

# Tư Duy Lập Trình

<https://cybersoft.edu.vn>

Định hướng nghề nghiệp

01

Lập trình là gì

02

Cốt lõi trong lập trình

03

Kiến trúc chương trình

04

Biến, kiểu dữ liệu, hằng số, bộ nhớ, lưu trữ, chú thích, cú pháp

05

Tóm tắt bài toán và nguyên lý 3 khối

06

Lệnh điều kiện If-Else

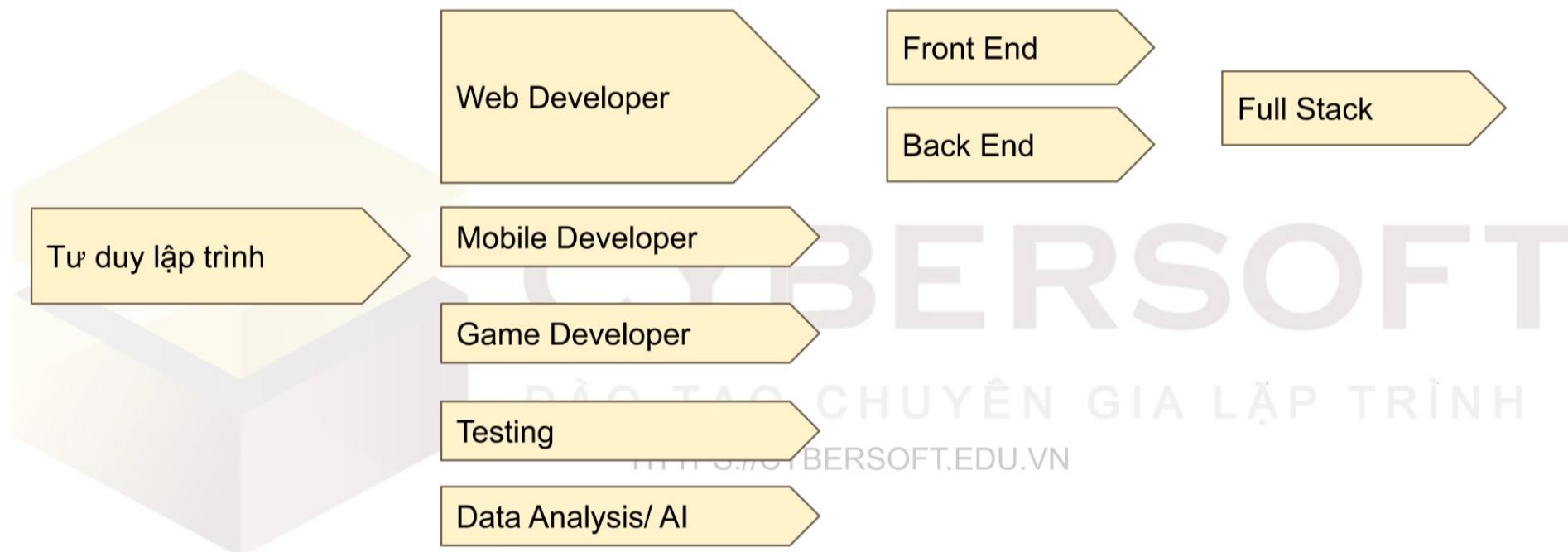
07

Hàm

08

## NỀN TẢNG CƠ BẢN - NỘI DUNG HỌC

# Định hướng nghề nghiệp



# Giới thiệu về Eclipse và Java

- Eclipse là 1 IDE (Integrated Development Environment) để lập trình Java, ngoài ra còn có một số IDE khác như Netbean, IntelliJ, ...
- Để lập trình Java, cần cài đặt bộ JDK (Java Development Kit), phiên bản chúng ta dùng là bản 8 (8u xxx).
- Trong bộ JDK các bạn cài đã tích hợp:
  - JVM (Java Virtual machine) là thành phần nền tảng Java thực thi các chương trình của bạn.
  - JRE (Java runtime environment) khởi tạo JVM và đảm bảo các phụ thuộc có sẵn cho các chương trình của bạn.
  - JDK cho phép bạn tạo các chương trình Java có thể được thực thi và chạy bởi JVM và JRE.

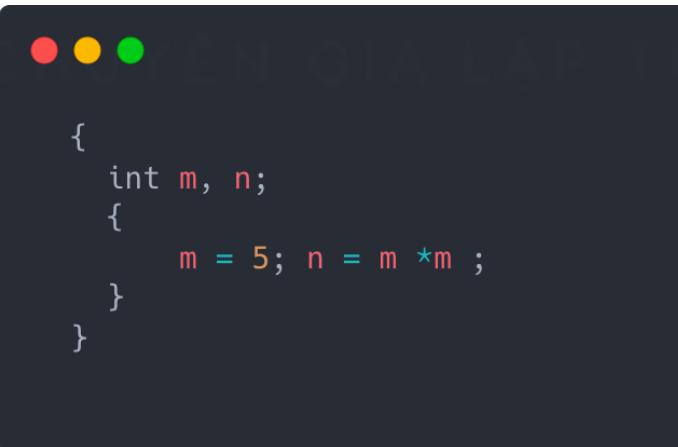
# Lệnh và Khối lệnh

- Khối lệnh (Scope) nằm trong { }, ví dụ:



```
{  
    int m, n;  
    m = 5;  
    n = m *m ;  
}
```

- Có thể lồng khối lệnh vào nhau, ví dụ:

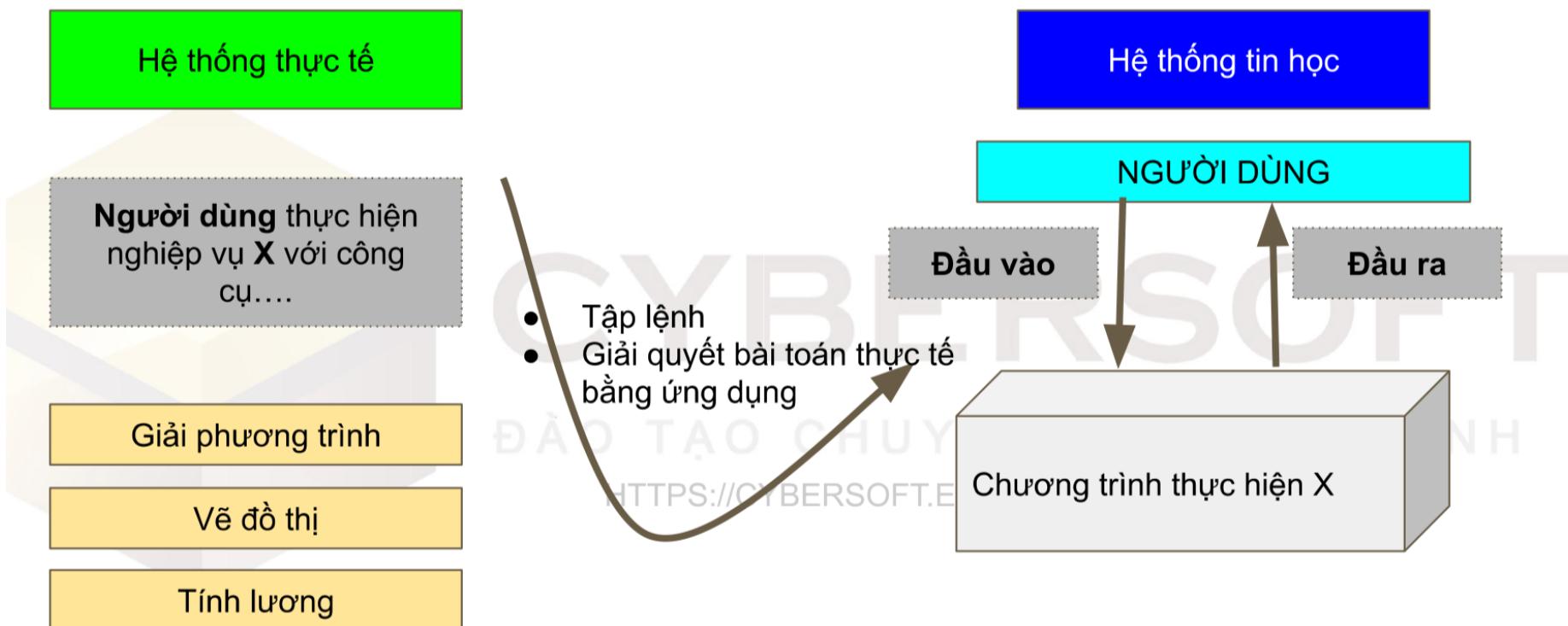


```
{  
    int m, n;  
    {  
        m = 5; n = m *m ;  
    }  
}
```

# Các chủ đề thảo luận

- 01 Thảo luận về phép Gán
- 02 Thảo luận về Phép gán & Toán tử logic
- 03 Thảo luận bài tập Xuất thông báo
- 04 Thảo luận bài tập Có tính toán
- 05 Thảo luận bài tập điều kiện
- 06 Thảo luận bài tập tại lớp

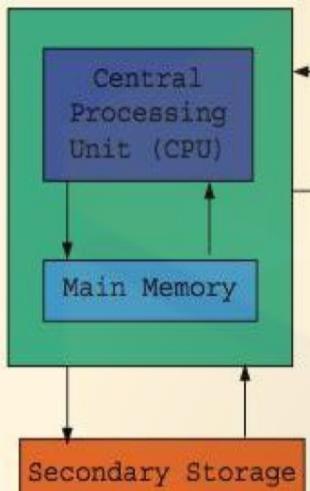
# Lập trình là gì



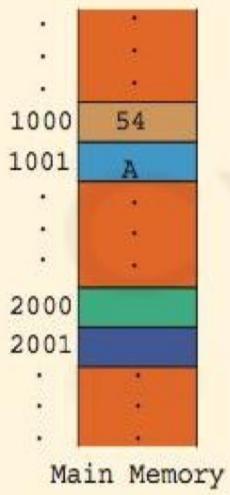
# Cốt lõi trong lập trình

1. Viết code chuẩn hóa ( Tên biến, tên hàm, hằng số, số dòng)
2. Phân tích kỹ trước khi làm
3. Chia nhỏ vấn đề (Divide and Conquer)
4. Kiên trì, code liên tục
5. Làm đi làm lại nhiều lần
6. Bắt đầu từ những gì là đơn giản nhất
7. Mô hình hóa bài toán thực tế vào lập trình
8. Đầu vào, đầu ra, cách thức xử lý
9. Tìm keyword để tìm kiếm
10. Tích cực sử dụng debug, gỡ rối chương trình
11. Kỹ năng test chức năng sau khi làm
12. Suy nghĩ Tái sử dụng (Chia nhỏ, tách hàm, tách lớp)

# Tổng quan lập trình



(a)



(b)

- CPU : Bộ xử lý trung tâm - Bộ xử lý chính
- Main Memory : Bộ nhớ chính (RAM)
- Storage : Lưu trữ
- Input : đầu vào
- Output: đầu ra
- Device : thiết bị

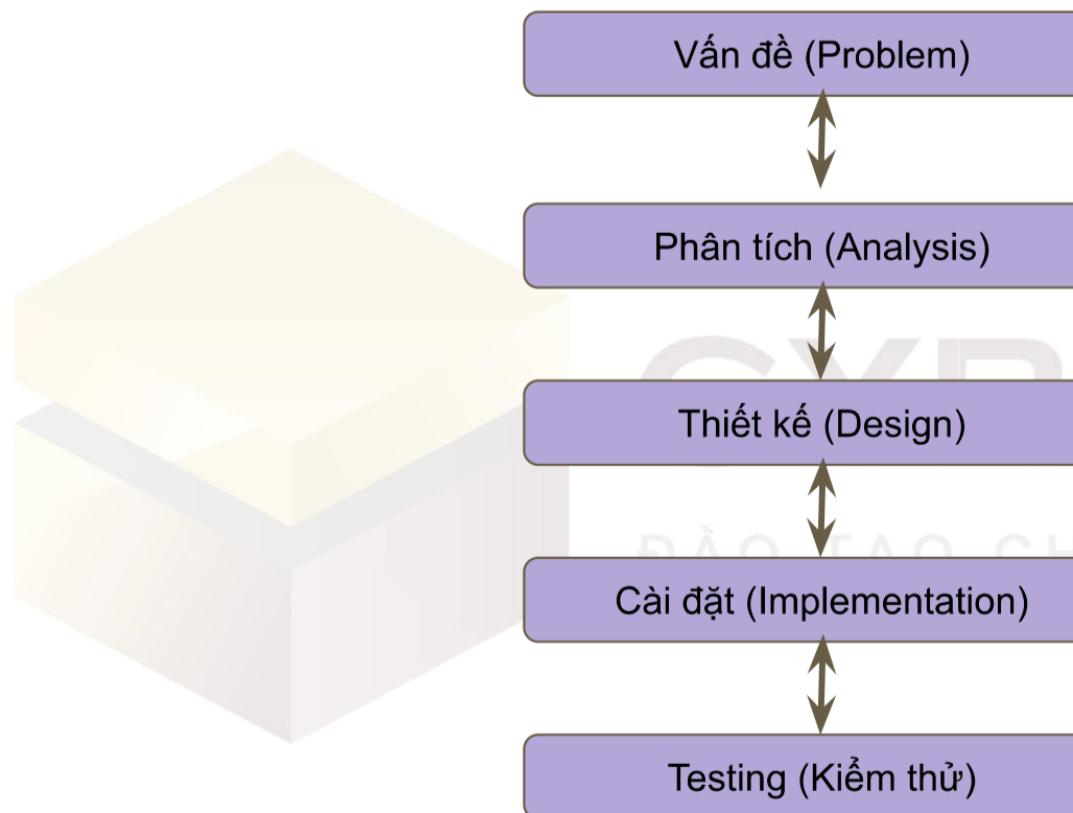
# Tổng quan lập trình

Unit	Symbol	Bits/Bytes
Byte		8 bits
Kilobyte	KB	$2^{10}$ bytes = 1024 bytes
Megabyte	MB	1024 KB = $2^{10}$ KB = $2^{20}$ bytes = 1,048,576 bytes
Gigabyte	GB	1024 MB = $2^{10}$ MB = $2^{30}$ bytes = 1,073,741,824 bytes
Terabyte	TB	1024 GB = $2^{10}$ GB = $2^{40}$ bytes = 1,099,511,627,776 bytes
Petabyte	PB	1024 TB = $2^{10}$ TB = $2^{50}$ bytes = 1,125,899,906,842,624 bytes
Exabyte	EB	1024 PB = $2^{10}$ PB = $2^{60}$ bytes = 1,152,921,504,606,846,976 bytes
Zettabyte	ZB	1024 EB = $2^{10}$ EB = $2^{70}$ bytes = 1,180,591,620,717,411,303,424 bytes

## Bảng đơn vị tính

- Bit : 0 hoặc 1

# Tổng quan lập trình



# Tổng quan lập trình

```
12 //Hàm chính của chương trình
13 public class Main {
14
15     /**
16      * @param args
17      */
18     public static void main(String[] args) {
19         /* Đầu vào */
20
21         String hoTen = "";
22         Scanner scan = new Scanner(System.in);
23         hoTen = scan.nextLine();
24
25         /* End đầu vào */
26
27         /* Xử lý */
28
29         /* End xử lý */
30
31         /* Đầu ra */
32
33         System.out.println("Hello " + hoTen);
34         /* End đầu ra */
35     }
36 }
37 }
```

```
18     public static void main(String[] args) {
19         /* Đầu vào */
20
21         String hoTen = "";
22         Scanner scan = new Scanner(System.in);
23         hoTen = scan.nextLine();
24
25         /* End đầu vào */
26
27         /* Xử lý */
28         int tong = TinhTong(4,5);
29         int hieu = TinhHieu(6, 3);
30         /* End xử lý */
31
32         /* Đầu ra */
33
34         System.out.println("Hello " + hoTen);
35         /* End đầu ra */
36     }
37
38
39     private static int TinhHieu(int i, int j) {
40         return i - j;
41     }
42
43     // Hàm tính tổng
44     static int TinhTong(int a, int b) {
45         return a+b;
46     }
47 }
```

Các loại lập trình:

1. Lập trình dòng lệnh

2. Lập trình hàm

3. Lập trình hướng đối tượng -  
Lớp tách riêng

# Kiến trúc trong file của một chương trình

1. Chú thích chương trình, người tạo, ngày tạo
2. Thư viện
3. Các hằng số
4. Các biến số
5. Hàm chính (Main)
6. Các hàm con

CYBERSOFT  
ĐÀO TẠO CHUYÊN GIA LẬP TRÌNH

```

1 /**
2  * Chương trình : In thông báo
3  * Người tạo : CyberSoft
4  * Ngày tạo : 19/11/2018
5 */
6 package HelloW; // Tên package chứa các lớp
7
8 //Danh sách các thư viện
9 import java.util.Scanner;
10 /**
11  * @author songle
12  *
13 */
14 //Hàm chính của chương trình
15 public class Main {
16
17 /**
18  * @param args
19  */
20 public static void main(String[] args) {
21     /* Đầu vào */
22
23     String hoTen = "";
24     Scanner scan = new Scanner(System.in);
25     hoTen = scan.nextLine();
26
27     /* End đầu vào */
28
29     /* Xử lý */
30
31     /* End xử lý */
32
33     /* Đầu ra */
34
35     System.out.println("Hello " + hoTen);
36     /* End đầu ra */
37 }
38
39 }
40

```

```

1 /**
2  * Chương trình : In thông báo
3  * Người tạo : CyberSoft
4  * Ngày tạo : 19/11/2018
5 */
6 //Danh sách các thư viện
7 using System;
8 namespace HelloW { // Tương tự package
9     public class Program
10    {
11        //Hàm chính của chương trình
12        public static void Main()
13        {
14            /* Đầu vào */
15
16            string hoTen = "";
17            hoTen = Console.ReadLine();
18            /* End đầu vào */
19
20            /* Xử lý */
21
22            /* End xử lý */
23
24            /* Đầu ra */
25
26            Console.WriteLine("Hello " + hoTen);
27            /* End đầu ra */
28        }
29    }
30}
31
32

```

```

1 /**
2  * Chương trình : In thông báo
3  * Người tạo : CyberSoft
4  * Ngày tạo : 19/11/2018
5 */
6 //Danh sách các thư viện
7 #include <iostream>
8
9 using namespace std; //Thư viện nhập xuất chuẩn
10 //Hàm chính của chương trình
11 int main()
12 {
13     /* Đầu vào */
14
15     string hoTen = "";
16     cin>> hoTen;
17
18     /* End đầu vào */
19
20     /* Xử lý */
21
22     /* End xử lý */
23
24
25     /* Đầu ra */
26
27     cout<<"Hello " + hoTen;
28     /* End đầu ra */
29
30 }
31
32

```

```

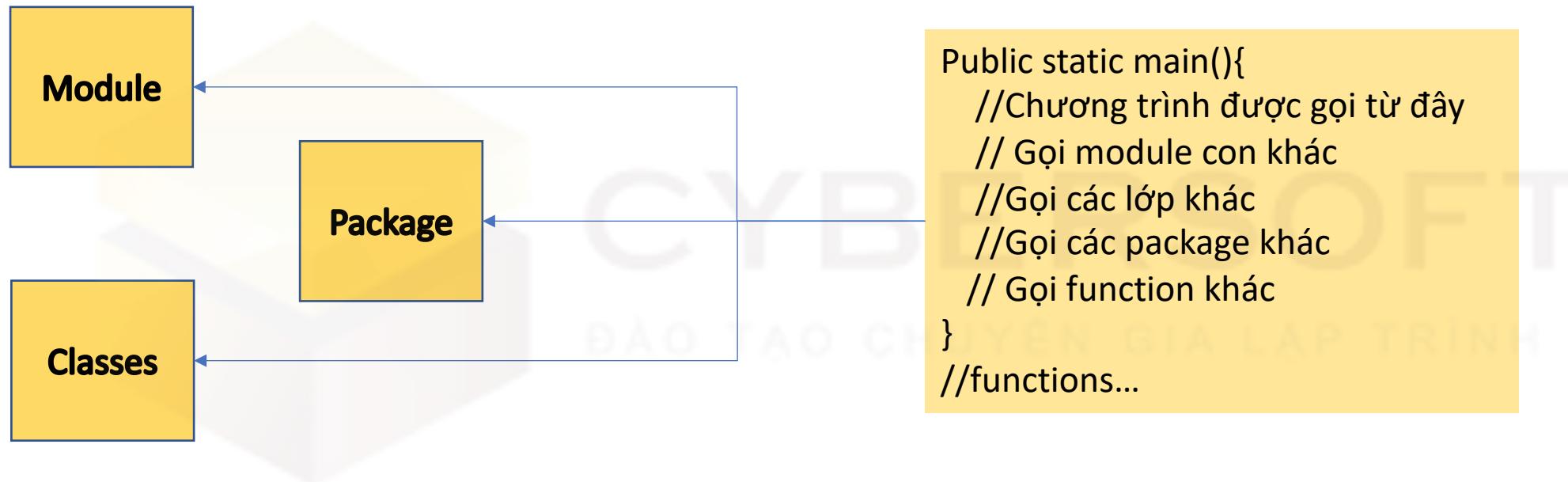
1 /**
2 * Chương trình : In thông báo
3 * Người tạo : CyberSoft
4 * Ngày tạo : 19/11/2018
5 */
6 package HelloW; // Tên package chứa các lớp
7
8 //Danh sách các thư viện
9 import java.util.Scanner;
10 /**
11 * @author songle
12 *
13 */
14 //Hàm chính của chương trình
15 public class Main {
16
17 /**
18 * @param args
19 */
20 public static void main(String[] args) {
21     /* Đầu vào */
22
23     String hoTen = "";
24     Scanner scan = new Scanner(System.in);
25     hoTen = scan.nextLine();
26
27     /* End đầu vào */
28
29     /* Xử lý */
30
31     /* End xử lý */
32
33     /* Đầu ra */
34
35     System.out.println("Hello " + hoTen);
36     /* End đầu ra */
37 }
38 }
39 }
40 }

1 /**
2 * Chương trình : In thông báo
3 * Người tạo : CyberSoft
4 * Ngày tạo : 19/11/2018
5 */
6
7 //Danh sách các thư viện
8 using System;
9 namespace HelloW { // Tương tự package
10     public class Program
11     {
12         //Hàm chính của chương trình
13         public static void Main()
14         {
15             /* Đầu vào */
16
17             string hoTen = "";
18             hoTen = Console.ReadLine();
19             /* End đầu vào */
20
21             /* Xử lý */
22
23             /* End xử lý */
24
25             /* Đầu ra */
26
27             Console.WriteLine("Hello " + hoTen);
28             /* End đầu ra */
29
30         }
31     }
32 }

1 /**
2 * Chương trình : In thông báo
3 * Người tạo : CyberSoft
4 * Ngày tạo : 19/11/2018
5 */
6 //Danh sách các thư viện
7 #include <iostream>
8
9 using namespace std; //Thư viện nhập xuất chuẩn
10 //Hàm chính của chương trình
11 int main()
12 {
13     /* Đầu vào */
14
15     string hoTen = "";
16     cin>> hoTen;
17
18     /* End đầu vào */
19
20     /* Xử lý */
21
22     /* End xử lý */
23
24     /* Đầu ra */
25
26     cout<<"Hello " + hoTen;
27     /* End đầu ra */
28
29
30     return 0;
31 }
32

```

# Chương trình chính (main)



# Syntax, Keyword

- Syntax : Cú pháp chương trình, yêu cầu phải viết đúng
- Keywords : Từ khóa do ngôn ngữ lập trình qui định sẵn, không được đặt tên Biến, Hằng, Hàm, Lớp trùng với tên này.

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

```

1  /**
2   * Chương trình : In thông báo
3   * Người tạo : CyberSoft
4   * Ngày tạo : 19/11/2018
5   */
6 //Danh sách các thư viện
7 package HelloW;
8 import java.util.Scanner;
9
10 //Hàm chính của chương trình
11 public class Main {
12     public static void main(String[] args) {
13         /* Đầu vào */
14         String hoTen = "";
15         Scanner scan = new Scanner(System.in);
16         hoTen = scan.nextLine();
17         /* End đầu vào */
18
19         /* Xử lý */
20         int tong = TinhTong(4, 5);
21         int hieu = TinhHieu(6, 3);
22         /* End xử lý */
23
24         /* Đầu ra */
25         System.out.println("Hello " + hoTen);
26         /* End đầu ra */
27     }
28     // Hàm tính hiệu
29     private static int TinhHieu(int i, int j) {
30         return i - j;
31     }
32     // Hàm tính tổng
33     static int TinhTong(int a, int b) {
34         return a + b;
35     }
36 }

```

## Comment – chú thích

# Lệnh – Khối lệnh

- **Khối lệnh (Scope) nằm trong { }, ví dụ:**

```
{  
    int m, n;  
    m = 5;  
    n = m *m ;  
}
```

- **Có thể lồng khối lệnh vào nhau, ví dụ:**

```
{  
    int m, n;  
    {  
        m = 5;  
        n = m *m ;  
    }  
}
```

CYBERSOFT  
ĐÀO TẠO CHUYÊN GIA LẬP TRÌNH  
[HTTPS://CYBERSOFT.EDU.VN](https://cybersoft.edu.vn)

# Lệnh – Khối lệnh

Từng câu lệnh kết thúc bởi dấu ;

- Int k; int j = 10;
- double d1, d2, d3;
- k=a+b-10;
- boolean p = (a >= b);
- System.out.printin("go");
- squareRoofTwo = Math.sqrt(2);

Mỗi câu lệnh nên xuống dòng

Nên đặt cùng 1 cột dọc nếu trong cùng 1 khối lệnh {}

# Lệnh xuất – print() & println()

Mỗi ngôn ngữ hỗ trợ 1 phương thức để xuất 1 dữ liệu ra màn hình.

Trong Java, chúng ta dùng câu lệnh sau:

- **System.out.print**: In dữ liệu ra màn hình và không xuống dòng
- **System.out.println**: In dữ liệu ra màn hình và xuống dòng (Lần in kế tiếp sẽ ở dòng dưới)

Ví dụ 1:

```
System.out.print(2020);
System.out.println(2020);
```

Ví dụ 2:

```
System.out.println(2019);
System.out.println("Chương trình đầu tiên");
System.out.println('t');
System.out.println(3.14f);
```

# Kiểu dữ liệu

KIỂU DỮ LIỆU		
byte (ví dụ: 1,32, -8,...)	1 byte	-128 -> 127
char (ví dụ: 'c' , 'a',...)	2 byte	Các ký tự trong bảng mã ASCII
int (ví dụ: 1, 112, -5,...)	4 bytes	-2,147,483,648 đến 2,147,483,647
long (ví dụ: 1, 2, -42, ...)	8 bytes	-9,223,372,036,854,775,808 đến 9,223,372,036,854,775,807
float (2.1, 2.3, -3.4, ..)	4 bytes	kiểu chấm động
double ( 2.111, 4.3 ...)	8 bytes	kiểu chấm động
boolean ( kiểu logic)	1 bit	true/ false

## Kiểu String (Chuỗi ký tự)

**String** là 1 kiểu dữ liệu đặc biệt, được xây dựng bởi sự liên kết giữa các **char** với nhau.

## Phân biệt kiểu dữ liệu

Trong các ngôn ngữ lập trình, có 2 dạng kiểu dữ liệu:

- Kiểu dữ liệu nguyên thủy (Primitive Type)
- Kiểu dữ liệu tham chiếu (Reference Type)

## Ký tự in đặc biệt

\-Tab → **thực hiện in khoảng tab**

\n – Newline → **xuống dòng mới**

\" – double quote → **in dấu "**

\' – single quote → **in dấu '**

\ – backslash → **in dấu \**

# Bài 1: Xuất thông báo - Mẫu

Viết chương trình đơn giản với yêu cầu xuất thông báo có nội dung như sau:

**Xin chao**

**Day la chuong trinh dau tien cua toi**



Thông tin cần nhập:

Không có

Thông tin cần xuất

Chuỗi thông báo gồm 2 dòng

Dòng 1: Xin chao

Dòng 2: Day la chuong trinh dau  
tien cua toi

Nghiệp vụ :

Không tính toán gì thêm

23

# Bài 1: Xuất thông báo - Mẫu



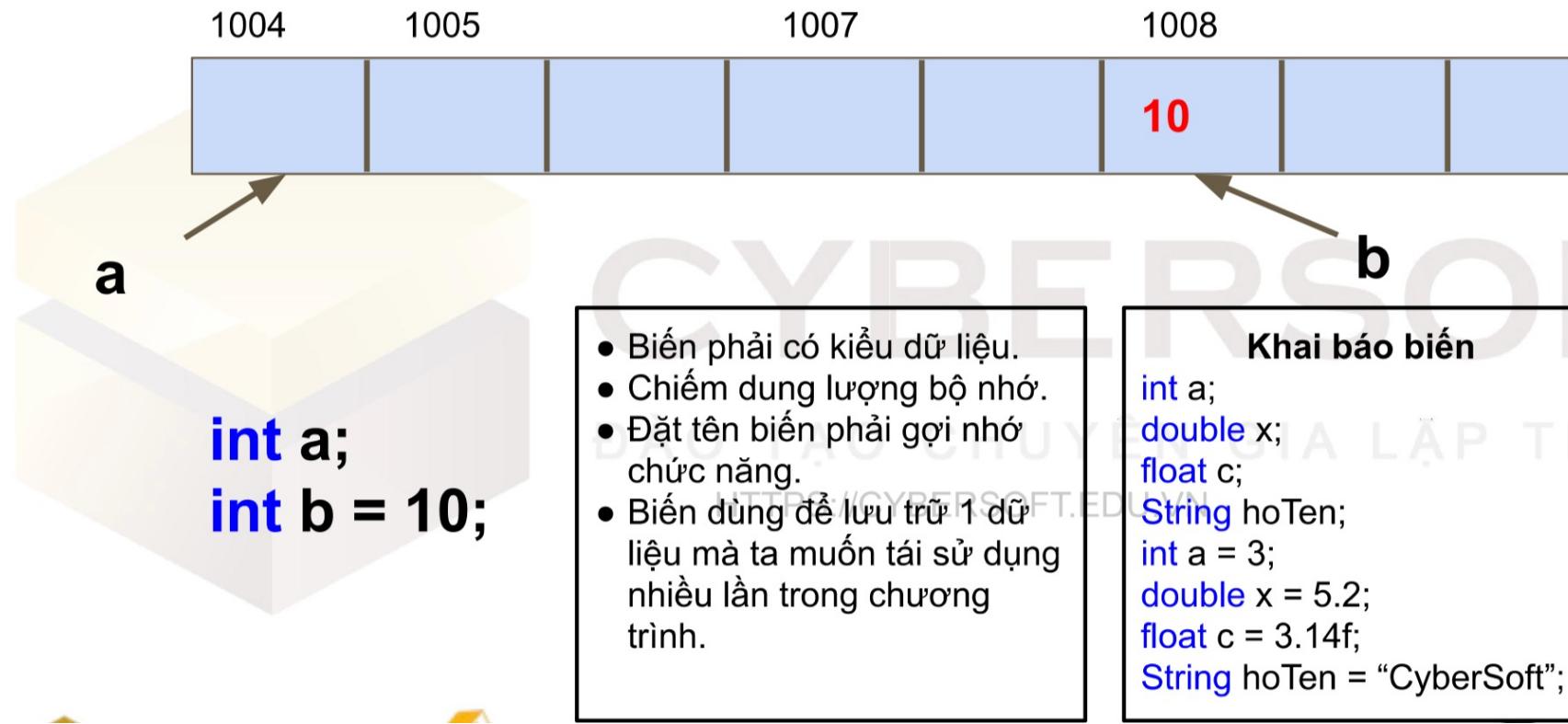
## Thảo luận thêm - Kết luận

1. Có cách 2 không ? Đặt biến và xuất chẵng hạn ?
2. Kết luận :
  - Cấu trúc chương trình .Java
  - Nguyên tắc code 3 khối
  - Chú thích trong Java
  - Lệnh Xuất ra màn hình
  - Lệnh gán
  - Biến (Nếu đặt biến để gán và xuất)

# Biến - Variables

- Lưu trữ dữ liệu tạm thời để xử lý
- Đại diện vùng nhớ được cấp phát
- Phải khai báo trước khi sử dụng
- Phân biệt hoa và thường
- Phải bắt đầu bằng kí tự là chữ, hoặc gạch dưới (\_), hoặc kí tự \$.
- Không được có khoảng trắng giữa các từ ( giá trị - giaTri)
- Không được sử dụng tiếng Việt có dấu

# Giải thích biến & Qui tắc đặt tên



# Qui tắc đặt tên - Naming convention

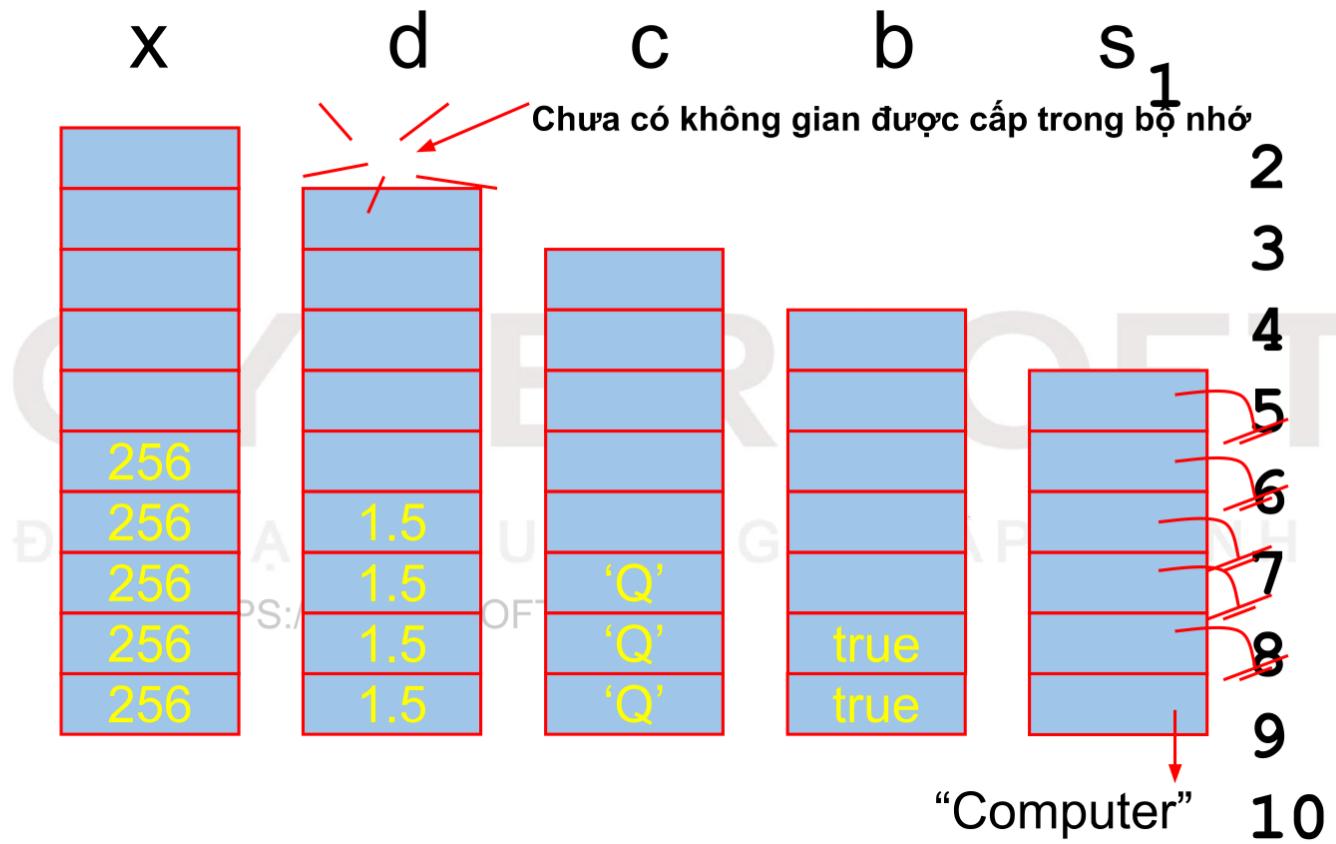
1. Đặt tên biến, tên hàm, tên lớp có ý nghĩa
2. Tên biến viết thường chữ đầu (kiểu Lạc đà - camelCase)  
Ví dụ:
  - int soLuong;
  - float donGia;
3. Xài tiết kiệm -> tăng performance

# Lệnh gán - Assignment

- Toán tử gán “=”
- Gán giá trị bên **PHẢI** cho biến bên **TRÁI**
  - `x = 3;`
  - `Y = 6.5;`
  - `myText = "CyberSoft Academy";`
  - `int z;`
  - `Z = X;`

## Ví dụ về lệnh gán

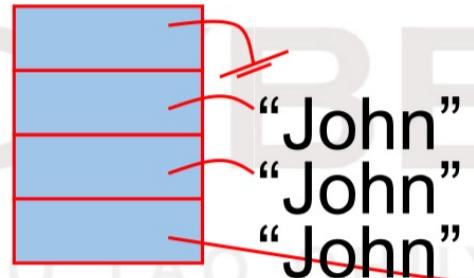
```
int x;  
double d;  
char c;  
boolean b;  
String s;  
x = 256;  
d = 1.5;  
c = 'Q' ;  
b = true;  
s = "Computer"
```



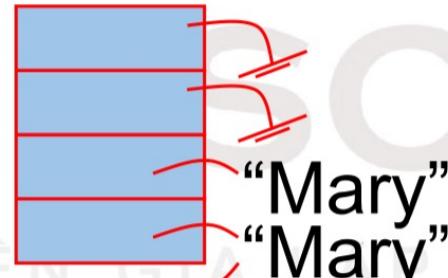
# Phép gán cho Chuỗi

```
String s1,s2;  
s1 = "John";  
s2 = "Mary";  
s1 = s2;
```

s1



s2



HTTPS://CYBERSOFT.EDU.VN

# Tìm lỗi

int i = 9

int j = 1.0;

boolean done = "false";  
Int k = 1;

double k; m = 5e-13;  
char class = "A';

String s = "W';

# Hằng số - Constant

Lý do ?

Cú pháp trong Java :

- **final [kieu du\_lieu] [ten hang so] = [gia\_tri] ;**
- Ví dụ:
  - final double PI = 3.14;
  - final int S0\_NGUYEN = 5;
- Chỉ được gán duy nhất 1 lần, không được gán lần 2.
- Đặt theo thứ tự in hoa và gạch dưới (\_) để cách mỗi từ
- Nên định nghĩa hằng số ở đầu class

# Lệnh nhập

```
Scanner scan = new Scanner(System.in);
scan.nextLine(); // Lấy ra chuỗi nhập
scan.nextInt(); // Lấy ra số nguyên int
scan.nextDouble(); // Lấy ra số thực Double
scan.nextFloat(); // Lấy ra số thực Float
```

```
// Chuyển đổi chuỗi nhập là Số sang Số
String chuoNhap = scan.nextLine();
int so1 = Integer.parseInt(chuoNhap); // Đổi sang kiểu int
float so1 = Float.parseFloat(chuoNhap); // Đổi sang kiểu float
int so1 = Double.parseDouble(chuoNhap); // Đổi sang kiểu float
```

# Coding - Live code

```
// CODE ĐẦU VÀO  
// KHÔNG CÓ  
  
// CODE XỬ LÝ  
// KHÔNG CẦN XỬ LÝ  
  
// CODE XUẤT  
System.out.println("Xin chào");  
System.out.println("Đây là chương trình đầu tiên của tôi");
```

## BT2 - Nhập, Xuất thông báo

Đề: Viết chương trình nhập vào họ và tên của người sử dụng chương trình và sau đó xuất lời chào với nội dung:

Xin chào.. [Tên đã nhập ]

Cảm ơn vì đã sử dụng chương trình này

**Thông tin cần nhập:**

Tên người dùng

**Thông tin cần xuất**

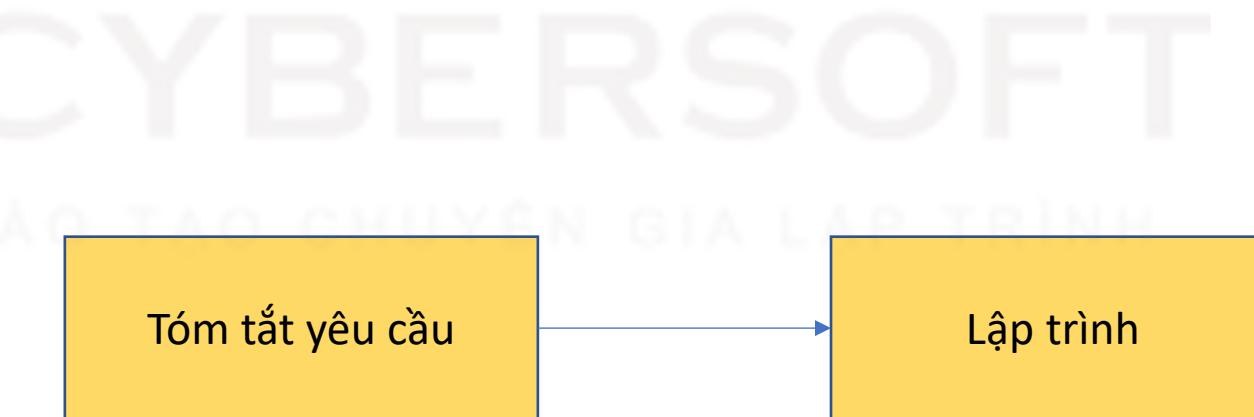
Chuỗi thông báo gồm 2 dòng

Dòng 1: Xin chao + Tên đã nhập

Dòng 2: Cảm ơn vì đã sử dụng  
chương trình này

**Nghệp vụ :**

Xử lý cộng chuỗi nhập



# BT2 - Nhập, Xuất thông báo

Lập trình

Đầu vào

Tên người dùng

1. Lấy chuỗi nhập từ người dùng và gán cho biến **hoTen**
2. Gán chuỗi **toiChao** với thông tin chào + **hoTen**
3. Bổ sung thêm lời chào Dòng 2

Đầu ra

2 Dòng thông báo  
Xin chao + [Họ Tên]  
Xuất dòng 2

# Coding - Live code

```
// Code Đầu vào  
// Khai báo biến  
String hoTen;  
String loiChao;  
Scanner scan: = new Scanner(System.in);  
  
// Code Xử lý  
hoTen = scan.nextLine();  
loiChao = "Xin chào" + hoTen;  
  
// Code Xuất  
System.out.println(loiChao);  
System.out.println("Cảm ơn vì đã sử dụng chương trình này");
```

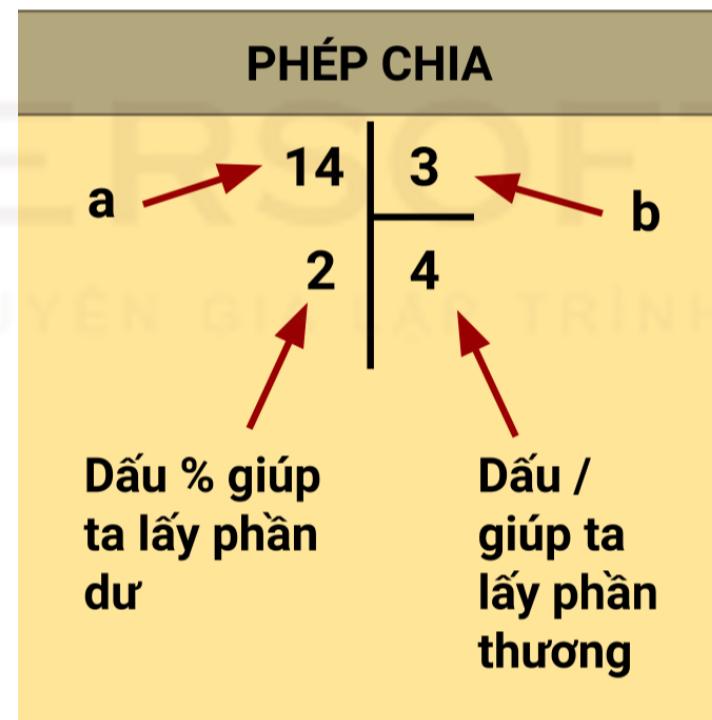
## Thảo luận thêm - Kết luận

1. Có cách 2 không ? Xài một biến ?
2. Kết luận :
  - Cấu trúc chương trình .Java
  - Nguyên tắc code 3 khối
  - Chú thích trong Java
  - Lệnh Xuất ra màn hình
  - Lệnh gán
  - Biến (Nếu đặt biến để gán và xuất)
  - LỆNH NHẬP và lấy dữ liệu người dùng nhập vào

# Toán tử trong lập trình

- Tương tự như toán học, tin học cũng hỗ trợ các phép **cộng, trừ, nhân, chia** giữa 2 số (Cả số nguyên và số thực).
- Cụ thể:

```
int a = 14;  
int b = 3;  
int tong = a + b; => tong = 17  
int hieu = a - b; => hieu = T1  
int tích = a * b; => tích = 42  
int thuong = a / b; => thuong = 4  
int phanDu = a % b; => phanDu = 2
```



## Thảo luận thêm - Kết luận

1. Phép cộng (+) giữa String và số, String và String, String và char ?
2. Phép chia (/) trên số thực?
3. Kết luận :
  - String và số nối + nhau sẽ thành phép nối chuỗi
  - Trong biểu thức chỉ cần 1 biến là kiểu dữ liệu “lớn nhất”, theo thứ tự như sau: int < long < float < double
  - Đồng nhất kiểu dữ liệu?
  - Tránh mất dữ liệu

# BT3 - Xử lý - Mẫu

Đề : Viết chương trình tính tổng 2 số nguyên cho trước

**Thông tin cần nhập:**

2 số nguyên so 1, so 2

**Thông tin cần xuất**

Kq là tổng của so1, so2

**Nghệ vụ :**

Xử lý tính toán 2 số theo công thức tong = so1 + so2



# BT3 - Xử lý - Mẫu

Lập trình

Đầu vào

2 số nguyên so1, so 2

1. so1 = Lấy dữ liệu **so1** từ người dùng
2. so2 = Lấy dữ liệu **so2** từ người dùng
3. **tong = so1 + so2**
4. chuoi = Chuỗi thông báo 2 số **so1, so2** và kết quả **tong**

Đầu ra

Kq là tổng của so1, so2

## Thảo luận thêm - Kết luận

1. Có cách 2 không ?
2. Kết luận :

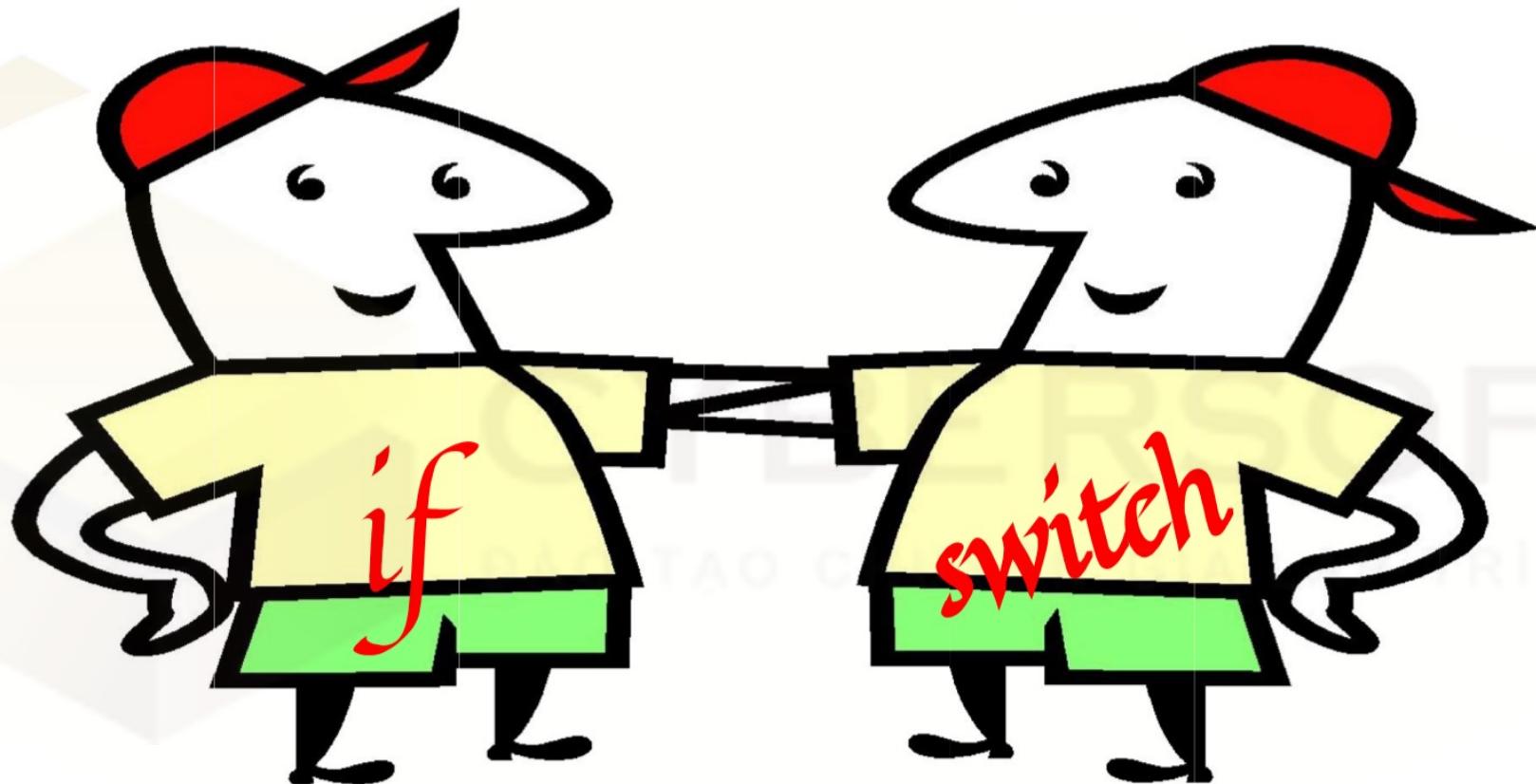
- Lệnh Nhập
- Chuyển đổi dữ liệu từ người dùng nhập vào

# Bài tập tại lớp ( Nhập - xuất)

**Tóm tắt yêu cầu và vẽ mô hình 3 block, lập trình để xử lý các bài toán sau đây:**

1. Viết chương trình tính chiều dài cạnh huyền của một tam giác vuông khi biết 2 cạnh góc vuông.
2. Viết chương trình tính giá trị đơn thức  $P(x) = ax^n$  (a là số thực, n là số nguyên không âm ) với x8 cho trước. (Sử dụng thư viện Math)
3. Viết chương trình tính nhập vào số nguyên dương n với 2 ký số, tính và xuất tổng 2 ký số của n.
4. Tính giá trị trung bình của 5 số được nhập từ bàn phím (Sử dụng hàng số, biến tích lũy, toán tử +=, chú ý phép chia nguyên - ép kiểu )
5. Nhập vào độ C -> đổi sang độ F :  ${}^{\circ}\text{F} = ( {}^{\circ}\text{C} \times 1.8 ) + 32$
6. Cho trước tỉ giá VND-USD là : 23.500đ, viết chương trình nhập vào số USD và tính ra tiền VNĐ.

# Cấu trúc điều kiện



## Kiểu dữ liệu boolean

- boolean chỉ chứa được 2 giá trị là true hoặc false.
- Ví dụ:

```
boolean giaTri_1 = true;  
boolean giaTri_2 = false;
```

- Lý do dùng boolean? Ý nghĩa? Các phép so sánh trong lập trình: >, >=, <, <=, ==, !=

# Toán tử logic giữa 2 giá trị boolean

- A&&B đúng khi tất cả đều đúng
- A| |B đúng khi một trong 2 đúng

A	B	A&&B	A  B	!A
true	true	true	true	false
false	true	false	true	true
true	false	false	true	false
false	false	false	false	true

HTTP://CYBERSOFT.EDU.VN

- a == b : So sánh bằng. Trả về True Nếu a và b cùng giá trị
- a!=b : So sánh khác . Trả về True Nếu a và b khác giá trị

# Cấu trúc điều kiện (TT)

- Nếu
  - Xét trường hợp
- => Sử dụng các cấu trúc:
- If
  - if .. else
  - switch()

CYBERSOFT  
ĐÀO TẠO CHUYÊN GIA LẬP TRÌNH

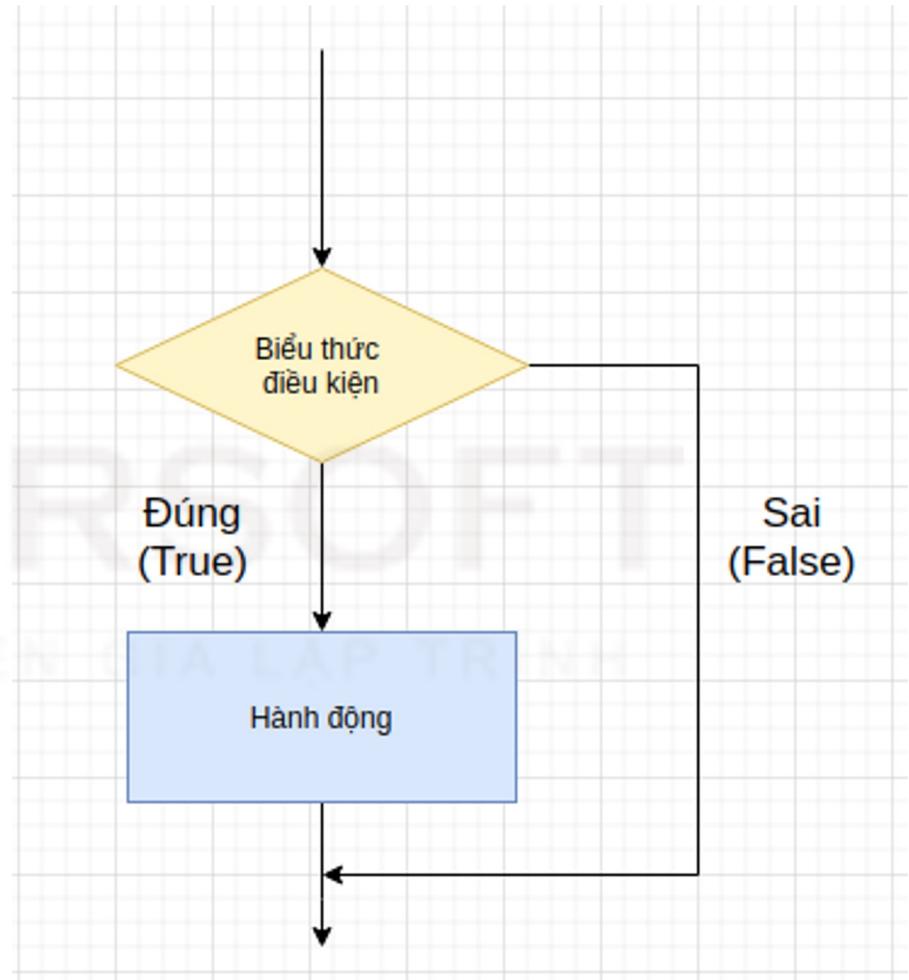
# Lệnh If

Cú pháp

**if** (*Biểu thức điều kiện*)

*Hành động*

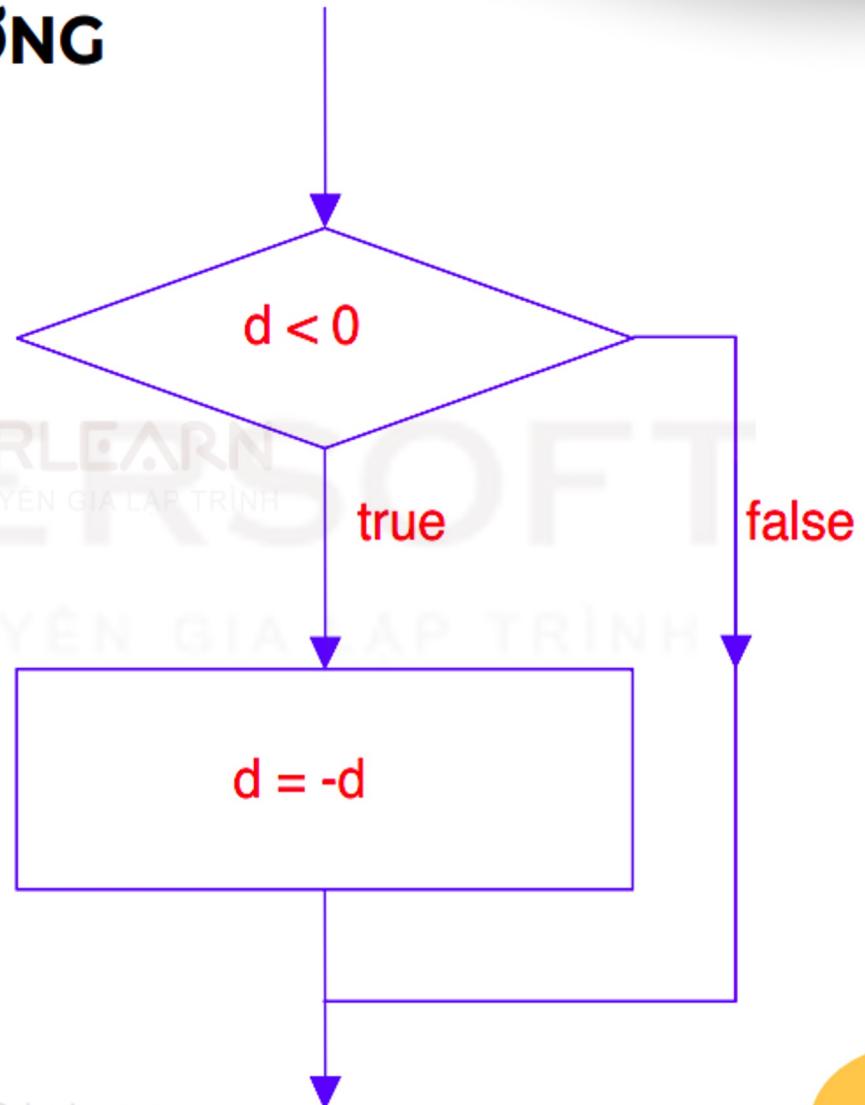
- **Hành động** có thể là một dòng hoặc nhóm các xử lý gồm nhiều dòng lệnh trong cặp {}
- Nếu **biểu thức điều kiện** là đúng (True) thì sẽ thực hiện **hành động**
- Ngược lại, chương trình sẽ không thực hiện **hành động** và sẽ chạy các dòng lệnh tiếp theo ngoài if



## VÍ DỤ 1: CHUYỂN SỐ ÂM SANG DƯƠNG

```
if (d < 0) {  
    d = -d;  
}
```

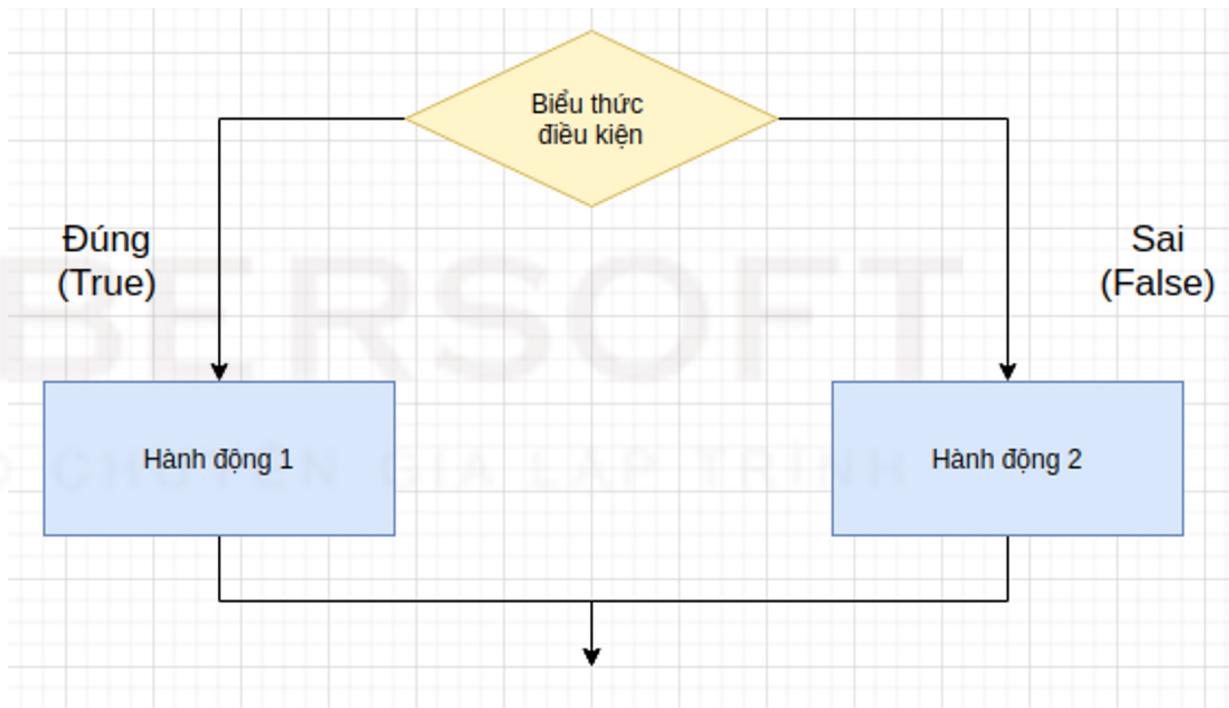
- \* Giả sử  $d = 4$
- \* Giả sử  $d = -9$



# Lệnh if else

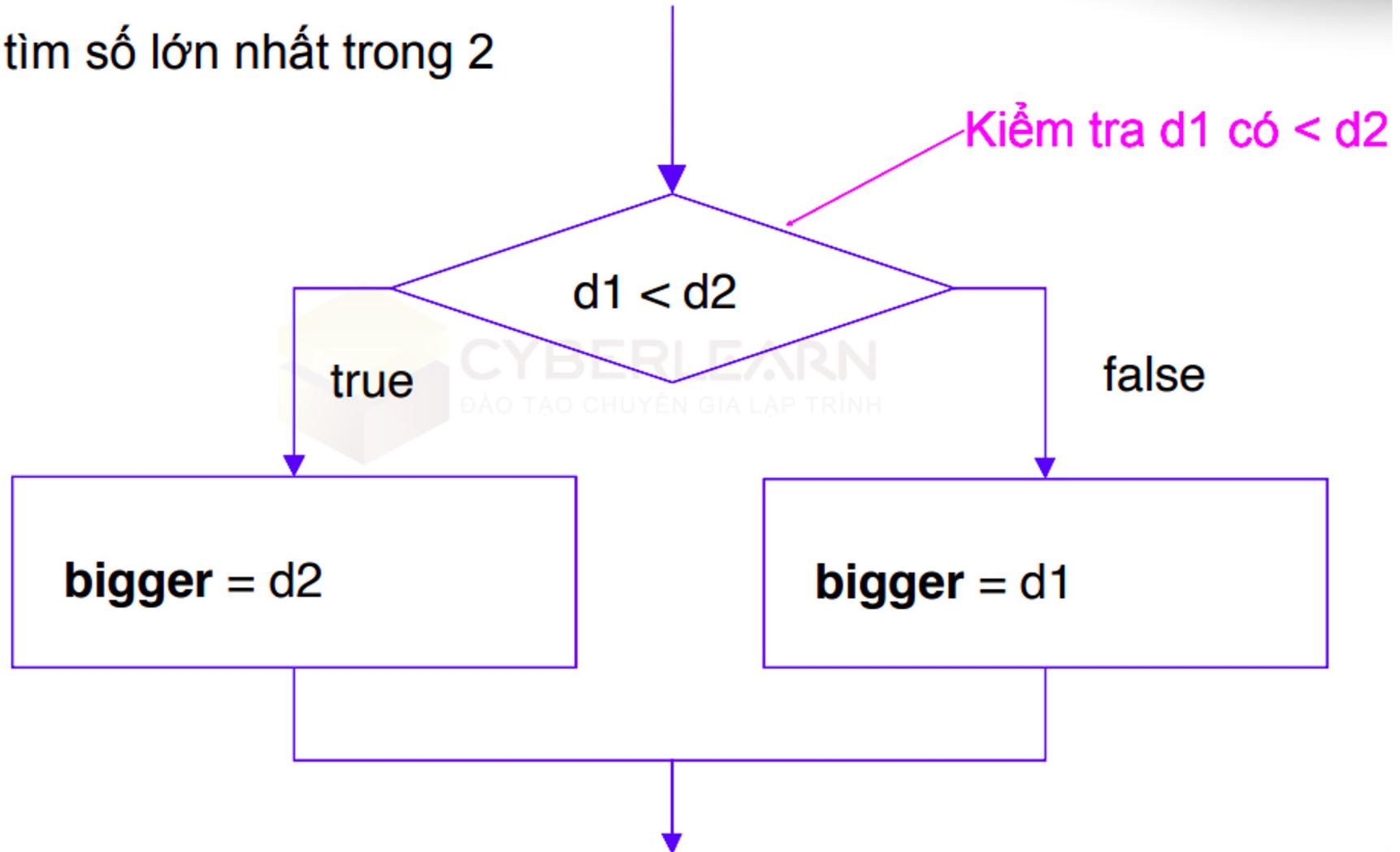
- Cú pháp  
if (*biểu thức điều kiện*)  
    *Hành động 1*  
else  
    *Hành động 2*

- Nếu **biểu thức điều kiện** là đúng (True) thì sẽ thực hiện **hành động 1** ngược lại sẽ thực hiện **hành động 2**
- **Hành động** có thể là một dòng hoặc nhóm các xử lý gồm nhiều dòng lệnh trong cặp {}



# VÍ DỤ 1

Nhập vào 2 số, tìm số lớn nhất trong 2 số d1 và d2



# TOÁN TỬ ? :

- Lệnh *if ...else* thỉnh thoảng được viết sử dụng toán tử **? :** nếu câu lệnh sau biểu thức đơn giản.
- Cú pháp

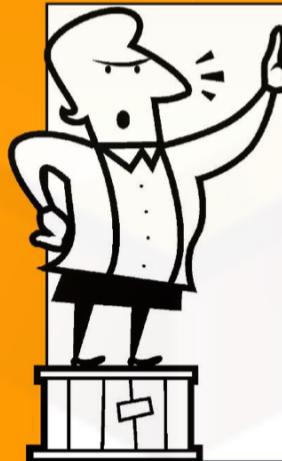
*Biểu thức điều kiện ? Biểu thức1 : Biểu thức2*

- Nếu *Biểu thức điều kiện* là **True**, giá trị trả về là *Biểu thức1*.
- Nếu *Biểu thức điều kiện* là **false**, giá trị trả về là *Biểu thức2*.

# Ví dụ

```
int bigger = (intA > intB) ? intA : intB;
```

tương đương :



```
int bigger;
if(intA > intB){
    bigger = intA;
} else{
    bigger = intB;
}
```

# Toán tử ++, --, += , -=, \*=

- 1) **x++;** giống **x = x + 1;** giống **x+=1;** // tăng thêm 1
- 2) **x--;** giống **x = x -1;** giống **x-=1;** // giảm bớt 1
- 3) **x+= y;** giống **x = x + y;**
- 4) **x\*= y** giống **x = x \* y;**
- 5) **x/=y** giống **x = x/y;**

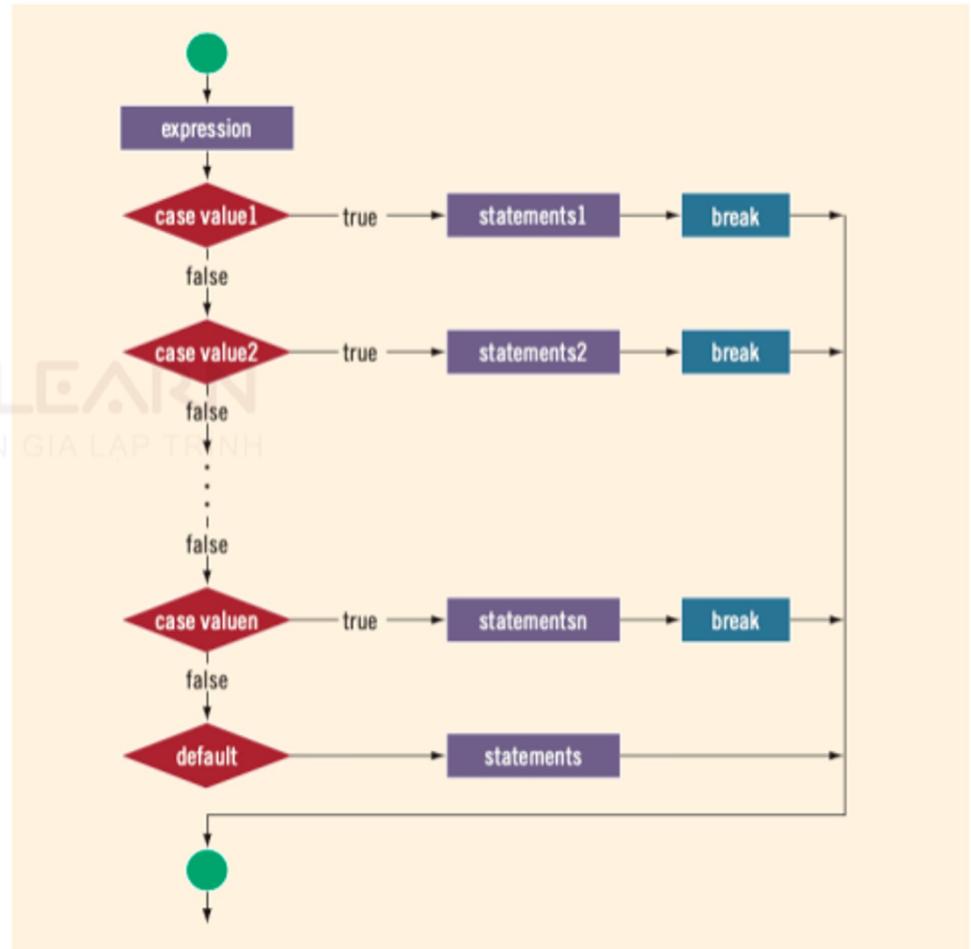
```
a= 5;  
++a;      //a = 6  
a++;      //a = 7  
--a;      //a = 6  
a--;      //a = 5
```

```
int n1 = 1;    n1 = 5  
int n2 = ++n1; n2 = 2  
//int n3 = ++ ++n1; n3 = 4  
int n4 = n1++; n4 = 4
```

# Lệnh switch ... case

Cú pháp

```
switch(Biểu thức switch){  
    case case1 :  
        Action1  
        break;  
    case case2 :  
        Action2  
        break;  
    :  
    :  
    case caseN :  
        ActionN  
        break;  
    default :  
        DefaultAction  
}
```



# Lệnh switch ... case

- Khi Biểu thức switch được xét đến, chương trình sẽ chạy đến kiểm tra các lệnh tương ứng với các giá trị trong CASE.
- Nếu Biểu thức switch không khớp với tất cả các CASE sẽ chạy lệnh trong default.
- Nếu đúng trường hợp CASE, lệnh sẽ chạy CHO ĐẾN KHI GẶP LỆNH break thì mới kết thúc switch
- Gặp lệnh break chương trình sẽ nhảy ngay đến cuối lệnh switch .

# ĐỌC SỐ TƯƠNG ỨNG TỪ NGƯỜI DÙNG

Đọc các số do người dùng nhập vào từ 1 đến 4

```
switch(n){  
    case 1:  
        chuoit = "số 1";  
        break;  
    case 2:  
        chuoit = "số 2";  
        break;  
    case 3:  
        chuoit = "số 3";  
        break;  
    case 4:  
        chuoit = "số 4";  
        break;  
    default:  
        chuoit = "Vui lòng nhập số từ 1 đến 4";  
}
```

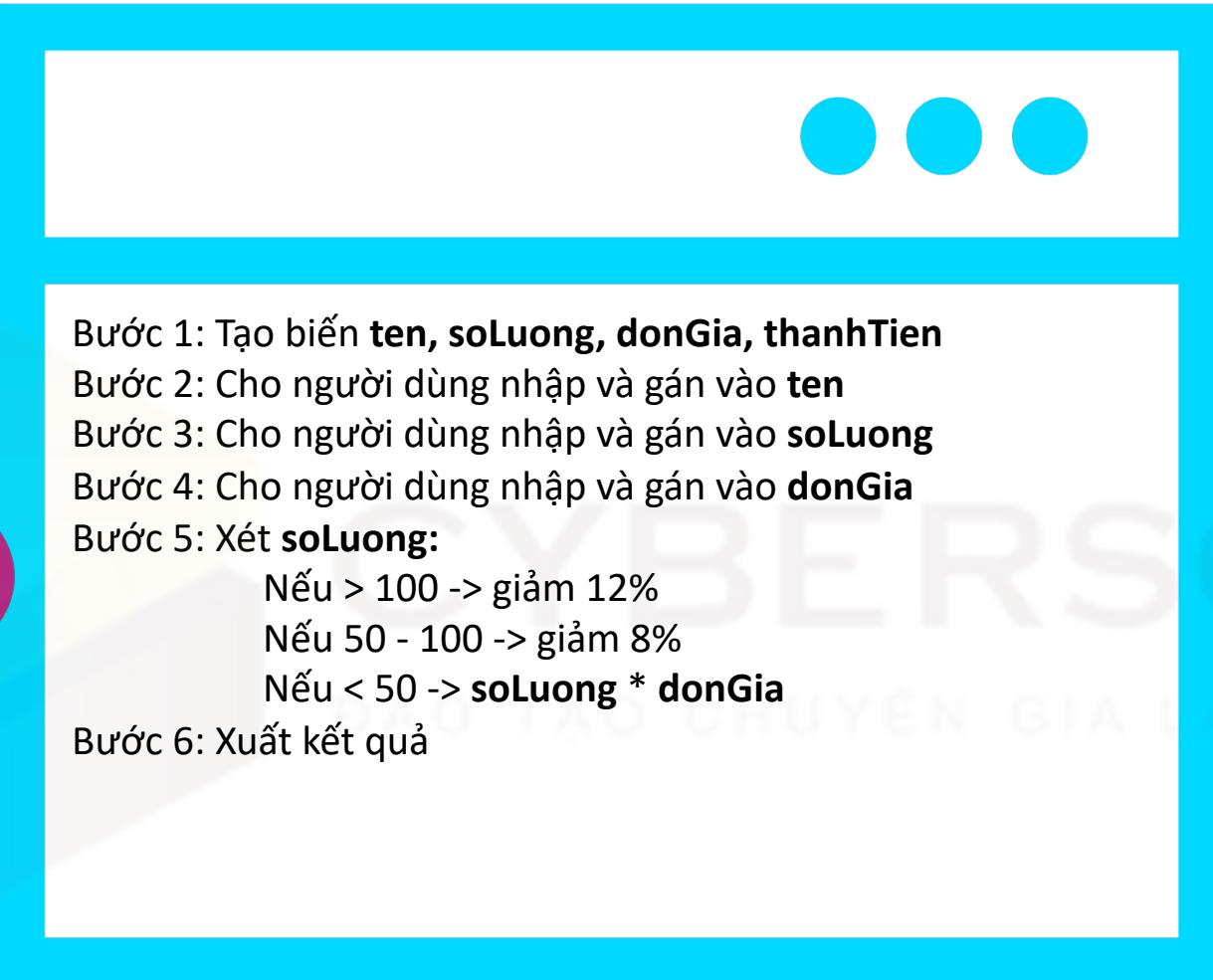
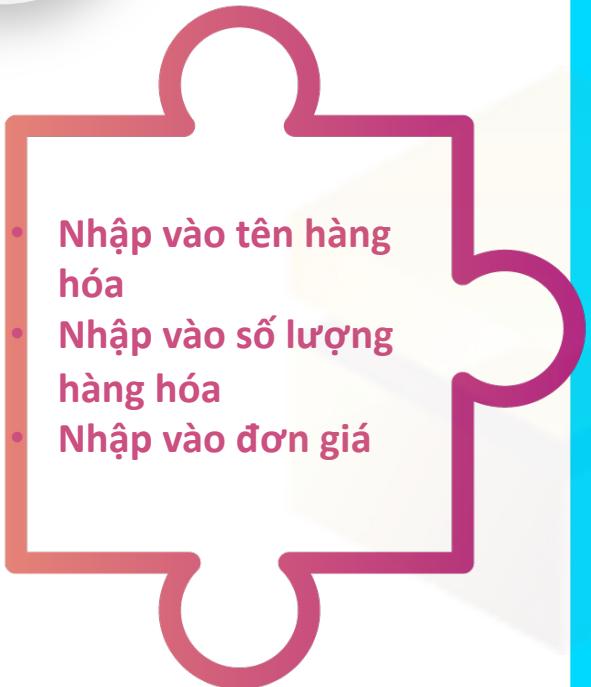
n = 3



# Bài 1: Luyện if ... else

Viết chương trình nhập vào thông tin 1 mặt hàng Tên, số lượng, đơn giá.Tính và xuất tiền phải trả dựa theo quy tắc: Nếu mua với số lượng từ 50 đến 100 sẽ được giảm 8%, mua với số lượng trên 100 thì số lượng từ 100 trở đi sẽ được giảm 12%.





## Bài 2: Xếp loại sinh viên

Viết chương trình nhập vào thông tin 1 sinh viên: Họ tên, Toán, Lý, Hóa

Tính và xuất kết quả xếp loại theo bảng xếp loại sau:

- Loại Giỏi: Điểm TB $\geq$ 8,5
- Loại Khá: 6,5= $\leq$ Điểm TB $<$ 8,5
- Loại TB: 5= $\leq$ Điểm TB $<$ 6,5
- Loại Yếu: Điểm TB $<$ 5

# Bài tập ( Lệnh điều kiện)

**Tóm tắt yêu cầu và vẽ mô hình 3 block, lập trình để xử lý các bài toán sau đây:**

1. Viết chương trình nhập vào 2 số nguyên m, n. Xác định và xuất số nguyên lớn nhất.
2. Cho người dùng nhập vào 3 số nguyên, viết chương trình xuất 3 số theo thứ tự tăng dần.
3. Cũng yêu cầu như câu 2, nhưng cho phép Hỏi người dùng muốn xuất theo thứ tự nào, nếu người dùng chọn 1 thì xuất Tăng dần, người dùng chọn 2 thì xuất giảm dần.
4. Viết chương trình cho phép nhập vào số nguyên có hai chữ số. Hãy in ra cách đọc của số nguyên này.
5. Hãy viết chương trình “Chào Hỏi” các thành viên của gia đình có đặc điểm sau đây: đầu tiên máy sẽ hỏi ai đang sử dụng máy. Sau đó căn cứ vào câu trả lời nhận được mà máy cần đưa ra một câu chào hợp lý, dễ thương. Giả sử trong gia đình có 4 thành viên là : Bố (B), Mẹ (H), anh trai (A) và em gái (E)

# Bài tập ( Lệnh điều kiện)

Tóm tắt yêu cầu và vẽ mô hình 3 block, lập trình để xử lý các bài toán sau đây:

1. Viết chương trình nhập vào 3 số thực. Hãy thay tất cả các số âm bằng trị tuyệt đối của nó.
  - Ví dụ:  $a=2$   $b= -5$   $c= -8$
  - Kết quả :  $a = 2$   $b= 5$   $c=8$
2. Viết chương trình nhập vào số nguyên có ba chữ số. Hãy in ra cách đọc nó.
3. Viết chương trình nhập 3 cạnh của một tam giác. Hãy cho biết tam giác đó là tam giác gì. (Chú ý kiểm tra hợp lệ)  
Ví dụ:
  - $a = 2$   $b=2$   $c=1$  Tam giác cân
  - $a =3$   $b=3$   $c=3$  Tam giác đều
  - $a =3$   $b=4$   $c=5$  Tam giác vuông
4. Viết chương trình nhập tháng, năm. Hãy cho biết tháng đó có bao nhiêu ngày.

# Bài tập ( Lệnh điều kiện)

**Tóm tắt yêu cầu và vẽ mô hình 3 block, lập trình để xử lý các bài toán sau đây:**

1. Nhập tên và năm sinh của 3 sinh viên. Hãy viết chương trình xuất ra tên và năm sinh của sinh viên trẻ nhất.
2. Cho 3 số nguyên. Hãy viết chương trình xuất ra có bao nhiêu số lẻ và bao nhiêu số chẵn.
3. Cho biết tên, năm sinh và giới tính của 3 cán bộ. Hãy viết chương trình tìm và in tên cán bộ Nam trẻ nhất.
4. Cho biết tên và tọa độ nhà của 3 sinh viên. Cho biết tọa độ của trường đại học. Hãy viết chương trình cho máy in tên sinh viên ở xa trường đại học nhất.

# Bài tập: Tính tiền điện

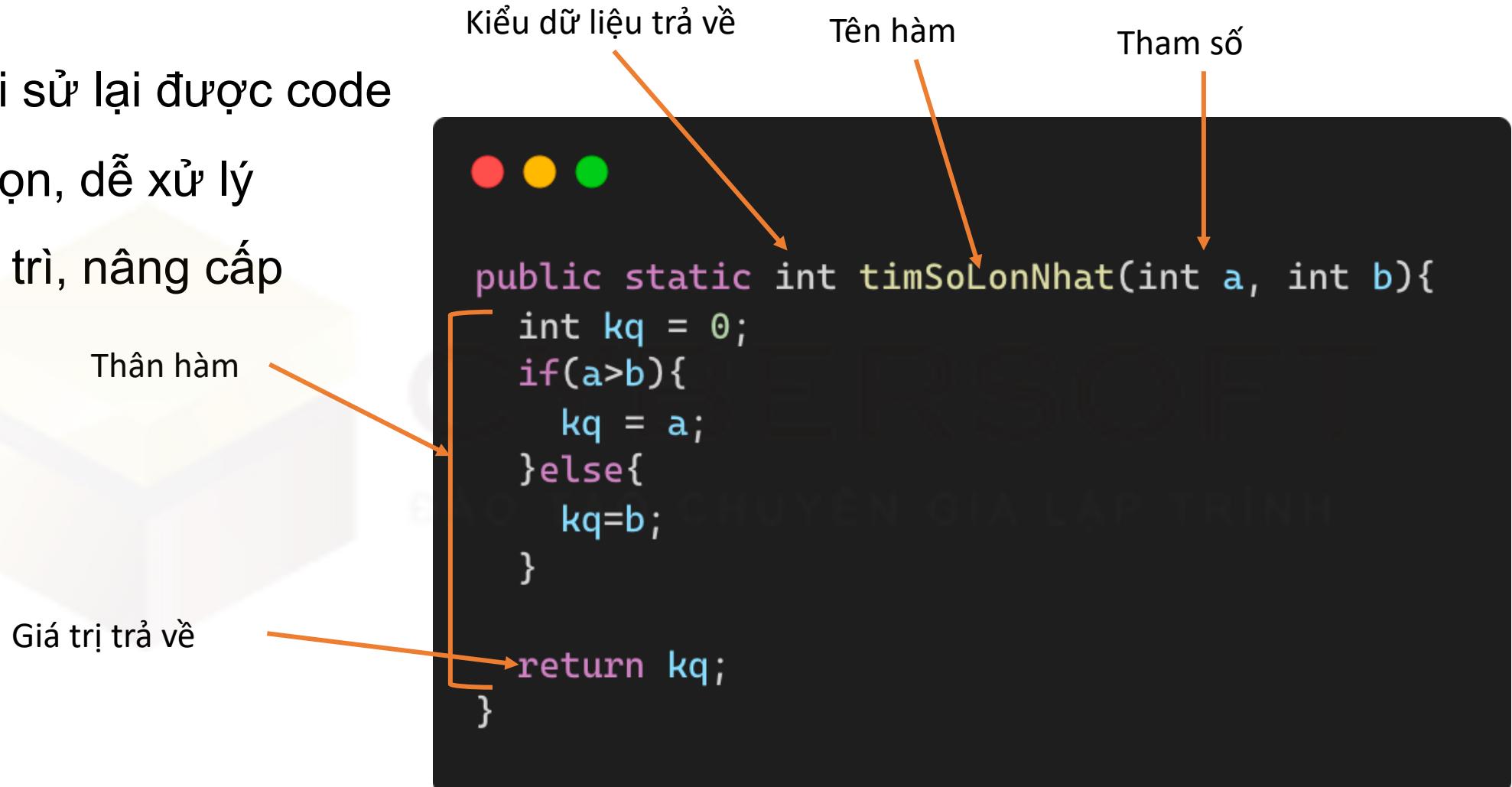
Viết chương trình nhập vào thông tin tiêu thụ điện (Tên, số Kw)

Tính và xuất Tiền trả theo quy tắc:

- 50kw đầu :  $500d/kw \times = 45 (x \leq 50) \rightarrow x * 500$
- 50kw kế:  $650d/kw \times (50-100) \rightarrow 50 * 500 + (x - 50) * 650$
- 100kw Kế :  $850d/kw \times (100-200) \rightarrow 50 * 500 + 50 * 650 + (x - 100) * 850$
- 150kw kế :  $1100d/kw \times (200-350) \rightarrow 50 * 500 + 50 * 650 + 100 * 850 + (x - 200) * 1100$
- Còn lại :  $1300d/kw x > 350 \rightarrow$

# Hàm – phần 1

- Giúp tái sử lại được code
- Code gọn, dễ xử lý
- Dễ bảo trì, nâng cấp



# Hàm(tt)

- Tên Hàm : Gợi nhớ, Động từ
- Tham số truyền vào : Không có, có 1 hoặc nhiều, **TÊN THAM SỐ ĐẶT TÊN GÌ CŨNG ĐƯỢC**
- Giá trị trả về : Có thể có hoặc không, nếu không có thì hàm sẽ trả về **void**

```
● ● ●  
public static void XuatDuLieu (int giaTri) {  
    System.out.println("Giá trị là :" + giaTri);  
}
```

# Cách gọi hàm

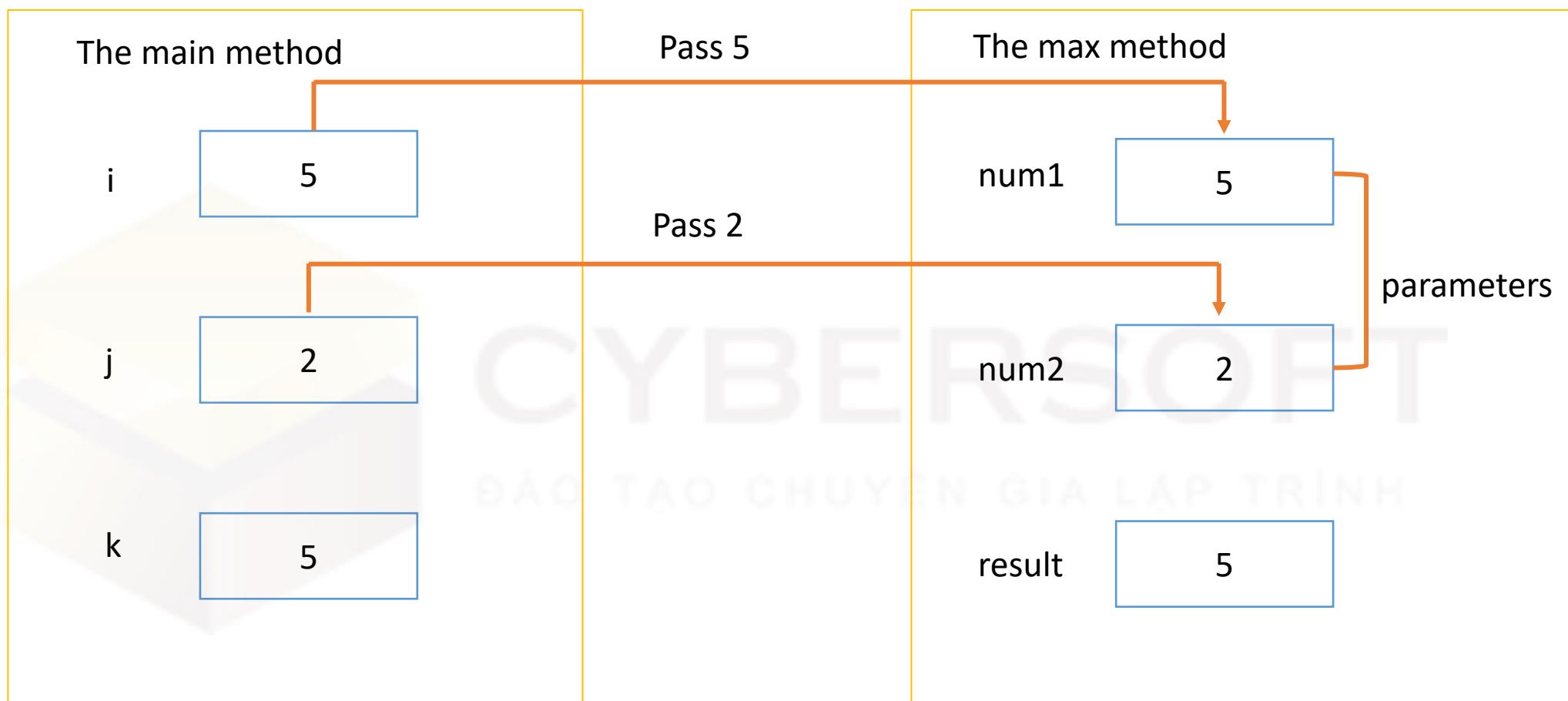


```
public static void main (String[] args) {  
    int i = 5;  
    int j = 2;  
    int k = max(i, j);  
  
    System.out.println("The maximum between " + i + "  
        and " + j + " is "+k);  
}
```



```
public static int max (int num1, int num2) {  
    int result;  
  
    if(num1 > num2)  
        result = num1;  
    else  
        result = num2;  
  
    return result;  
}
```

# Cách gọi hàm



# Bài tập

1. Viết hàm tìm số lớn nhất trong 3 số a,b, c do người dùng nhập vào
2. Viết hàm kiểm tra một số do người dùng nhập vào là số chẵn hay lẻ
3. Viết hàm tính tổng 3 số do người dùng nhập vào
4. Xây dựng chương trình tính toán đơn giản cộng, trừ, nhân chia số nguyên khác không. Yêu cầu SỬ DỤNG HÀM. Xuất thông báo cho người dùng chọn : Nếu 1 là +, 2 là -, 3 là \*, 4 là chia.

# Bài tập : Tính tiền Grab

Cho người dùng chọn 1 trong 3 loại Grab :

1: GrabCar

2 : Grab SUV

3 : GrabBlack

Cho người dùng nhập vào số KM đi được và thời gian chờ

Yêu cầu:

1. Tính tổng tiền đi được
2. In hoá đơn chi tiết như biểu mẫu

# Bài tập : Tính tiền Grab

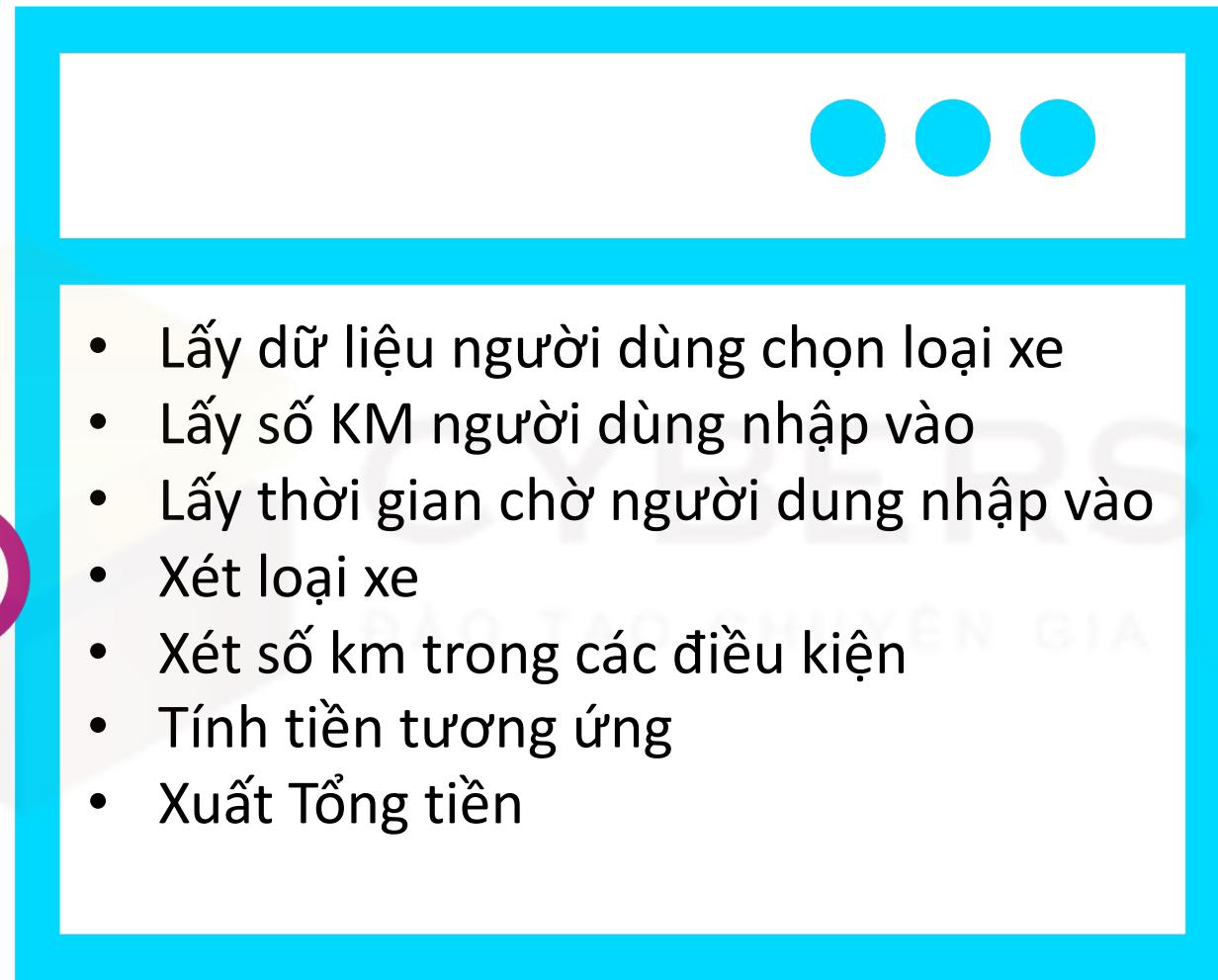
**Chương trình cho phép người dùng nhập vào loại xe, Số KM đi được, Thời gian chờ và xuất ra tổng tiền phải trả theo biểu giá như sau:**

BẢNG GIÁ CƯỚC GRAB			
THEO KM	GRAB CAR (Đ)	GRAB SUV (Đ)	GRAB BLACK (Đ)
KM ĐẦU TIÊN	8000	9000	10000
Từ 1 đến 19	7500	8500	9500
Từ 19 trở lên	7000	8000	9000
Thời gian chờ trên 3 phút (mỗi 3 phút)	2000	3000	3500

# Bài tập : Màn hình chi tiết hóa đơn

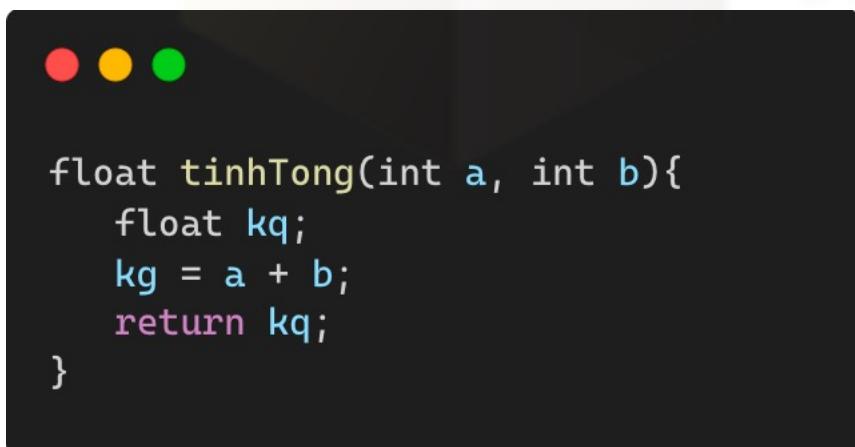
CHI TIẾT HÓA ĐƠN			
CHI TIẾT	SỬ DỤNG(KM)	ĐƠN GIÁ( 1000đ)	THÀNH TIỀN(1000đ)
KM ĐẦU TIÊN	0.8	(tùy theo loại grab)	
Từ .... đến	....		
Từ.....đến	.....		
Thời gian chờ	.....		
TỔNG TIỀN:.....			

# Sơ đồ 3 block



# Chồng hàm và Phạm vi biến

- Cùng tên nhưng khác đối số truyền vào (DEMO sau)
- Phạm vi biến
  - Biến cục bộ (local variables) : Biến nằm trong hàm => chỉ có giá trị trong thân hàm đó => Ra khỏi hàm biến không có ý nghĩa và giá trị
  - Biến toàn cục : Biến khai báo ngoài hàm nhưng được xài bên trong các khối code



```
float tinhTong(int a, int b){  
    float kq;  
    kq = a + b;  
    return kq;  
}
```

Trong hàm bên, khi ra khỏi hàm tinhTong, các giá trị biến a, b hay biến kq sẽ không có giá trị

# Phép chia số nguyên

- Int a =4;
- Int b = 5;
- Float kq =a/b; => kq = 0;
- Cách 1: kq = a \* 1.0/ b; hoặc kq =a/b \* 1.0
- Cách 2: kq = (float) a/ b; Ép kiểu của a

# BÀI TẬP : TÍNH THUẾ THU NHẬP CÁ NHÂN

Viết chương trình nhập vào thông tin của 1 cá nhân (Họ tên, tổng thu nhập năm, số người phụ thuộc). Tính và xuất thuế thu nhập cá nhân phải trả theo quy định sau:

- ❖ **Thu nhập chịu thuế = Tổng thu nhập năm - 4tr- Số người phụ thuộc \* 1.6tr**

Thu nhập chịu thuế(triệu)	Thuế suất (%)
Đến 60	5
Trên 60 đến 120	10
Trên 120 đến 210	15
Trên 210 đến 384	20
Trên 384 đến 624	25
Trên 624 đến 960	30
Trên 960	35