

Lập trình Java

- Lý thuyết: 2 chỉ
- Thực hành: 1 chỉ
- Giữa kỳ: 30% (Thi trên máy)
- Cuối kỳ: 50% (Thi trên giấy)
- Bài tập: 20% (Trên lớp+ về nhà)

Quá trình học

Điểm danh đầu giờ.

Bài tập nộp vào email:

[Google classroom](#)

Cú pháp BTJV_tuan?_ten

Lập trình Java

Tuần 1: Những khái niệm cơ bản

Nội dung

- Ngôn ngữ lập trình Java
- Thiết lập môi trường cho ứng dụng java
- Cấu trúc chương trình
- Khái niệm về biến và kiểu dữ liệu

Ngôn ngữ lập trình Java

Ngôn ngữ phổ biến, phát triển năm 1995 bởi Oracle.



- Hướng đối tượng
- Chạy trên mọi nền tảng
- Bảo mật cao
- Mạnh mẽ
- Phân tán
- Đa luồng

Vai trò của lập trình Java

- Đây là ngôn ngữ lập trình nền tảng cho những môn học về lập trình sau này:
- Ví dụ: Lập trình android, lập trình game, lập trình web.

Thiết lập môi trường cho ứng dụng java

- **JDK -Java Development Kit**
- **Java IDE: NetBeans, Eclipse**

Cấu trúc chương trình

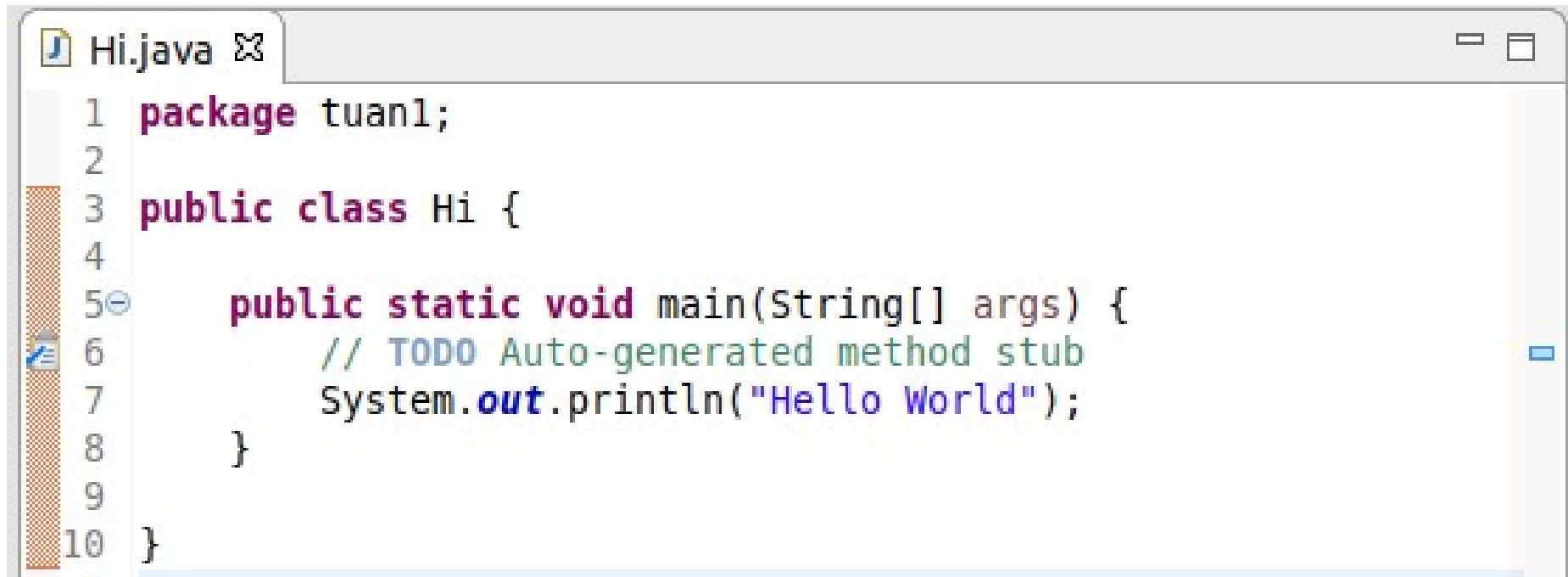
```
package com.poly;  
public class Program{  
    public static void main(String[] args){  
        // mã thực thi  
    }  
}
```

Save as

Program.java

- **com.ly**: tên gói chứa lớp (package) (sử dụng ký tự thường và dấu chấm ngăn cách)
- **Program**: tên lớp (class) (phải giống tên file.java, viết hoa ký tự đầu)
- **Main()**: tên phương thức (method) (lớp có thể có nhiều phương thức nhưng **main()** được gọi tự động khi chạy ứng dụng)

Cấu trúc chương trình

A screenshot of a Java IDE window titled 'Hi.java'. The code is as follows:

```
1 package tuan1;
2
3 public class Hi {
4
5     public static void main(String[] args) {
6         // TODO Auto-generated method stub
7         System.out.println("Hello World");
8     }
9
10 }
```

The code is color-coded: keywords like 'package', 'public', 'class', 'static', 'void', and 'main' are in purple; the package name 'tuan1' is in blue; the class name 'Hi' is in purple; the method name 'main' is in purple; the parameter 'String[] args' is in black; the comment '// TODO Auto-generated method stub' is in green; the string 'Hello World' is in blue; and the 'out' property of 'System' is in blue. The IDE has a light gray background with a vertical scrollbar on the right and a horizontal scrollbar at the bottom.

Java Comments

- Sử dụng kí hiệu “//” khi muốn chú thích về dòng lệnh hoặc đóng dòng lệnh cho không thực thi:

// This is a comment

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

- Sử dụng cặp kí hiệu “/*” và “*/” khi muốn chú thích một đoạn hoặc đóng nhiều dòng lệnh.

/ The code below will print the words Hello World
to the screen, and it is amazing */*

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

Một số khái niệm cơ bản

Khái niệm biến:

- Là thành phần nắm giữ dữ liệu được chương trình sử dụng trong các biểu thức tính toán
- Mỗi biến có kiểu dữ liệu riêng và phải được khai báo trước.

Cú pháp

<kiểu dữ liệu><tên biến> (= giá trị)

Khai báo biến

Ví dụ:

```
public class MyClass{  
    public static void main(String[] args){  
        int a = 5;  
        int b = 7;  
        int c = a + b;  
        System.out.println("Tổng: " + c);  
    }  
}
```

Lưu ý: tên biến sử dụng các ký tự alphabet, số, \$ hoặc dấu gạch dưới (_), tên có phân biệt hoa thường, không bắt đầu bằng số, không đặt tên giống từ khóa

Khai báo biến

Ví dụ:

```
int myNumber = 10;
```

```
System.out.println(myNumber);
```

Output: 10

```
int myNumber;
```

```
myNumber = 10;
```

```
System.out.println(myNumber);
```

Khai báo biến

Ví dụ:

```
int myNumber = 10;
```

```
myNumber = 20;
```

```
System.out.println(myNumber);
```

Output: ??

```
final int myNumber = 10;
```

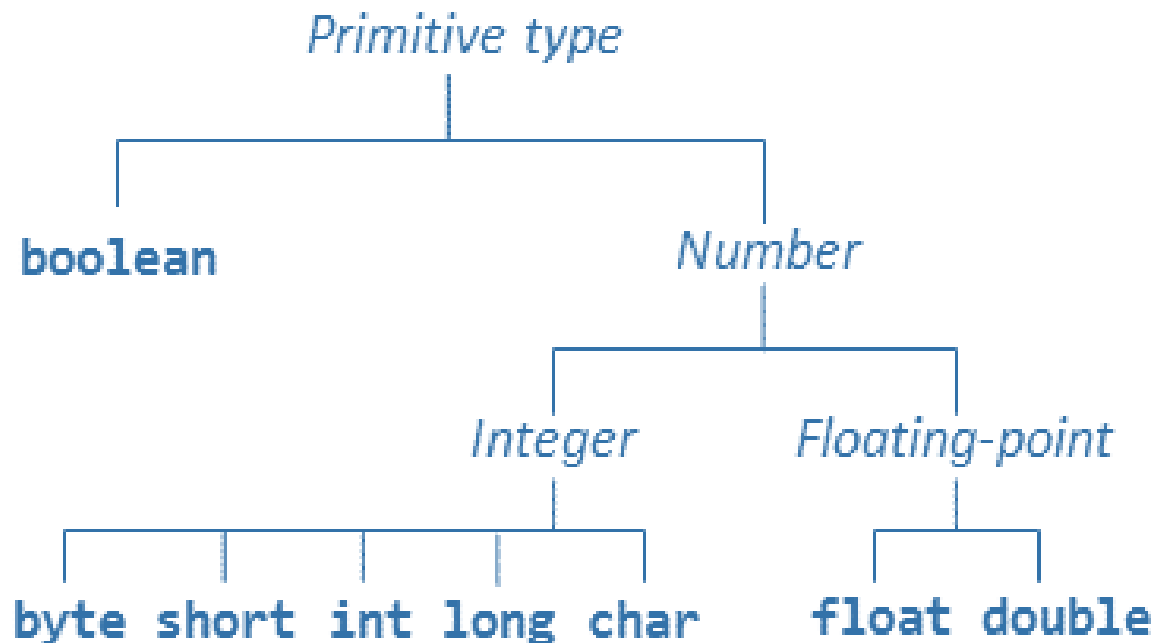
```
myNumber = 20;
```

```
System.out.println(myNumber);
```

Kiểu dữ liệu

Java có 2 loại kiểu dữ liệu:

- Các kiểu dữ liệu nguyên thủy (Primitive Data Types)
- Các kiểu dữ liệu tham chiếu (Reference Types)



Kiểu dữ liệu

Kiểu dữ liệu	Giá trị mặc định	Kích thước
boolean	false	1 bit
char	'\u0000'	2 byte
byte	0	1 byte
short	0	2 byte
int	0	4 byte
long	0L	8 byte
float	0.0f	4 byte
double	0.0d	8 byte

1 byte = 8 bit

1 bit đầu thể hiện dấu.

7 bit sau có thể biểu diễn số từ 0 -> 127.

⇒ **kiểu byte có phạm vi từ [-128,127]**

⇒ **Kiểu char [0,65535]; ký tự nhỏ nhất là '\u0000', ký tự lớn nhất là '\u65535'.**

Kiểu dữ liệu

Ví dụ:

```
int myNumber = 5;
```

```
float myFloatNumber =  
5.99f;
```

```
char myLetter = 'D';
```

```
boolean myBool = true;
```

Ví dụ:

```
float myFloatNumber = 5.99f;
```

```
double my double Number = 12E4d;
```

Tương ứng với $12 \cdot 10^4$

Kiểu dữ liệu

- Kiểu tham chiếu (Reference type) được tạo ra bởi sự kết hợp các kiểu nguyên thủy với nhau. Bao gồm: String, Arrays, Classes, Interface, v.v.
- Kiểu tham chiếu thường được sử dụng nhất đó là String, nó là sự kết hợp của các ký tự.

String a = "Hello World";

Kiểu dữ liệu

Primitive Data Types	Reference type
Không thể gọi phương thức	Có thể gọi phương thức
Luôn có một giá trị	có thể là null
Bắt đầu bằng một chữ cái viết thường	Bắt đầu bằng một chữ cái viết hoa

```
class Student {  
  
    string fullName;  
  
    int age;  
}
```

Ép kiểu

Widening Casting (tự động): chuyển từ kiểu dữ liệu nhỏ hơn sang kiểu dữ liệu lớn hơn.

Narrowing Casting (thủ công): chuyển từ kiểu dữ liệu lớn hơn sang kiểu dữ liệu nhỏ hơn.

Ví dụ

```
int a = 5;
```

```
double b = 9.4;
```

```
b = a;      //ép kiểu tự động
```

```
a = (int)b; //ép kiểu tường minh phần thập phân sẽ bị bỏ
```

Chuyển chuỗi sang kiểu nguyên thủy

Chuỗi => Nguyên thủy
<code>byte Byte.parseByte(String)</code>
<code>short Short.parseShort(String)</code>
<code>int Integer.parseInt(String)</code>
<code>long Long.parseLong(String)</code>
<code>float Float.parseFloat(String)</code>
<code>double Double.parseDouble(String)</code>
<code>boolean Boolean.parseBoolean(String)</code>

Lớp bao (Wrapper) kiểu nguyên thủy

Wrapper giúp xây dựng ra những phương thức khác nhằm bổ sung thêm cho tính năng đơn giản ban đầu của kiểu nguyên thủy.

Chú ý: các lớp Wrapper là các lớp final => không có lớp con.

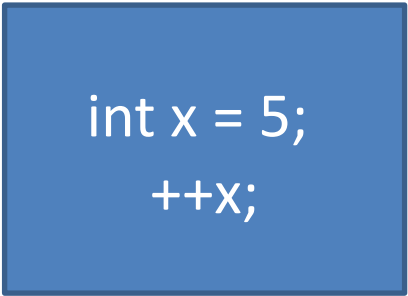
- Rất nhiều hàm trong Java chỉ làm việc với kiểu đối tượng mà không làm việc với kiểu nguyên thủy

Nguyên Thủy	Lớp bao
byte	⇔ Byte
short	⇔ Short
int	⇔ Integer
long	⇔ Long
float	⇔ Float
double	⇔ Double
char	⇔ Character
boolean	⇔ Boolean

Toán tử số học

- Toán tử số học là các phép toán thao tác trên các số nguyên và số thực

+	Tính tổng của 2 số
-	Tính hiệu của 2 số
*	Tính tích của 2 số
/	Tích thương của 2 số
%	Thực hiện chia có dư của 2 số
++	Tăng giá trị của biến lên 1 đơn vị
--	Giảm giá trị của biến xuống 1 đơn vị



```
int x = 5;  
++x;
```

Toán tử số học

Operator	Example	Same As
=	<code>x = 5</code>	<code>x = 5</code>
+=	<code>x += 3</code>	<code>x = x + 3</code>
-=	<code>x -= 3</code>	<code>x = x - 3</code>
*=	<code>x *= 3</code>	<code>x = x * 3</code>
/=	<code>x /= 3</code>	<code>x = x / 3</code>
%=	<code>x %= 3</code>	<code>x = x % 3</code>

Dùng toán tử +=

```
int x = 10;
```

```
x += 5;
```

```
System.out.println(x);
```

=> 15

Toán tử so sánh

- Toán tử so sánh là các phép toán so sánh hai toán hạng

==	So sánh bằng
>	So sánh lớn hơn
>=	So sánh lớn hơn hoặc bằng
<	So sánh nhỏ hơn
<=	So sánh nhỏ hơn hoặc bằng
!=	So sánh khác

Toán tử logic

- Toán tử logic là các phép toán trên các toán hạng logic

&&	Trả về giá trị true khi tất cả biểu thức tham gia biểu thức có giá trị true
	Trả về giá trị true khi có 1 biểu thức tham gia biểu thức có giá trị là true
!	Lấy giá trị phủ định của biểu thức

Toán tử logic

Using the calculator above, can you compute the result of the following Boolean expressions?

```
true && true;
```

```
true || false;
```

```
true && !false;
```

```
!y || x;
```

where

```
x = true && !false;
```

```
y = false || !false;
```

```
342 / 17 >= 32
```

```
85 / 5 != 21 && true
```

```
x <= 30 && y;
```

where

```
x = 84 / 24;
```

```
y = !false && true;
```

Toán tử logic

Using the calculator above, can you compute the result of the following Boolean expressions?

```
true && true;
```

true



```
true || false;
```

true



```
true && !false;
```

true



```
!y || x;
```

where

```
x = true && !false;
```

```
y = false || !false;
```

true



```
342 / 17 >= 32
```

false



```
85 / 5 != 21 && true
```

true



```
x <= 30 && y;
```

where

```
x = 84 / 24;
```

```
y = !false && true;
```

true



Các hàm xuất ra màn hình

- `System.out.print();` //không xuống dòng
- `System.out.println();` // xuống dòng
- `System.out.printf();` có định dạng, `%d` (số nguyên), `%f` (số thực, mặc định là 6 số, `%.3f` định dạng 3 số), `%s` (chuỗi)

Các hàm xuất ra màn hình

Given the following strings:

```
String a = "I";  
String b = "learn";  
String c = "Java";
```

Select the statement that prints on screen the string "I learn Java" (without quotes).

☐ `System.out.println("a" + "b" + "c");`

☐ `System.out.println("a " + "b " + "c");`

☐ `System.out.println(a + b + c);`

☐ `System.out.println(a + " " + b + " " + c);`

Các hàm xuất ra màn hình

Given the following strings:

```
String a = "I";  
String b = "learn";  
String c = "Java";
```

Select the statement that prints on screen the string `"I learn Java"` (without quotes).

☐ `System.out.println("a" + "b" + "c");`

☐ `System.out.println("a " + "b " + "c");`

☐ `System.out.println(a + b + c);`

☒ `System.out.println(a + " " + b + " " + c);`

☐ `System.out.println(a + + b + + c);`

Nhập từ bàn phím

- Tạo đối tượng Scanner

```
Scanner scanner = new Scanner(System.in);
```

- Các phương thức thường dùng:
- `scanner.nextLine()`: nhập một dòng
- `scanner.nextInt()`: nhập số nguyên
- `scanner.nextDouble()`: nhập số thực

Các hàm toán học

- Các hàm được cung cấp sẵn để thực hiện các phép tính: làm tròn số, tính căn bậc 2, tính lũy thừa,..
- Ví dụ: `double a = Math.sqrt(7);`

Các hàm toán học

Hàm	Diễn giải	Ví dụ
Math.min(a, b)	Lấy số nhỏ nhất của 2 số a và b	$x = \text{Math.min}(5, 3.5) \Rightarrow x = 3.5$
Math.max(a, b)	Lấy số lớn nhất của 2 số a và b	$x = \text{Math.max}(5, 3.5) \Rightarrow x = 5$
Math.pow(a, n)	Tính a^n (a lũy thừa n)	$x = \text{Math.pow}(5, 3) \Rightarrow x = 75$
Math.sqrt(a)	Tính \sqrt{a} (căn bậc 2 của a)	$x = \text{Math.sqrt}(16) \Rightarrow x = 4$
Math.abs(a)	Lấy giá trị tuyệt đối của a	$x = \text{Math.abs}(-5) \Rightarrow x = 5$
Math.ceil(a)	Lấy số nguyên trên của a	$x = \text{Math.ceil}(3.5) \Rightarrow x = 4$
Math.floor(a)	Lấy số nguyên dưới của a	$x = \text{Math.floor}(3.5) \Rightarrow x = 3$
Math.round(a)	Làm tròn số của a	$x = \text{Math.round}(3.5) \Rightarrow x = 4$
Math.random()	Sinh số ngẫu nhiên từ 0 đến 1	$x = \text{Math.random}() \Rightarrow x = 0..1$

Bài tập tuần 1

Nhập vào số nguyên n , in ra màn hình kết quả hình tam giác như ví dụ dưới đây. Sau đó yêu cầu nhập lại n , khi $n=0$ thì dừng chương trình.

Ví dụ: $n=9$

```
      1
     2  2
    3  3  3
   4  4  4  4
  5  5  5  5  5
 6  6  6  6  6  6
 7  7  7  7  7  7  7
 8  8  8  8  8  8  8  8
 9  9  9  9  9  9  9  9  9
```