 1;
        Integer num2 = Integer.valueOf(num1);    //đổi int thành Integer
        //auto boxing
        Integer num3 = num1;                    //tự động chuyển đổi int thành Integer trong nội bộ trình biên dịch

        //unboxing
        int num4 = num2.intValue();              //đổi Integer thành int
        int num5 = num3;                        //tự động chuyển đổi Integer thành int trong nội bộ trình biên dịch
    }
}
```

Kiểu dữ liệu nguyên thủy	Kiểu Wrapper
byte	Byte
short	Short
int	Integer
long	Long
char	Char
boolean	Boolean
float	Float
double	Double

Việc chuyển đổi tự động các kiểu dữ liệu nguyên thủy thành kiểu **Wrapper** tương đương của nó được gọi là hoạt động **autoboxing (hay boxing)** và ngược lại được gọi là **unboxing**.

Phần 2

Khai báo biến



Khai báo biến

Biến là gì?

- Biến là vùng nhớ dùng để lưu trữ các giá trị của chương trình.
- Mỗi biến gắn liền với một kiểu dữ liệu và một định danh duy nhất gọi là tên biến.

Một số quy tắc đặt tên biến

- Tên phân biệt chữ HOA và chữ thường.
- Tên biến chấp nhận các ký tự **chữ cái**, ký **số**, dấu **_** và dấu **\$**.
- Tên biến phải bắt đầu bằng một **chữ cái**, một dấu **_** hay dấu **\$**. Không bắt đầu bằng ký tự số.
- Tên biến không được trùng với các từ khóa.
- Tên biến không có khoảng trắng trong tên.
- Biến có thể được khai báo ở bất kỳ đâu trong chương trình.

Khai báo biến

Khai báo một biến

Tên Kiểu Dữ Liệu **Tên Biến ;**

Khai báo nhiều biến

Tên Kiểu Dữ Liệu **Tên Biến1, Tên Biến2... ;**

Ví dụ:

khai báo biến a là số nguyên:

Khai báo biến b,c là kiểu thực:

Khai báo biến d là kiểu ký tự:

```
int      a;  
double   b,c;  
char     c;
```

```
1  
2   int soNguyen1, soNguyen2, soNguyen3;  
3  
4   double soThuc1, soThuc2;  
5  
6   char kyTu1, kyTu2;
```

Khai báo biến

```
//Tên biến được viết thường và cần sát nghĩa với hoàn cảnh dùng
int count, temp;
double price, salary;
char key;
//Nếu tên có nhiều hơn 1 từ, sử dụng dấu "_" để phân tách
int time_count;
double monthly_salary;
char key_start;

//Phong cách bố trí mã nguồn theo kiểu Nhật
int          count, temp;          /* Lý giải về sử dụng count, temp */
double       price, salary;        /* Lý giải về sử dụng price, salary */
char         key;                  /* Lý giải về sử dụng key */
```

Phần 3

Phép gán dữ liệu



Phép gán dữ liệu

Gán giá trị cho một biến

Tên Biến = Giá Trị;

Gán biểu thức cho một biến

Tên Biến = Biểu thức;

Gán ngay khi khai báo biến

Kiểu dữ liệu Tên Biến = Giá trị hoặc Biểu thức;

Khi một biến được gán bởi một giá trị: bất kể biến đó có giá trị là gì trước đó, sau khi được gán, biến sẽ nhận lấy giá trị này

Khi một biến được gán bởi một biểu thức: chương trình thực hiện tính giá trị biểu thức trước, sau đó lấy kết quả tính được đem gán cho biến.

Ví dụ

```
1
2 //Hãy lý giải các giá trị nhận được bởi biến value
3 int value;
4 value = 10;
5 value = 10 + value;
6 value = 0;
7
8 //Hãy lý giải các giá trị của các biến number1, number2, number3
9 int number1 = 10, number2 = 20;
10 int number3 = number1 + number2;
11
```

Phần 4

Các phép toán cơ bản



Các phép toán cơ bản

Loại toán tử	Các phép	Mô tả	Ví dụ
Số học	<div>+ , - , * / % ++ , --</div>	<div>Phép cộng, trừ, nhân thông thường</div> <div>Chia lấy nguyên nếu cả 2 số chia cùng nguyên</div> <div>Chia lấy dư trong số nguyên</div> <div>Tăng giá trị biến hiện tại lên hoặc giảm xuống 1</div>	<div>int a = 11;</div> <div>a++;</div> <div>int b = a/5;</div> <div>int c =a%5</div>
Gán	<div>=</div> <div>+= , -=, *=</div> <div>/= , %/=</div> <div>&= , =, ^=</div> <div>>>= , <<=</div>	<div>Gán giá trị biến với một giá trị cùng kiểu</div> <div>Tăng, giảm giá trị biến với một giá trị cụ thể</div> <div>Chia nguyên, dư hoặc thường rồi gán lại biến</div> <div>Thực hiện phép và, hoặc, phủ định biến rồi gán lại</div> <div>Dịch bit của biến với 1 giá trị cụ thể rồi gán lại</div>	<div>int x = 10;</div> <div>x +=5;</div> <div>x *=x;</div> <div>x %=3;</div>
So sánh	<div>== , !=</div> <div>> , < , >=, <=</div>	<div>So sánh bằng hoặc khác</div> <div>So sánh thông thường</div>	<div>int a = 10,b=20;</div> <div>bool c= a!=b;</div>
Logic	<div>&&</div> <div> </div> <div>!</div>	<div>Phép và: chỉ đúng khi các toán hạng logic cùng đúng</div> <div>Phép hoặc: chỉ sai khi các toán hạng cùng sai</div> <div>Phép phủ định: đảo ngược giá trị logic</div>	<div>int a = 10,b=20;</div> <div>bool c = !(a==b) (a>=(b-a));</div>

Các phép toán cơ bản

```
1
2 //Kết thúc dòng số 5, value1, value2 có giá trị là bao nhiêu?
3 int value1 = 10, value2 = 20;
4 value1 = ++value1 + value2 * (value2 / value1);
5 value2 += 10;
6
7 //Vì sao phải nhân 1.0 vào value2
8 double division_param = value1 / (value2 * 1.0);
```

Phần 5

Hướng dẫn debug



Hướng dẫn debug trên Visual Studio

(Xem Hướng Dẫn Trên Video)

Phần 6

Một số ví dụ phép gán cơ bản



Ví dụ đảo giá trị và ý nghĩa

Yêu cầu: Cho biến a có giá trị là 10, b có giá trị là 20. Hãy đổi giá trị 2 biến này với nhau.

Cách 1: dùng thêm biến phụ c

```
1 |  
2 |  
3 | int value1 = 10, value2 = 20, temp;  
4 | temp = value1;    //value1 nhờ temp giữ hộ giá trị  
5 | value1 = value2;  //sau đó value1 nhận giá trị của value2  
6 | value2 = temp;    //value2 nhận giá trị cũ của value1 đang lưu ở temp  
_ |
```

*Ưu nhược
điểm từng
cách là gì?*

Cách 2: không sử dụng biến phụ c

```
10 | int a = 10, b = 20;  
11 | a = a + b;  //a = 30  
12 | b = a - b;  //b = 10  
13 | a = a - b;  //a = 20
```

*Ta rút ra
được ý
nghĩa gì?*

Phần 7

Một số hàm toán học
thường dùng



Một số hàm toán học thường dùng

Hàm	Ví dụ	Kết quả
sin,cos,tan	Math.sin(x);	Tính sin, cos, tang cho biến x
log,log10	Math.log(x);	Tính logarit và logarit cơ số 10 của x
abs	Math.abs(x);	Tính trị tuyệt đối của x
sqrt	Math.sqrt(x);	Tính căn bậc 2 của x
pow	Math.pow(x,y);	Tính x mũ y
ceiling	Math.ceil(x);	Tìm số nguyên nhỏ nhất mà vẫn lớn hơn hoặc bằng x
floor	Math.floor(x);	Tìm số nguyên lớn nhất mà vẫn nhỏ hơn hoặc bằng x
exp	Math.exp(x);	Tính e mũ x

Ví dụ

$$5y^2 + \sqrt[3]{3y}$$

1
2
3
4

```
double value, y = 2;  
value = 5 * Math.Pow(y, 2) + Math.Pow(3 * y, (1.0 / 3));  
....
```

Chuyển đổi kiểu dữ liệu

Chuyển chuỗi thành kiểu cụ thể:

```
KiểuDữLiệu.Parse<Kiểu dữ liệu>("chuỗi")
```

Chuyển đổi kiểu bất kỳ sang chuỗi:

```
String.valueOf (Biến)  
hoặc  
<tên kiểu>.toString (Biến)
```

Hiển thị thông tin lên màn hình console

(Xem Hướng Dẫn Trên Video)



Xin trân trọng cảm ơn



Lotus Academy



lotusacademy.edu.vn