



Java™

LẬP TRÌNH JAVA 1

BÀI 1: LÀM QUEN VỚI JAVA, PHẦN 1

- ☐ Hiểu ngôn ngữ lập trình Java
- ☐ Biết cách thiết lập môi trường cho ứng dụng java
- ☐ Nắm cấu trúc chương trình Java
- ☐ Sử dụng công cụ NetBean
- ☐ Biết cách nhập dữ liệu từ bàn phím
- ☐ Biết cách xuất dữ liệu ra màn hình
- ☐ Biết cách thực hiện các phép toán số học
- ☐ Hiểu rõ và sử dụng kiểu nguyên thủy, lớp bao
- ☐ Chuyển đổi chuỗi sang kiểu nguyên thủy
- ☐ Hiểu và sử dụng toán tử, xây dựng biểu thức

- ☐ Giới thiệu ngôn ngữ lập trình Java
- ☐ Thiết lập môi trường cho ứng dụng java
- ☐ Giới thiệu cấu trúc chương trình Java
- ☐ Giới thiệu công cụ NetBean
- ☐ Nhập dữ liệu từ bàn phím
- ☐ Xuất dữ liệu ra màn hình
- ☐ Thực hiện các phép toán số học đơn giản

❑ Java là ngôn ngữ lập trình có các đặc điểm sau

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



Write once, run anywhere

LỊCH SỬ PHÁT TRIỂN CỦA JAVA

1991

Ra đời với tên gọi Oak bởi Sun Microsystem

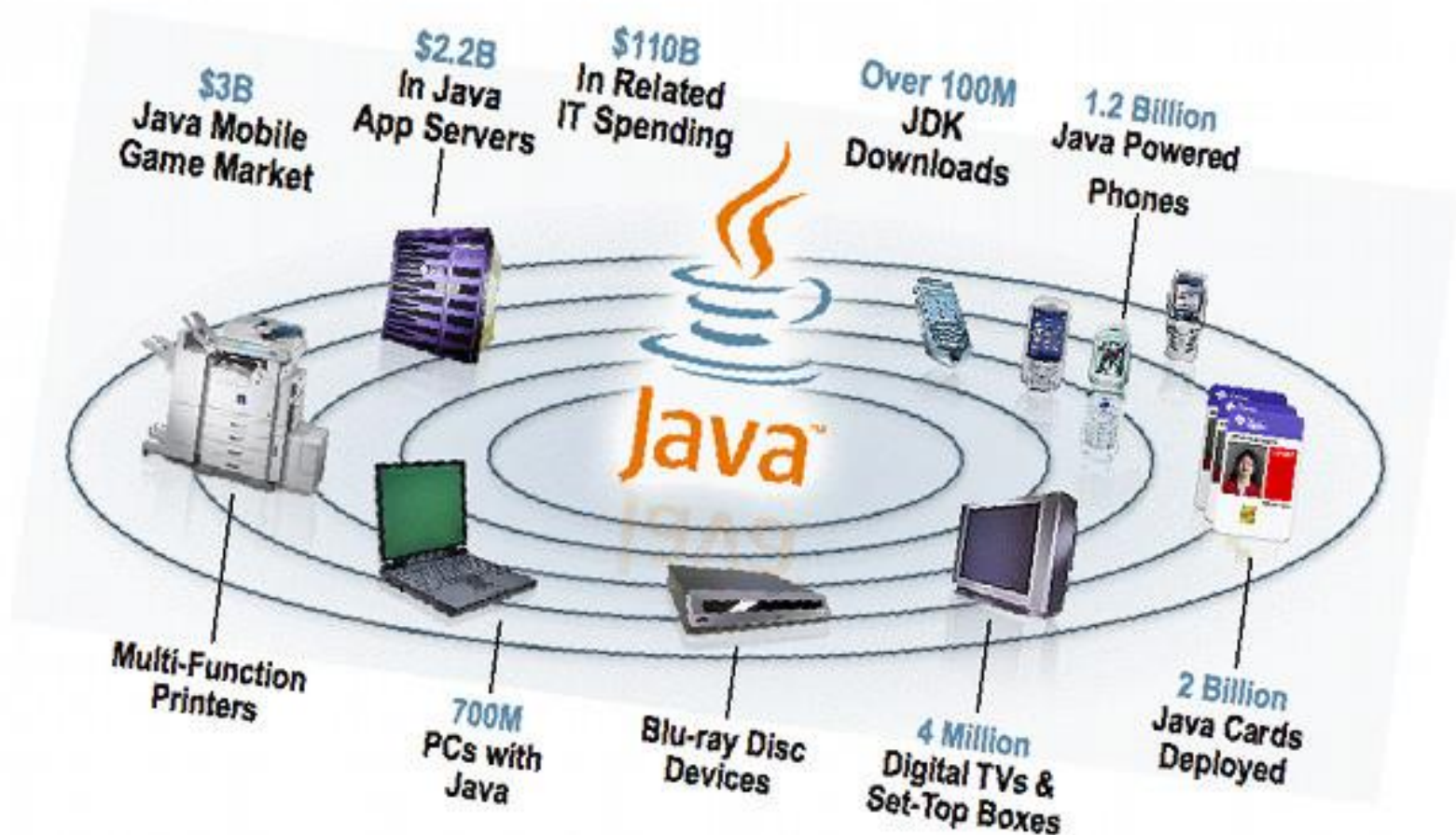
1995

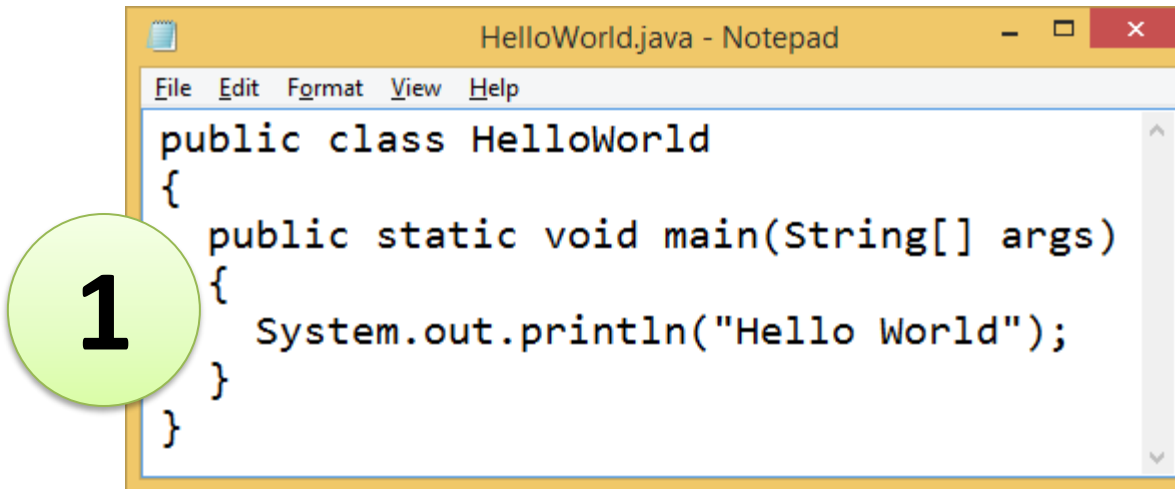
Đổi tên thành Java

2010

Oracle mua lại

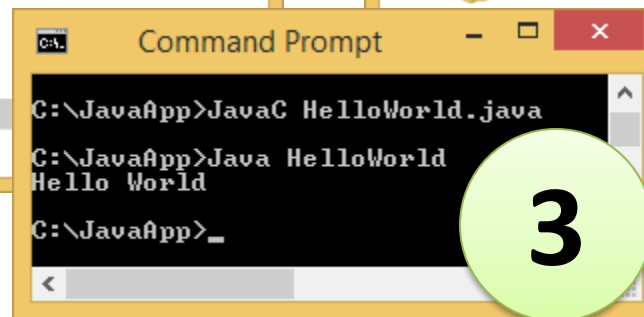
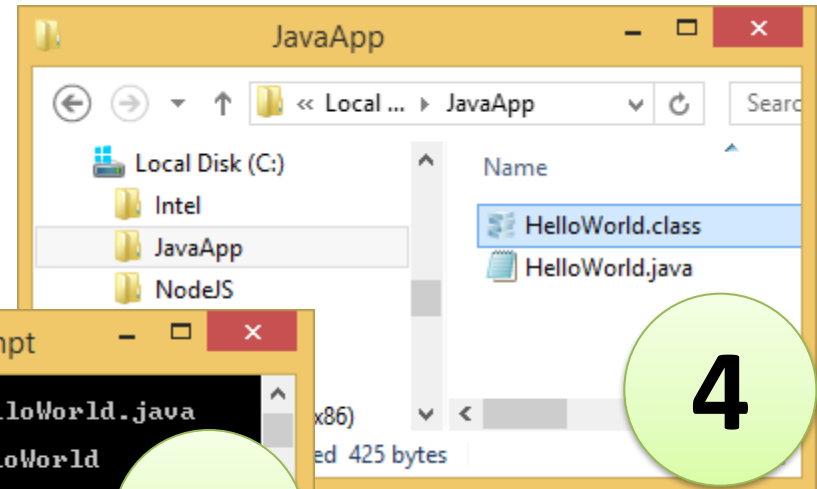
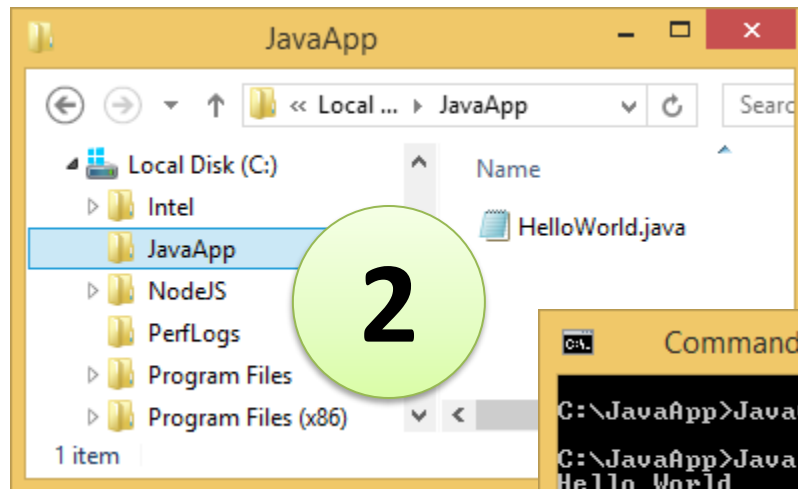
❑ Học Java có thể làm ra những sản phẩm gì?





```

File Edit Format View Help
public class HelloWorld
{
    public static void main(String[] args)
    {
        System.out.println("Hello World");
    }
}
    
```



Xem casestudy 1

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

Lưu thành

HelloWorld.java

❑ **com.poly**: tên gói chứa lớp

❖ Sử dụng ký tự thường và dấu chấm. Có thể xem package như folder còn class như file.

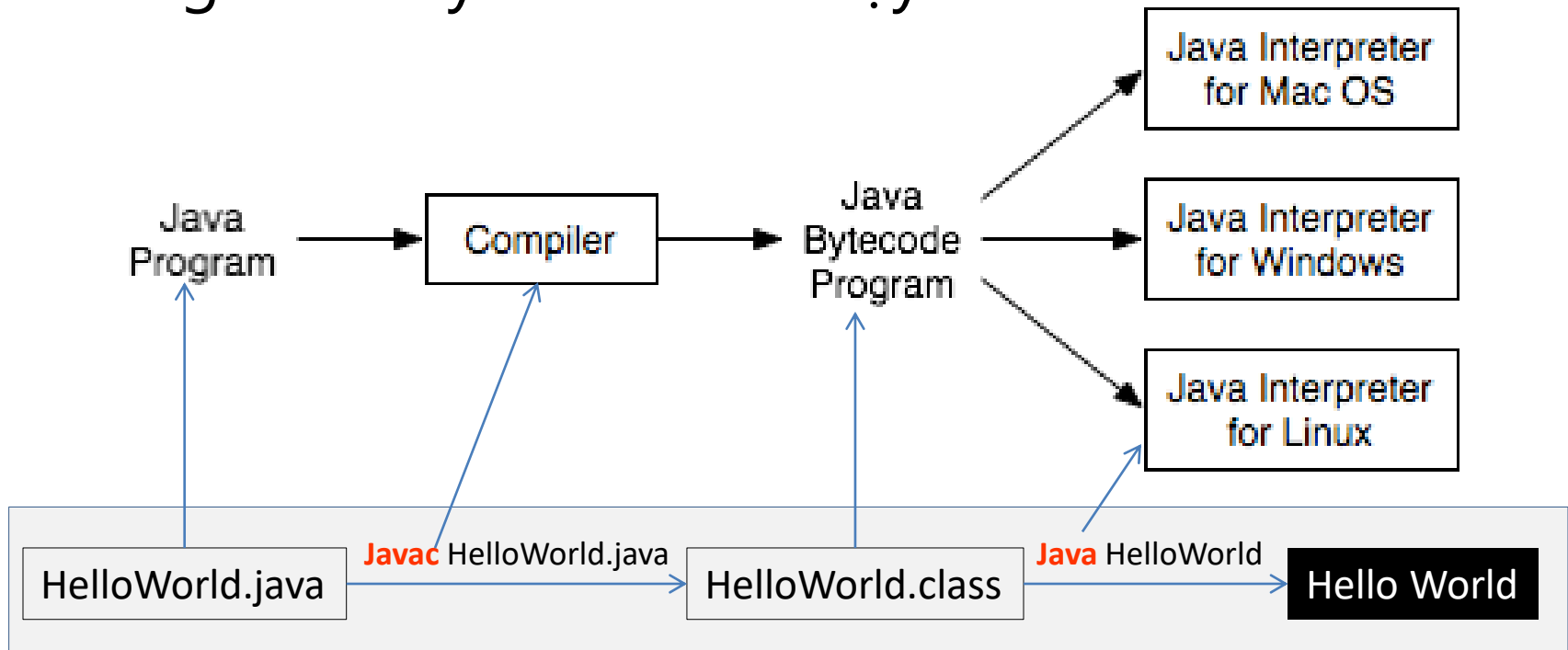
❑ **HelloWorld**: tên lớp

❖ Phải giống tên file java. Viết hoa ký tự đầu của mỗi từ

❑ **main()**: phương thức bắt đầu chạy

❖ Lớp có thể có nhiều phương thức nhưng **main()** được gọi tự động khi ứng dụng chạy

- ❑ Khác với ngôn ngữ lập trình khác, thay vì biên dịch mã nguồn thành mã máy, Java được thiết kế biên dịch mã nguồn thành bytecode
- ❑ Bytecode sau đó được môi trường thực thi dịch sang mã máy trước khi chạy



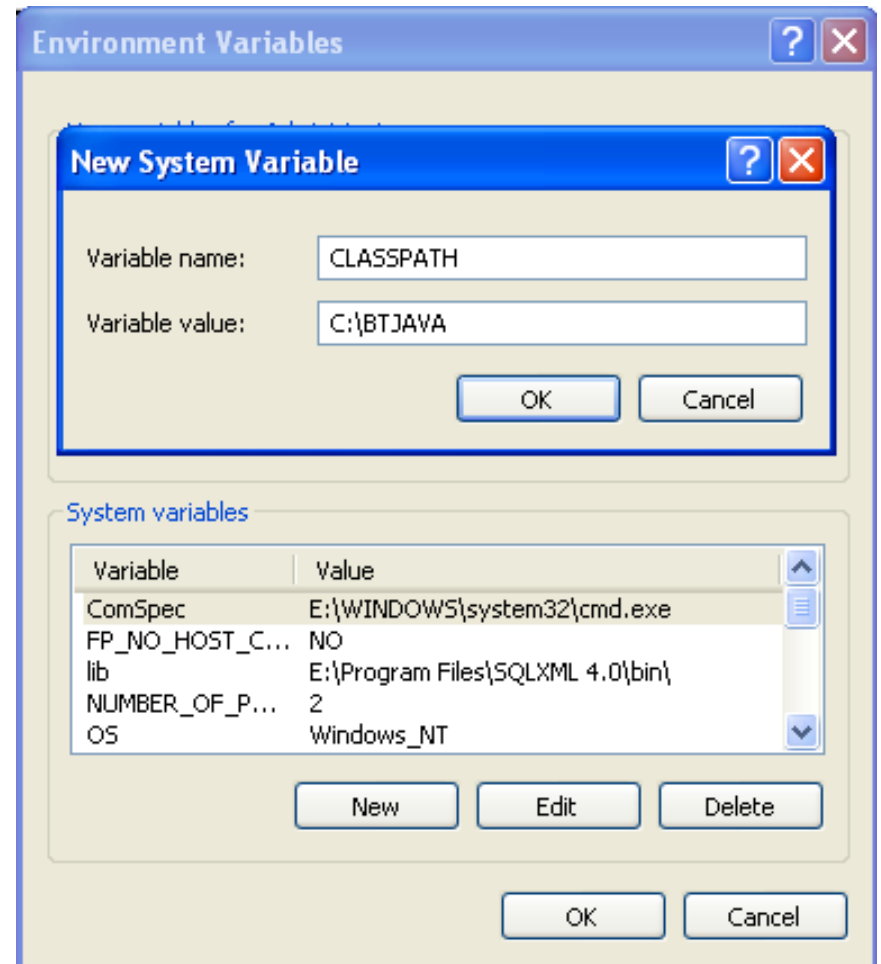
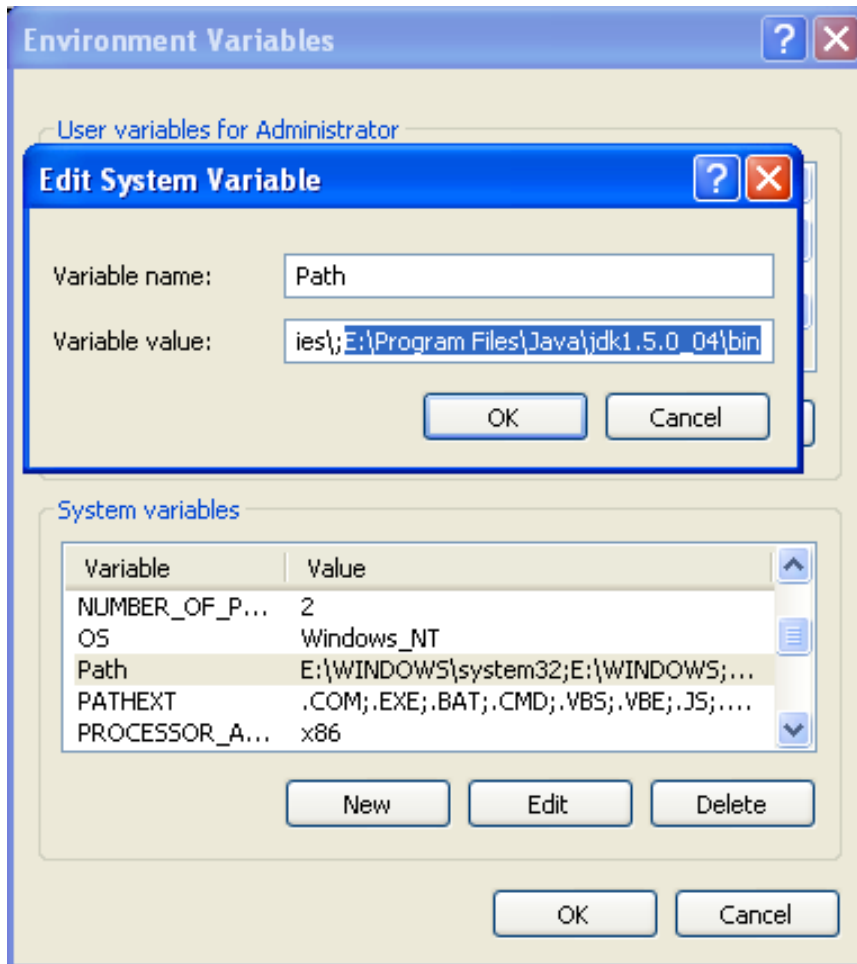
- ❑ JDK và các công cụ (javac, java)
- ❑ Cấu hình JDK (path, classpath)



PATH

Xem casestudy 2

CLASSPATH



Hỗ trợ việc phát triển và triển khai ứng dụng dễ dàng hơn



Xem casestudy 3



DEMO

Cài đặt và khởi tạo dự án Java



- ❑ `System.out.print()`: Xuất xong không xuống dòng
- ❑ `System.out.println()`: Xuất xong có xuống dòng
- ❑ `System.out.printf()`: Xuất có định dạng, các ký tự định dạng
 - `%d`: số nguyên
 - `%f`: số thực
 - ✓ Mặc định là 6 số lẻ
 - ✓ `%.3f` định dạng 3 số lẻ
 - `%s`: chuỗi

FPT Polytechnic
Đào tạo 12 nghề

❑ Ví dụ

```
System.out.print("FPT ");  
System.out.println("Polytechnic");  
System.out.printf("Đào tạo %d nghề", 12);
```



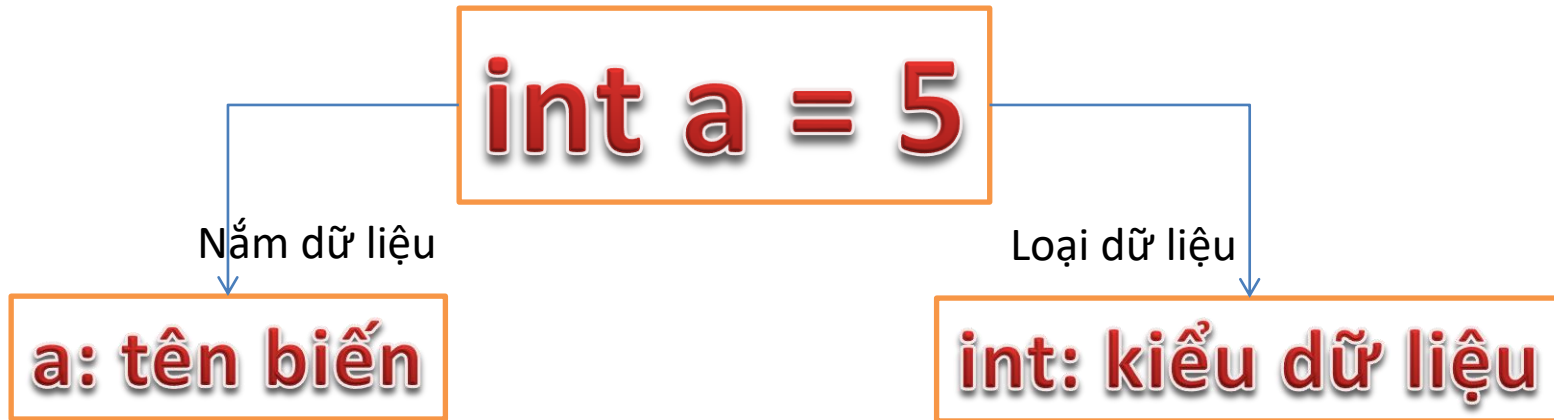
DEMO

Viết chương trình in ra màn hình dòng
chữ “Xin chào! Polytechnic”



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

- ❑ Đoạn mã trên gán các giá trị 5 cho a, 7 cho b và tổng a + b cho c sau đó xuất tổng ra màn hình
- ❑ a, b và c gọi là biến số nguyên
 - ❖ Biến là thành phần nắm giữ dữ liệu được chương trình sử dụng trong các biểu thức
 - ❖ Mỗi biến có kiểu dữ liệu riêng



❑ Biến là thành phần
nhập giữ dữ liệu
được chương trình
sử dụng trong các
biểu thức tính toán
(*biến a nhập giữ số 5*)

- ❑ int: Số nguyên
- ❑ double : số thực
- ❑ String: Chuỗi
- ❑ ...

❑ Cú pháp

<kiểu dữ liệu> <tên biến> [=giá trị khởi đầu];

❑ Ví dụ:

`int a;` // khai báo biến không khởi đầu giá trị

`double b = 5;` // khai báo biến có khởi đầu giá trị

❑ Khai báo nhiều biến cùng kiểu

`int a, b=5, c;`

❑ Gán giá trị cho biến

`c = 9;`

`a = 15;`

- ❑ Sử dụng ký tự **alphabet**, **số**, **\$** hoặc gạch dưới (**_**).
Không ~~bắt đầu bởi số~~, không sử dụng ~~từ khóa~~
- ❑ Tên có phân biệt **HOA/thường**

** Từ khóa là các từ được sử dụng để xây dựng ra ngôn ngữ lập trình java*

abstract	assert	boolean	break	byte	case
catch	char	class	const	continue	default
do	double	else	enum	extends	final
finally	float	for	goto	if	implements
import	instanceof	int	interface	long	native
new	package	private	protected	public	return
short	static	strictfp	super	switch	synchronized
this	throw	throws	transient	try	void
volatile	while				

TÊN BIẾN NÀO SAU ĐÂY KHÔNG HỢP LỆ

A •abc

B •1abc *Sai vì bắt đầu bởi số*

D •ab_c

E •_abc

F •\$abc

G •ab-c *Sai vì chứa dấu gạch ngang*

H •_123

I •\$123

J •if *Sai vì sử dụng từ khóa*

K •\$if

- ❑ java.util.Scanner cho phép nhận dữ liệu từ bàn phím một cách đơn giản
- ❑ Tạo đối tượng Scanner
 - ❖ Scanner scanner = new Scanner(System.in)
- ❑ Các phương thức thường dùng
 - ❖ scanner.**nextLine()**
 - Nhận 1 dòng nhập từ bàn phím
 - ❖ scanner.**nextInt()**
 - Nhận 1 số nguyên nhập từ bàn phím
 - ❖ scanner.**nextDouble()**
 - Nhận 1 số thực nhập từ bàn phím



DEMO

Khai báo 2 biến số nguyên a, b và c
Thực hiện phép cộng a và b được c
Xuất kết quả c





DEMO

Viết chương trình tìm điểm trung bình của các môn học a, b, c. Trong đó môn a được tính hệ số 2.



- ☐ Lab 1 – bài 1
- ☐ Lab 1 – bài 2



Java™

LẬP TRÌNH JAVA 1

BÀI 1: LÀM QUEN VỚI JAVA, PHẦN 2

Toán tử	Diễn giải	Ví dụ
+	Phép cộng	<code>int a = 5 + 7</code>
-	Phép trừ	<code>int b = 9 - 6</code>
*	Phép nhân	<code>double c = 9.5 * 2</code>
/	Phép chia	<code>double d = 3.5 / 5</code>

❑ Toán tử số học được sử dụng để thực hiện các phép toán số học

❑ Thứ tự ưu tiên

1. Nhân và chia
2. Cộng và trừ
3. Trái sang phải

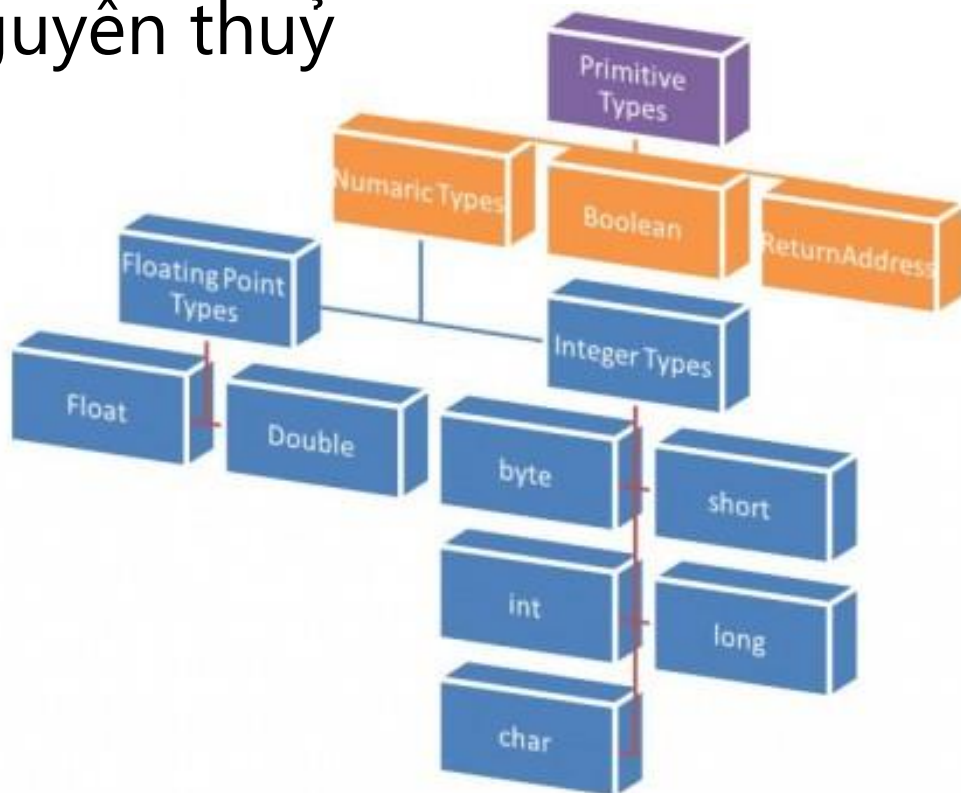
$5 + 7 * 2 - 4 / 2$

???

- ❑ Kiểu dữ liệu nguyên thủy là kiểu được giữ lại từ ngôn ngữ C (ngôn ngữ gốc của Java)
- ❑ Có 8 kiểu dữ liệu nguyên thủy
- ❑ Ví dụ

❖ `int a = 8;`

❖ `double b;`



Kiểu	Mặc định	Bit	Khả năng lưu trữ	
			Giá trị nhỏ nhất	Giá trị lớn nhất
byte	0	8	-2^7	$+2^7-1$
short	0	16	-2^{15}	$+2^{15}-1$
int	0	32	-2^{31}	$+2^{31}-1$
long	0L	64	-2^{63}	$+2^{63}-1$
float	0.0F	32	$-3.40292347 \times 10^{38}$	$+3.40292347 \times 10^{38}$
double	0.0	64	$-1.79769313486231570 \times 10^{308}$	$+1.79769313486231570 \times 10^{308}$
char	'\u0000'	16	'\u0000'	'\uFFFF'
boolean	false	1	false	true

Giá trị mặc định là giá trị sẽ được gán cho biến khi khai báo không khởi đầu giá trị cho biến

- ❑ Giá trị hằng là dữ liệu có kiểu là một trong các kiểu nguyên thủy

Kiểu int

```
int i = 3;
```

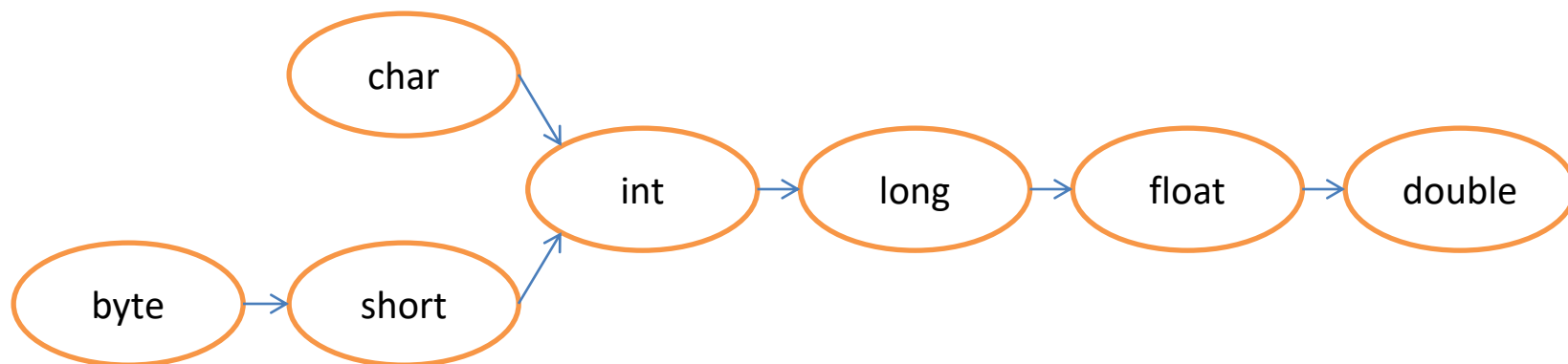
Kiểu long

```
long l = 12L;
```

Kiểu float

```
float = 10.19F;
```

- ❑ Đối với kiểu nguyên thủy, ép kiểu tự động xảy ra theo chiều mũi tên



- ❑ 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
```

❑ Xét biểu thức 1

```
String a = "3";
```

```
String b = "4";
```

```
String c = a + b;
```

=> c là **"34"**

❑ Xét biểu thức 2

```
int a = Integer.parseInt("3");
```

```
int b = Integer.parseInt("4");
```

```
int c = a + b;
```

=> c là **7**

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

❑ Xét trường hợp

```
int a = scanner.nextInt();
```

hoặc

```
int a = Integer.parseInt(s);
```

❑ Điều gì sẽ xảy ra khi người dùng **nhập không phải số** hoặc chuỗi **s không phải là chuỗi chứa số**

❑ Hãy sử dụng lệnh **try...catch** để kiểm soát các lỗi trên

```
try{  
    int a = scanner.nextInt();  
    System.out.println("Bạn đã nhập đúng");  
}  
catch (Exception ex){  
    System.out.println("Vui lòng nhập số !");  
}
```


- ❑ Tương ứng với mỗi kiểu nguyên thủy Java định nghĩa một lớp bao để bao giá trị của kiểu nguyên thủy tương ứng gọi là lớp bao kiểu nguyên thủy
- ❑ Rất nhiều hàm trong Java chỉ làm việc với 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

❑ Boxing là việc tạo đối tượng từ lớp bao để bọc giá trị nguyên thủy.

❖ Có 3 cách để bao giá trị nguyên thủy sau

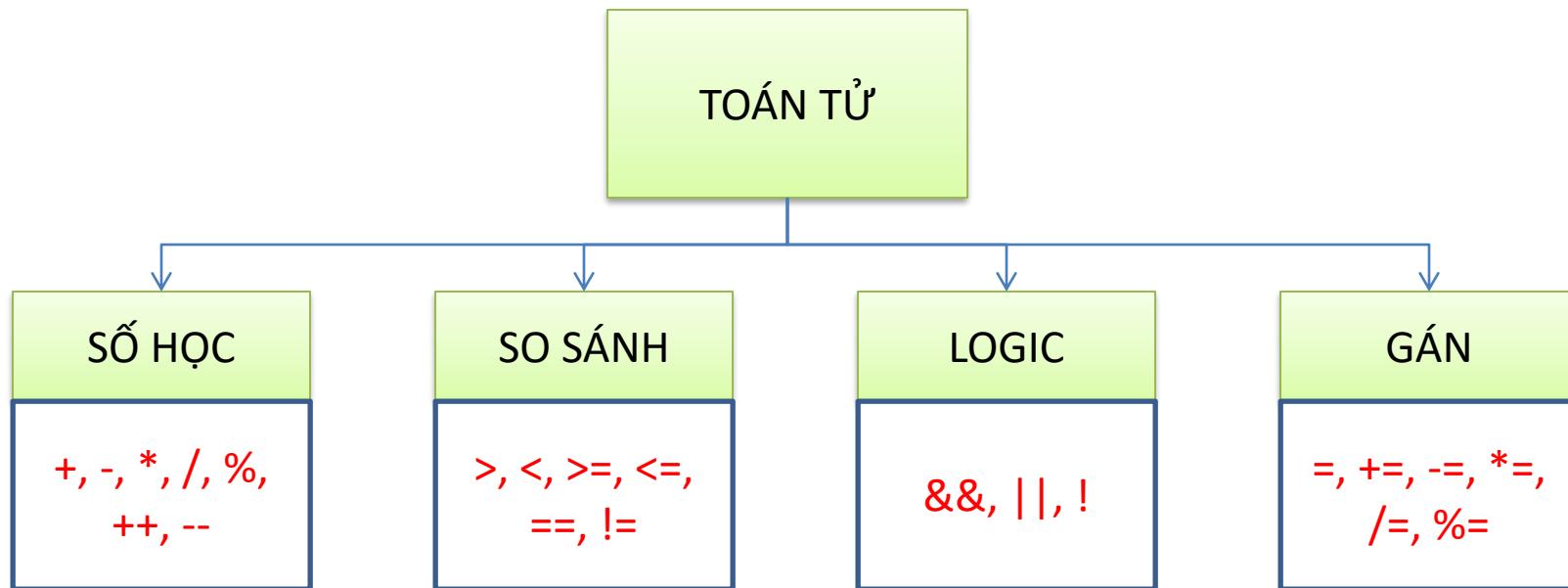
- Integer a = Integer.**valueOf**(5) // bao tường minh
- Integer a = **new Integer**(5) // bao tường minh
- Integer a = **5** // bao ngầm định

❑ Unboxing là việc mở lấy giá trị nguyên thủy từ đối tượng của lớp bao

❖ Có 2 cách mở bao để lấy giá trị nguyên thủy sau

- int b = a.**intValue**() // mở bao tường minh
- int b = **a**; // mở bao ngầm định

Boxing	Unboxing	Ví dụ
Byte.valueOf(byte)	<<Byte>>.byteValue()	long a = 5L; Long b = Long.valueOf(a); long c = b.longValue();
Short.valueOf(short)	<<Short>>.shortValue()	
Integer.valueOf(int)	<<Integer>>.intValue()	
Long.valueOf(long)	<<Long>>.longValue()	
Float.valueOf(float)	<<Float>>.floatValue()	
Double.valueOf(double)	<<Double>>.doubleValue()	
Boolean.valueOf(boolean)	<<Boolean>>.booleanValue()	



Biểu thức là sự kết hợp giữa toán tử và toán hạng. Kết quả của biểu thức là một giá trị.

Giá trị của các biểu thức sau?

`int x = 11 % 4;`

`boolean a = 9 < 2 && true || 4 > 3;`

❑ 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

❑ 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 là các phép toán thao tác 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 điều kiện là toán tử 3 ngôi duy nhất trong ngôn ngữ Java
- ❑ Cú pháp:
<điều kiện> ? <giá trị đúng> : <giá trị sai>
- ❑ Diễn giải:
 - ❖ Nếu biểu thức **<điều kiện>** có giá trị là true thì kết quả của biểu thức là **<giá trị đúng>**, ngược lại là **<giá trị sai>**
- ❑ Ví dụ: tìm số lớn nhất của 2 số a và b

```
int a = 1, b = 9;  
int max = a > b ? a : b;
```

Tìm số lớn nhất trong 3 số a, b và c?



DEMO

Khai báo 2 biến hoten và tuoi
Sử dụng cả 3 hàm trên để xuất dòng sau
<<hoten>> năm nay <<tuoi>> tuoi





DEMO

Khai báo 2 biến hoten và tuoi
Nhập họ tên và tuổi từ bàn phím
Xuất ra dòng theo định dạng sau
<<hoten>> năm nay <<tuoi>> tuoi



- ❑ Java cung cấp các hàm tiện ích giúp chúng ta thực hiện các phép tính khó một cách dễ dàng như:
 - ❖ Làm tròn số
 - ❖ Tính căn bậc 2
 - ❖ Tính lũy thừa
 - ❖ ...
- ❑ Ví dụ sau đây tính căn bậc 2 của 7
 - ❖ `double a = Math.sqrt(7)`
- ❑ Ngoài `Math.sqrt()` còn rất nhiều hàm khác được trình bày ở slide sau.

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	$x = \text{Math.random()} \Rightarrow x = 0..1$

Thảo luận: Làm tròn một số thực với 2 số lẻ



DEMO

1. Sinh số ngẫu nhiên từ 5 đến 12
Xuất số đó và căn bậc 2 của nó ra màn hình
2. Nhập 2 số thực a và b từ bàn phím
Tính và xuất a lũy b, giá trị nhỏ nhất của 2 số



- ❑ Giới thiệu Java
- ❑ Thiết lập môi trường làm việc (JDK) và IDE
- ❑ Biến và quy tắc đặt tên biến
- ❑ Toán tử số học
- ❑ Xuất ra màn hình
- ❑ Nhập từ bàn phím
- ❑ Sử dụng các hàm toán học



- ☐ Lab 1 – bài 3
- ☐ Lab 1 – bài 4
- ☐ Lab 1 – bài 5 (giáo viên cho thêm)