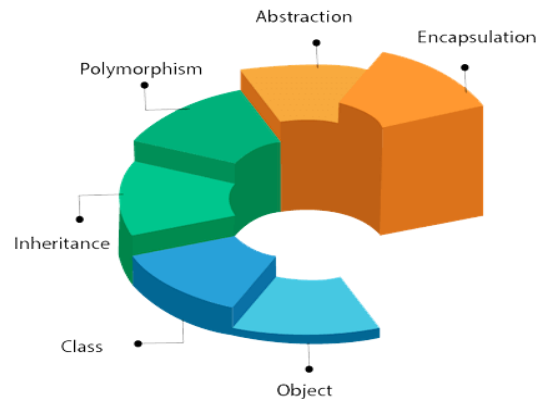


Lập Trình Hướng Đối Tượng – CT176



TS. Phan Thượng Cang

ptcang@cit.ctu.edu.vn

Khoa CNTT&TT-Đại học Cần Thơ

Phần 1



Các Thành Phần Cơ Bản của Java

Kiểu - Biến - Phép toán - Xuất nhập
Cấu trúc điều khiển - Phương thức - Tái định nghĩa

TS. Phan Thượng Cang
Khoa CNTT&TT-Đại học Cần Thơ

Ngôn ngữ lập trình Java



● Lịch sử:

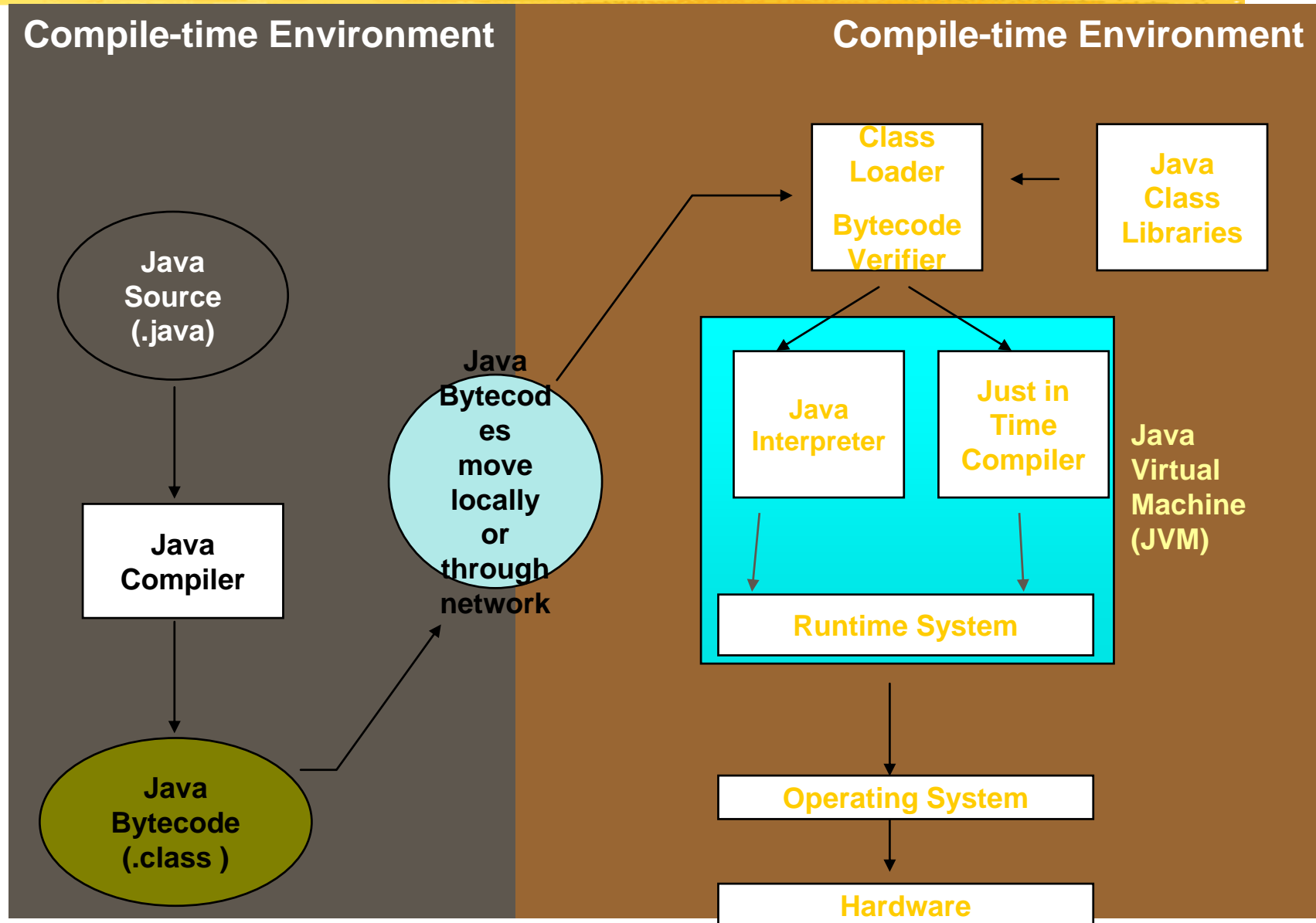
- Java được phát triển bởi Sun Microsystems do James Gosling (1995). Phiên bản mới nhất của Java (Oracle/Open JDK) là 16 (đến 8/2021).
- Ngôn ngữ đa nền tảng (multi platform) nên Java được dùng rộng rãi trên nhiều thiết bị từ máy tính đến mobile và nhiều thiết bị phần cứng khác, cho nhiều ứng dụng (Desktop, Web, Mobile).

● Đặc điểm:

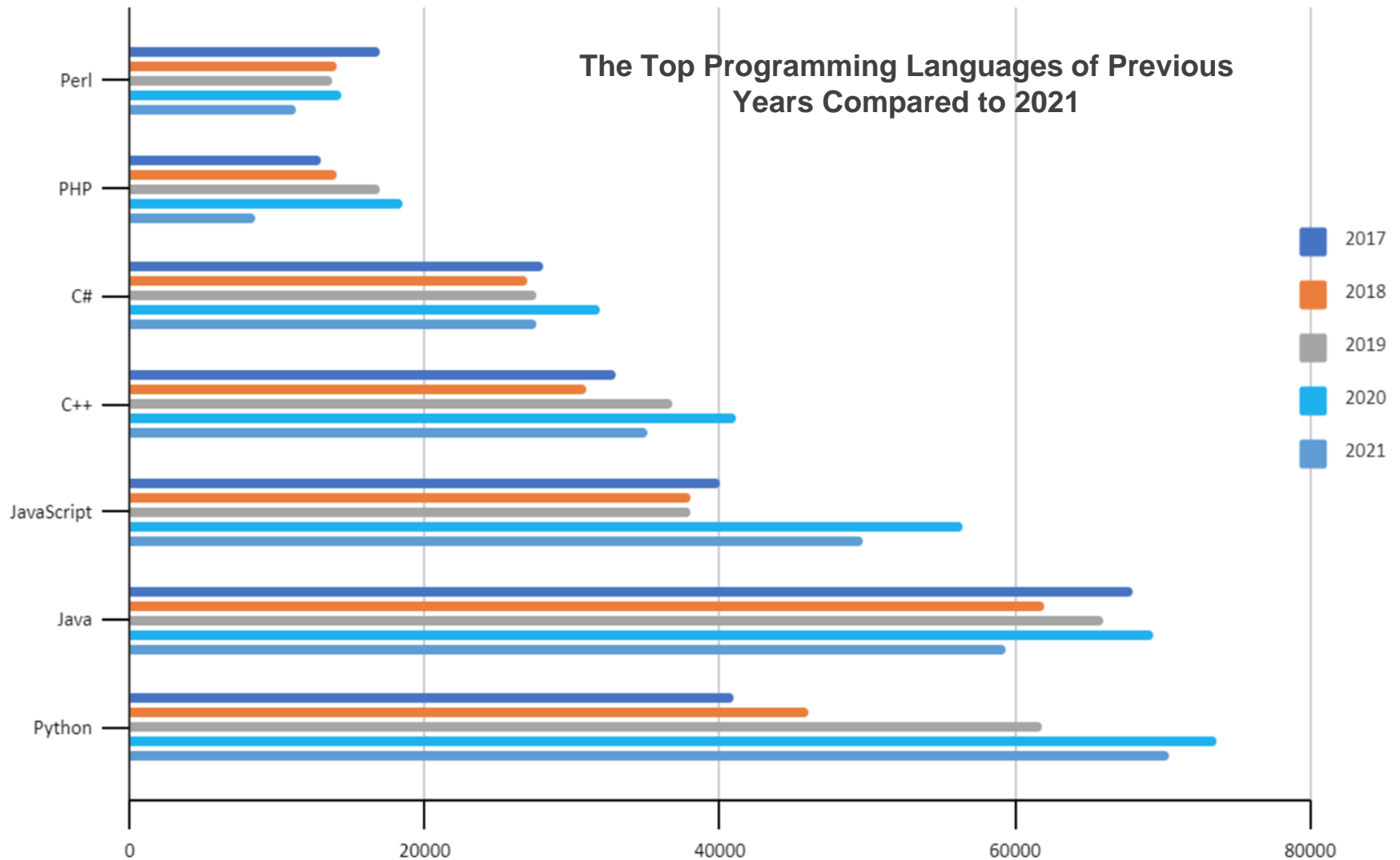
- Ngôn ngữ lập trình hướng đối tượng.
- Độc lập nền.
- Đơn giản, Bảo mật, Linh hoạt (thích ứng nhiều môi trường phát triển).
- Đa luồng và Hiệu suất cao (garbage collection).



How it works...!



Top các ngôn ngữ lập trình



Java

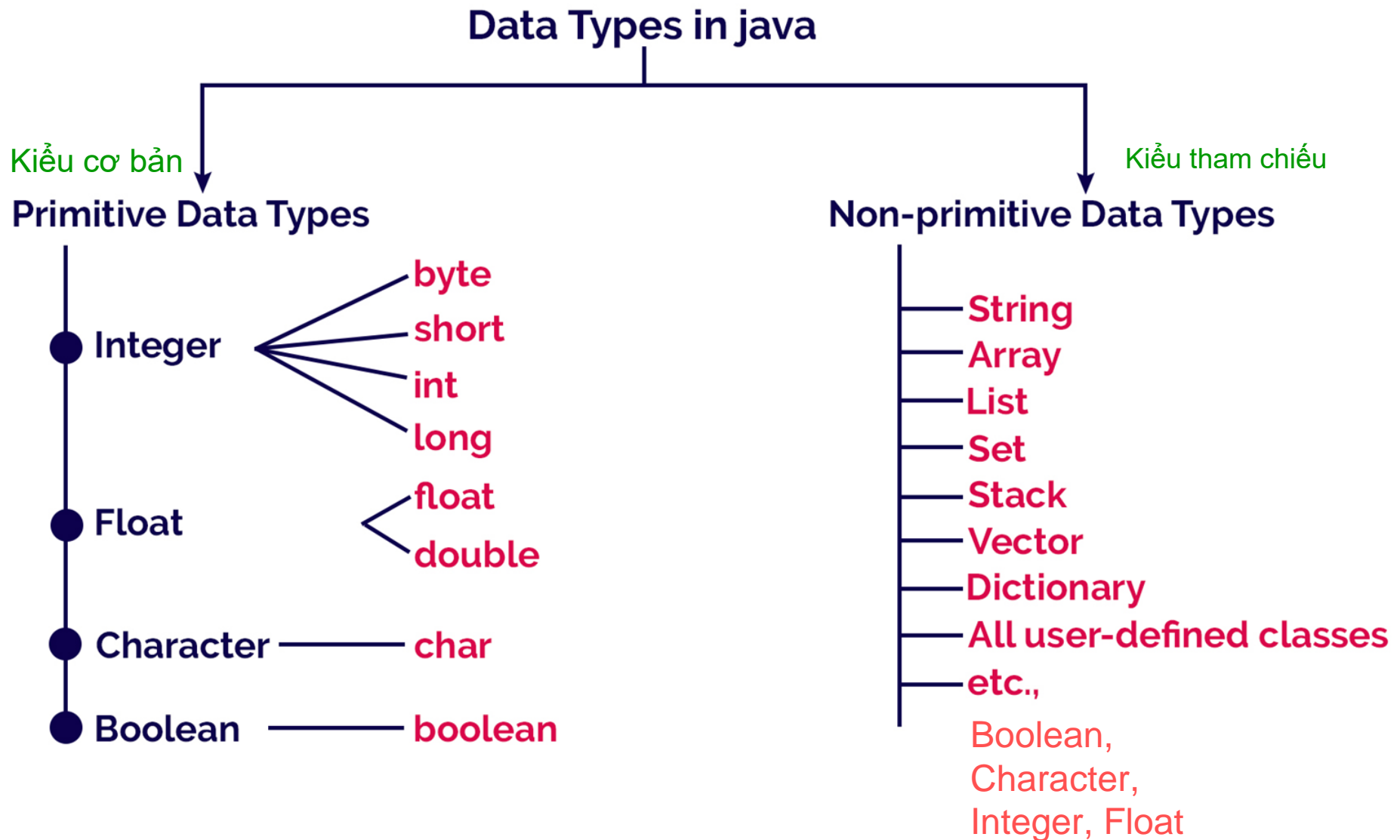


Từ khóa

Java gồm một tập các từ khóa mà không được sử dụng làm tên biến, phương thức, lớp hoặc bất kỳ id nào khác:

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	true	false
null			

Kiểu dữ liệu



Biến – Hằng số

- Khai báo biến:

<kiểu d.liệu> <tên_biến> [=giá trị] ;

<kiểu d.liệu> <tên_biến1>, <tên_biến2> ;

Ví dụ:

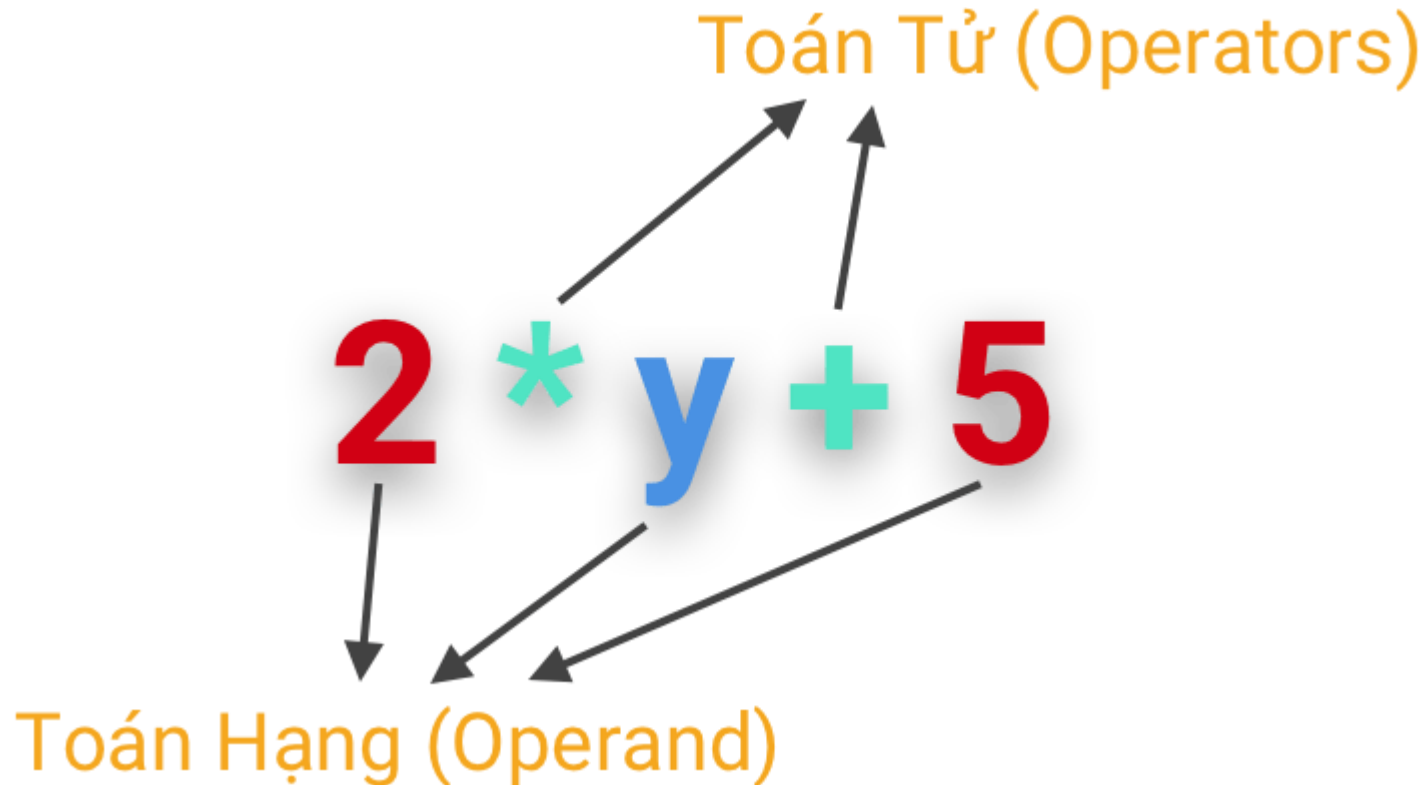
- boolean b1, b2 = true, b3=false ; char c = 'c';
- int a=10, b; long l = 10000;
- float f = 3.1f ; double d=3.1;
- String s1="CNTT" , s2 = s1 + " – K46 ";

- Khai báo hằng số:

final <kiểu d.liệu> <tên_biến> = <giá_trị> ;

Ví dụ: final int N = 10;

Biểu thức



Toán tử (operator)

- Toán tử gán: =
- Toán tử số học: +, -, *, /, %

Ví dụ:

- `int a = 10, b = 3, c = a + b, d = a / b, e = a % b ;`
- `String s1 = "CNTT", s2 = s1 + " - K46 ";`

- Toán tử viết tắt: +=, -=, *=, /=, %=
- Toán tử tăng giảm: ++, --

Ví dụ:

- `int a = 10;`
`int c = a++ ;`
`int d = ++a ;`
`=> c = ? , d = ?`

Toán tử (2)

- **Toán tử quan hệ:** `==, !=, >, >=, <, <=`

Ví dụ:

- `int b=13, c=2;`
`if(b%2 ==0) b=2;`
`if (b==c) c = c/2;`

- **Toán tử logic:** `&&, ||, !`

Ví dụ:

- `int a= 10, b =60;`
`if(a % 2 ==0 && b % 2 ==0) a += b;`
`if(!x) b=10; else b = 20; => x =??`

Ép kiểu

- **Ép kiểu tự động**

Ví dụ:

- `long l = 100;`
`float f = l; // f = 100.0`

- **Ép kiểu tường minh**

Ví dụ:

- `double d = 7.5;`
`int i = (int) d; // i = 7`

`float f = i/2; // f = ??`

`f = i/2.0;`

`f = ((float) i)/2;`

Xuất / Nhập

- **Xuất ra màn hình:**

System.out.print (biểu thức);

Ví dụ:

- String s = "CNTT-TT" ;
System.out.print(" \n s = " + s);
System.out.println("Hello " + s + " - K46");

- **Nhập từ bàn phím:**

- Import gói: `import java.util.Scanner;`
- Tạo biến sc: `Scanner sc = new Scanner (System.in);`
- Nhập dữ liệu: `boolean b = sc.nextBoolean();`
 - `int n = sc.nextInt(); float f = sc.nextFloat();`
 - `String s = sc.nextLine();`
 - `char c = sc.nextLine().charAt(0);`

Viết chương trình trong Java

```
package ctu.cit;           // khai báo gói
import java.util.Scanner;   // thêm gói thư viện

public class Hello { // tên lớp trùng tên file

    public static void main(String[] args) {
        int n = 0;
        System.out.println("\n Nhập nam = " );
        Scanner sc = new Scanner (System.in);
        n = sc.nextInt();
        System.out.println("\n Hello World, " + n);
    }
}
```

Cấu trúc điều khiển

- **Cấu trúc điều kiện if:**

```
if(<biểu_thức_điều_kiện>){
```

```
    <khởi_lệnh_1>;
```

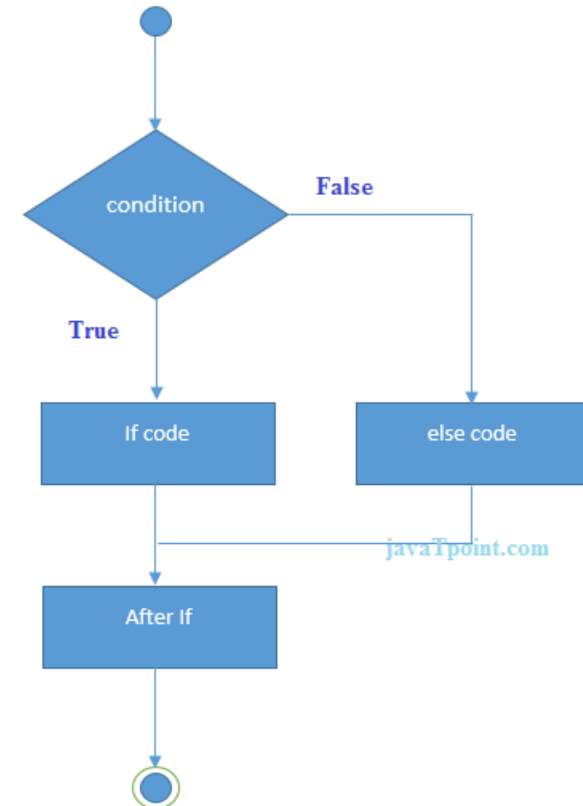
```
}
```

```
else {
```

```
    <khởi_lệnh_2>;
```

```
}
```

Hoặc (b.thức_điều_kiện) ? v1 : v2 ;



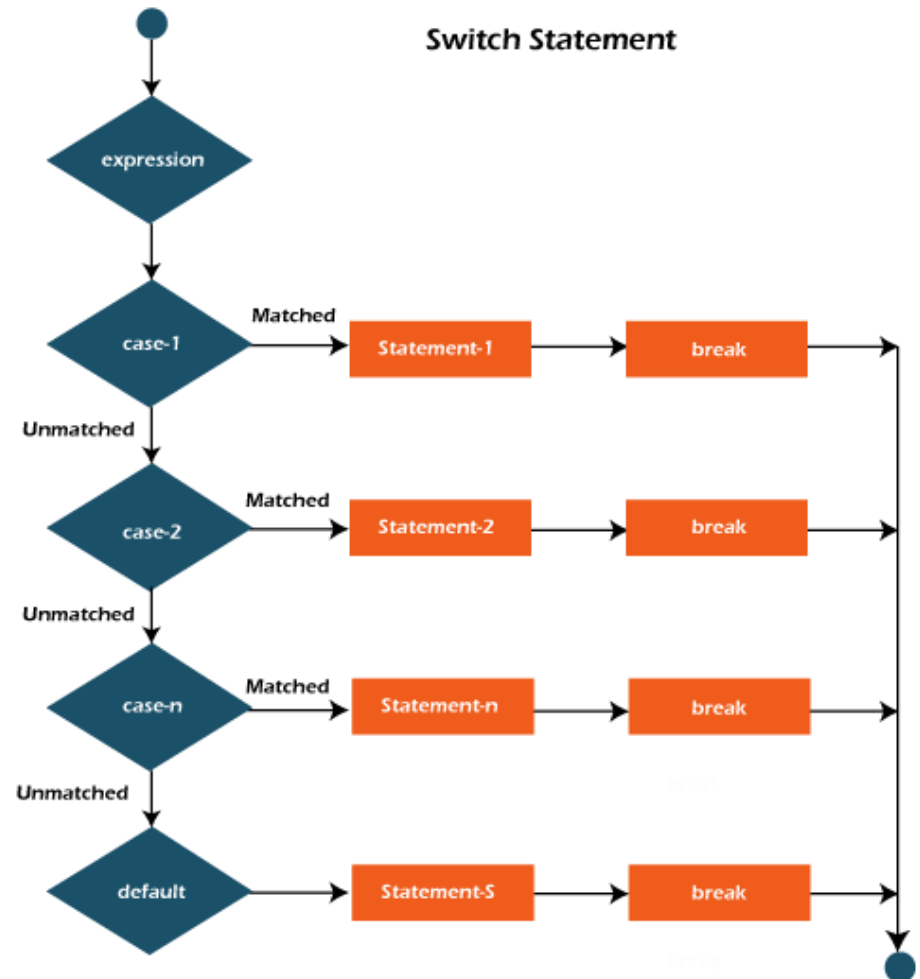
Ví dụ:

- `if(a > 0) System.out.print(a + “ là số dương ”);`
`else System.out.print(a + “ là số âm ”);`
- `c = (a > b) ? a : b;`

Cấu trúc điều khiển

- Cấu trúc điều kiện switch:

```
switch ( tên_biến ) {  
    case value1:  
        // khối lệnh 1 ...  
        break;  
    case value2:  
        // khối lệnh 2 ...  
        break;  
    default:  
        // khối lệnh n ...  
}
```



Cấu trúc điều khiển

- **Cấu trúc điều kiện switch:**

Ví dụ:

```
int day = 4;  
switch (day) {  
    case 6:  
        System.out.println("Hôm nay thứ sáu");  
        break;  
    case 7:  
        System.out.println("Hôm nay thứ bảy");  
        break;  
    default:  
        System.out.println("Hôm nay cuối tuần");  
}
```

Cấu trúc điều khiển

- **Cấu trúc lặp for:**

```
for(<bthức_khởi_tạo>; <bthức_điều_kiện>; <bthức_lặp>){  
    <khối_lệnh>;  
}
```

Ví dụ:

```
int n = 400000, tong = 0;  
// tong = 1 + 2 + 3 + 4 + .... + 400000  
//           i       i               i  
for (int i=1; i<=n; i++) {  
    tong = tong + i;  
}
```

```
System.out.println("tong = " + tong);
```

Cấu trúc điều khiển

- **Cấu trúc lặp for:**

```
for(<bthức_khởi_tạo>; <bthức_điều_kiện>; <bthức_lặp>){  
    <khối_lệnh>;  
}
```

Ví dụ:

```
int n = 400000, tong = 0;  
// tong = 1 + 2 + 3 + 4 + .... + 400000  
//           i       i       i  
for (int i=2; i<=n; i+=2) {  
    tong = tong + i;  
}
```

```
System.out.println("tong = " + tong);
```

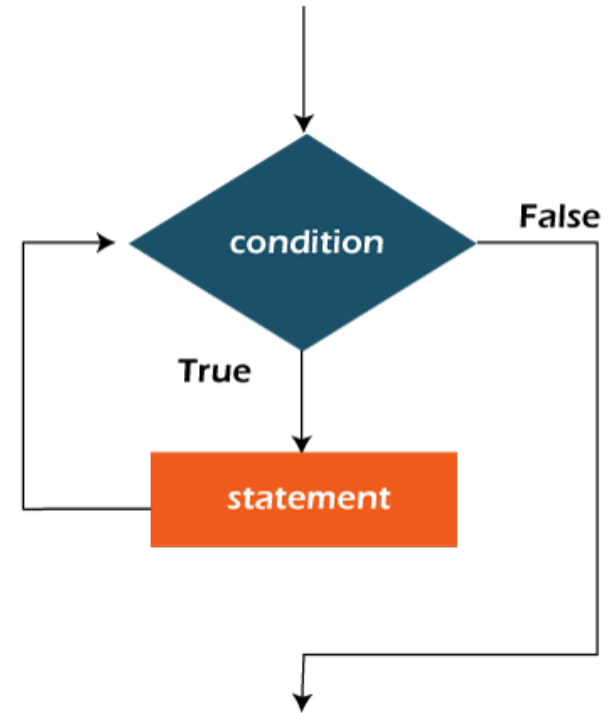
Cấu trúc điều khiển

- **Cấu trúc lặp while:**

```
<bthức_khởi_tạo>;  
while(<bthức_điều_kiện>){  
    <khối_lệnh>;  
    <bthức_lặp>;  
}
```

Ví dụ:

```
int n = 400000, tong = 0;  
int i=2;  
while (i<=n) {  
    tong = tong + i;  
    i+=2;  
}
```



Cấu trúc điều khiển

- **Cấu trúc lặp do .. while:**

<bthức_khởi_tạo>;

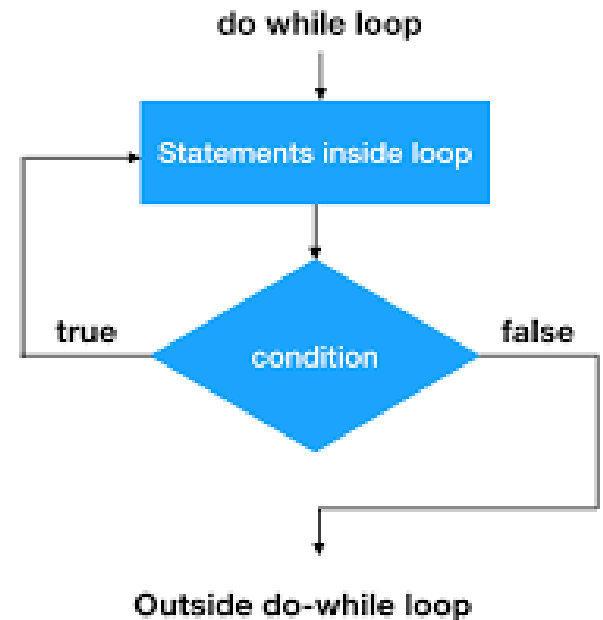
```
do {  
    <khởi_lệnh>;  
    <bthức_lặp>;  
} while(<bthức_điều_kiện>)
```

Ví dụ:

```
int n = 400000, tong = 0;
```

```
int i=2;
```

```
do {  
    tong = tong + i;  
    i+=2;  
} while (i<=n) ;
```

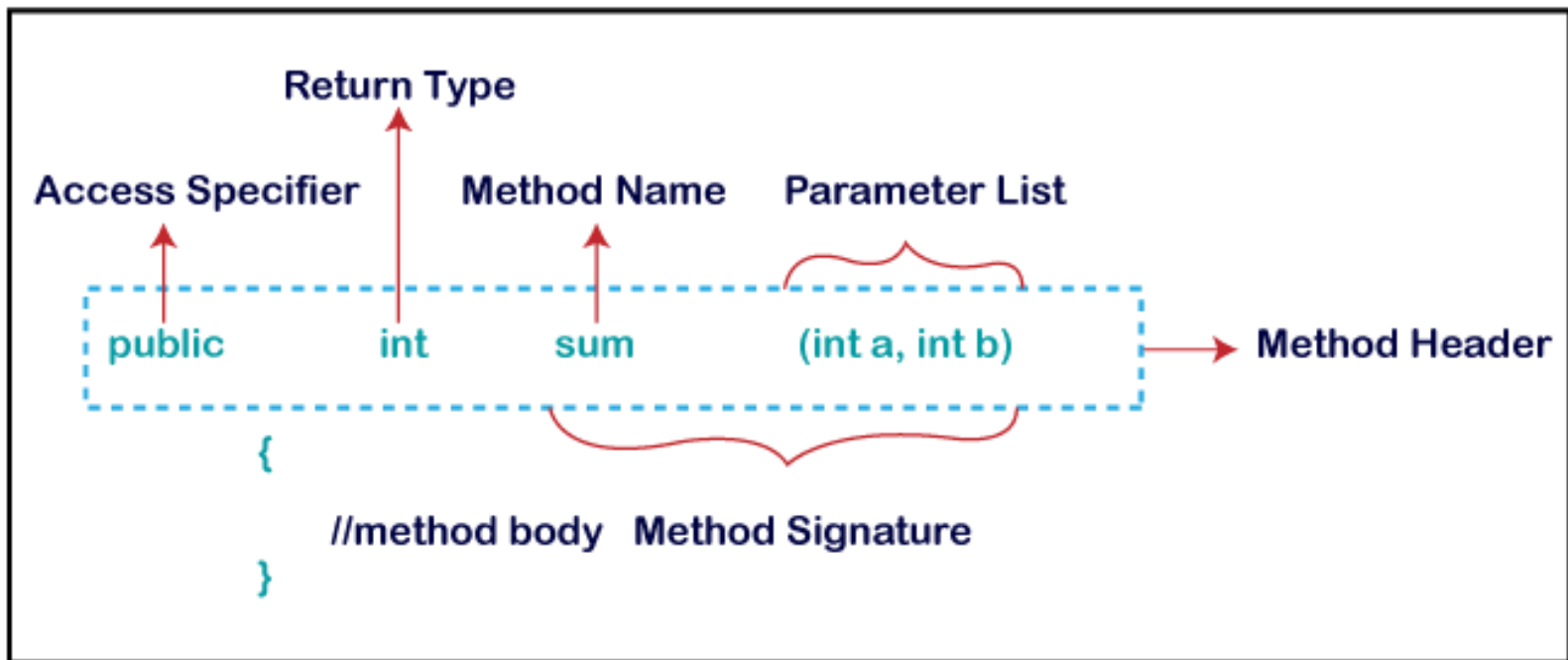


Phương thức (method)

- **Khai báo:**

```
public <kiểu> <tên_phương_thức>(<các_đối_số>){  
    // khối lệnh ...  
    [ return <biểu_thức>; ]  
}
```

Method Declaration



Phương thức (method)

```
public class Tong2So {  
    public      int sum1(int a, int b) { return (a+b); }  
    public static int sum2(int a, int b) { return (a+b); } // hàm tĩnh  
  
    public static void main(String[] args) {  
        int n;  
        // n = sum1(3,3) ; -> Sai  
        Tong2So t = new Tong2So(); // tạo đối tượng t  
        n = t.sum1(3 , 77);  
        n = t.sum2(3 , 77);  
  
        // n = Tong2So.sum1(3 , 77) -> Sai  
        n = Tong2So.sum2(3 , 77); // hàm tĩnh  
    }  
}
```


Tái định nghĩa phương thức

- **Tái định nghĩa** (overloading): cho phép định nghĩa nhiều hàm cùng tên nhưng khác tham số trong lớp. Có 3 cách để tái định nghĩa: Đổi kiểu tham số, Đổi thứ tự tham số, Đổi số tham số.

```
class Cat{  
  
    public void Sound(){  
        System.out.println("meow");  
    }  
  
    //overloading method  
    public void Sound(int num){  
        for(int i=0; i<2;i++){  
            System.out.println("meow");  
        }  
    }  
}
```

OVERLOADING

**Same method
name but
different
parameters**

Tái định nghĩa phương thức

Đổi kiểu tham số

```
public class Tong2So {  
    public int    sum( int a,      int b ) { return (a+b); } // (1)  
    public double sum( double a,  int b ) { return (a+b); } // (2)  
  
    public static void main(String[] args) {  
        Tong2So t = new Tong2So(); // tạo đối tượng t  
        int      n = t.sum(3      , 77); // (1)  
        double   d = t.sum(3.5    , 77); // (2)  
  
    }  
}
```

Tái định nghĩa phương thức

Đổi số tham số

```
public class Tong2So {  
    public int sum(int a, int b )      { return (a+b); }    // (1)  
    public int sum(int a, int b, int c ) { return (a+b+c); } // (2)  
  
    public static void main(String[] args) {  
        Tong2So t = new Tong2So(); // tạo đối tượng t  
        int n = t.sum(3 , 77);    // (1)  
        int d = t.sum(3 , 77, 5); // (2)  
  
    }  
}
```

Tái định nghĩa phương thức

Đổi thứ tự tham số

```
public class Tong2So {  
    public double sum( int a, double b ) { return (a+b); } // (1)  
    public double sum( double a, int b ) { return (a+b); } // (2)  
  
    public static void main(String[] args) {  
        Tong2So t = new Tong2So(); // tạo đối tượng t  
        double n = t.sum(3 , 77.7); // (1)  
        double d = t.sum(3.3 , 77 ); // (2)  
  
        //double k = t.sum(3 , 77 ); //=> lỗi mơ hồ: ambiguous  
        //double m = t.sum(3.0 , 77.0); //=> lỗi sai đối số  
    }  
}
```

Tái định nghĩa phương thức

Đổi thứ tự tham số

```
public class Tong2So {  
    public double sum( int a, double b ) { return (a+b); } // (1)  
    public double sum( double a, int b ) { return (a+b); } // (2)  
    public int sum( int a, int b ) { return (a+b); } // (3)  
    public double sum(double a, double b ) { return (a+b); } // (4)  
  
    public static void main(String[] args) {  
        Tong2So t = new Tong2So(); // tạo đối tượng t  
        double n = t.sum(3 , 77.7); // (1)  
        double d = t.sum(3.3 , 77 ); // (2)  
        int k = t.sum(3 , 77 ); // (3)  
        double m = t.sum(3.0 , 77.0); // (4)  
    }  
}
```