

Kiến thức nền

Biến, Hằng, Toán tử, Mô hình 3 khối



Biến, kiểu dữ liệu, mô hình 3 khối

Khái niệm và bài tập về biến, phép gán, tăng giảm biến

Mô hình 3 khối & luyện tập

Các câu lệnh điều kiện & bài tập

Các câu lệnh điều kiện

Các câu lệnh điều kiện & bài tập

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	-32,768 -> 32,767
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

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) - lưu trữ địa chỉ vùng nhớ

Sử dụng biến

```
//Cách 1
// Khai báo biến
int number;
//Khởi tạo biến
number = 6;

// Cách 2: Khai báo và khởi tạo
int number2 = 6;

boolean check = true;
char digit = 'C';
double salary = 50.4;

double sum = salary;
```

```
check = number;  
number2 = sum ;
```

Kiểu dữ liệu - Kiểu String



String

```
String s1, s2;
```

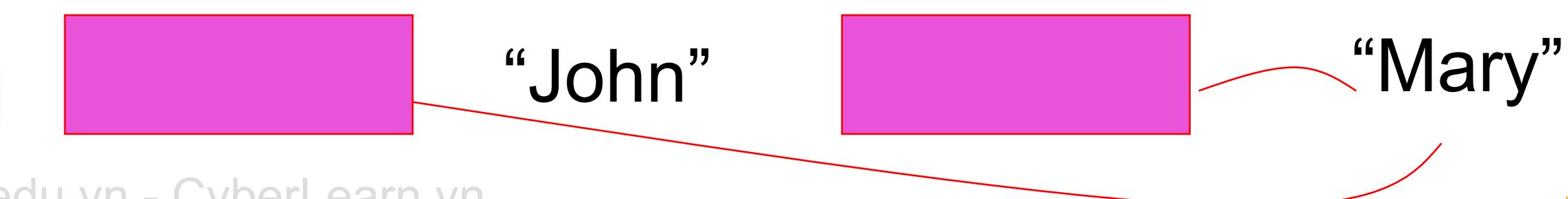
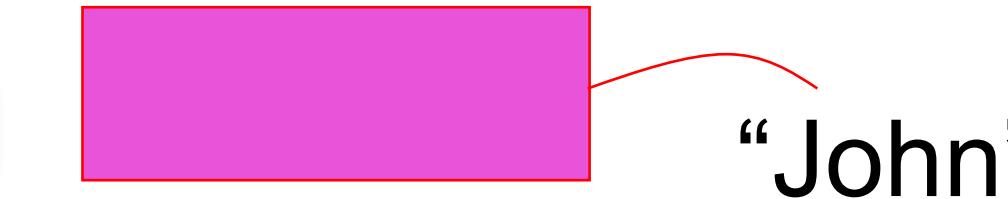
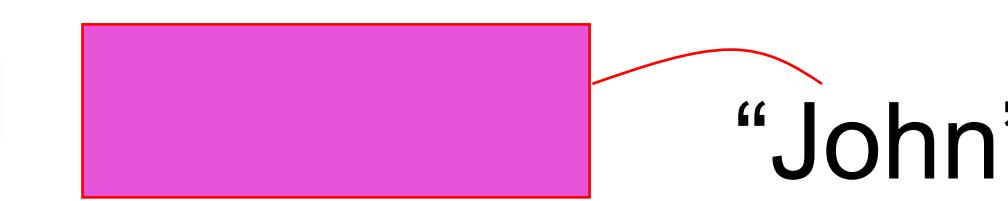
```
s1 = "John";
```

```
s2 = "Mary";
```

```
s1 = s2;
```

s1

s2



Sử dụng biến



- Lưu trữ dữ liệu tạm thời để xử lý
- 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 (**gia tri** → **giaTri**)
- Không được sử dụng tiếng Việt có dấu
- Đặt tên biến có ý nghĩa → Bảo trì, người khác đọc Code
- Tên biến viết thường chữ đầu (kiểu Lạc đà - camelCase)

Ví dụ:

```
int soLuong;  
float donGia;
```



Tìm lỗi trong các lệnh sau

• • •



- int i = 9
- int j = 1.0;
- boolean done = “false”;
- Int k = 1;
- double k; m = 5e-13;
- char class = ‘A’;
- String s = ‘W’;

Lệnh nhập, biến, phép gán



```
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
```

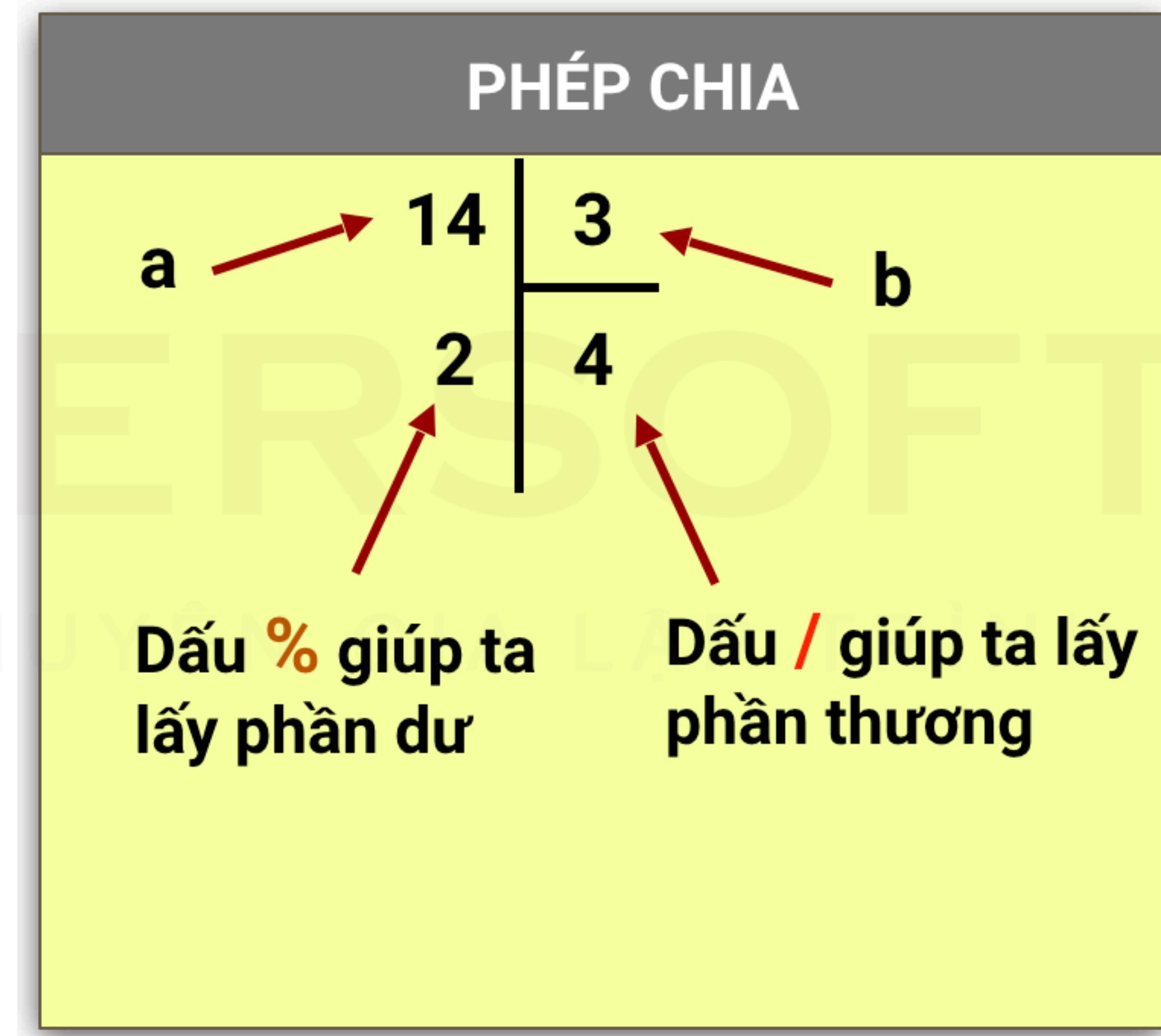
```
// Chuyển đổi chuỗi nhập là SỐ sang SỐ
String chuoiNhap = scan.nextLine();
int so1 = Integer.parseInt(chuoiNhap); // Đổi sang kiểu int
float so2 = Float.parseFloat(chuoiNhap); // Đổi sang kiểu float
int so3 = Double.parseDouble(chuoiNhap); // Đổi sang kiểu float
```



Các loại 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 = 11`
`int tich = a * b; => tich = 42`
`int thuong = a / b; => thuong = 4`
`int phanDu = a % b; => phanDu = 2`



Toán tử tăng giảm biến `++, --, += , -=, *=`

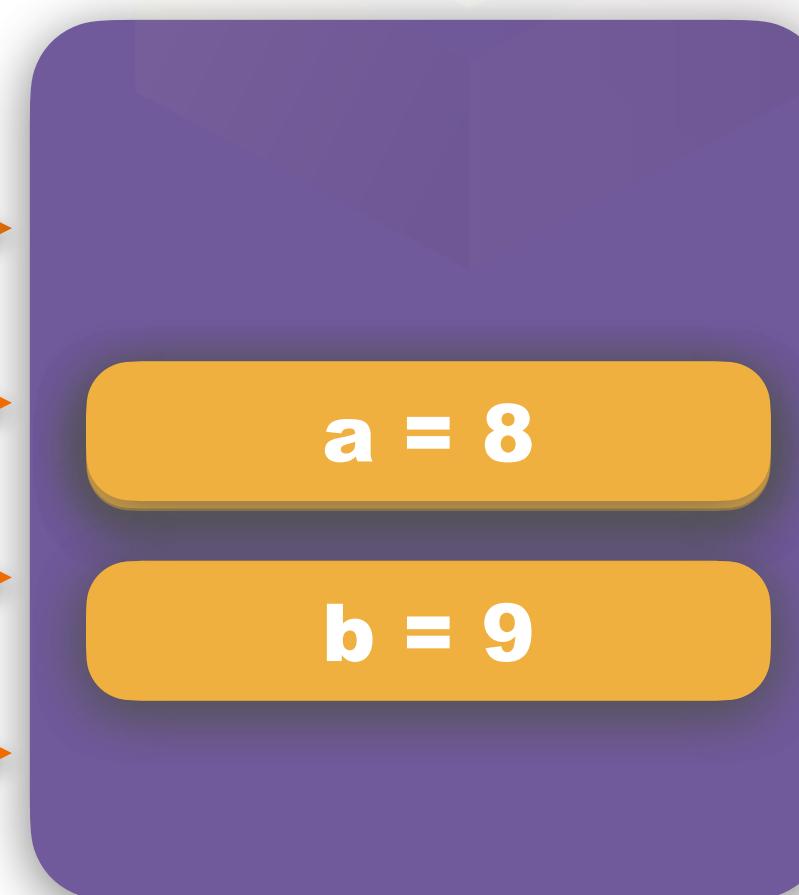


- 1) `x++;` giống `x = x + 1;` // tăng thêm 1
- 2) `x--;` giống `x = x - 1;` giống `x -= 1;` // giảm bớt 1
- 3) `x += y;` giống `x = x + y;`
- 4) `x *= y` giống `x = x * y;`
- 5) `x /= y` giống `x = x/y;`

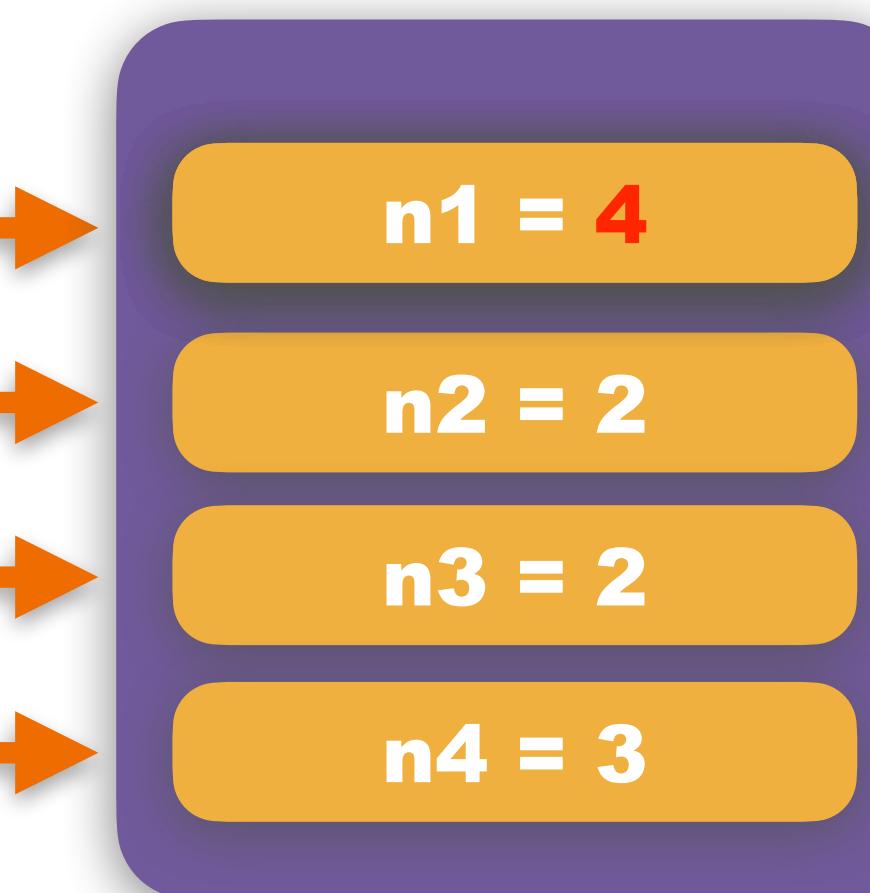
Ví dụ:

```
int a, b;
```

```
a = 5;  
b = 2 + (++a);  
a++;  
b = 2 + (a++);
```



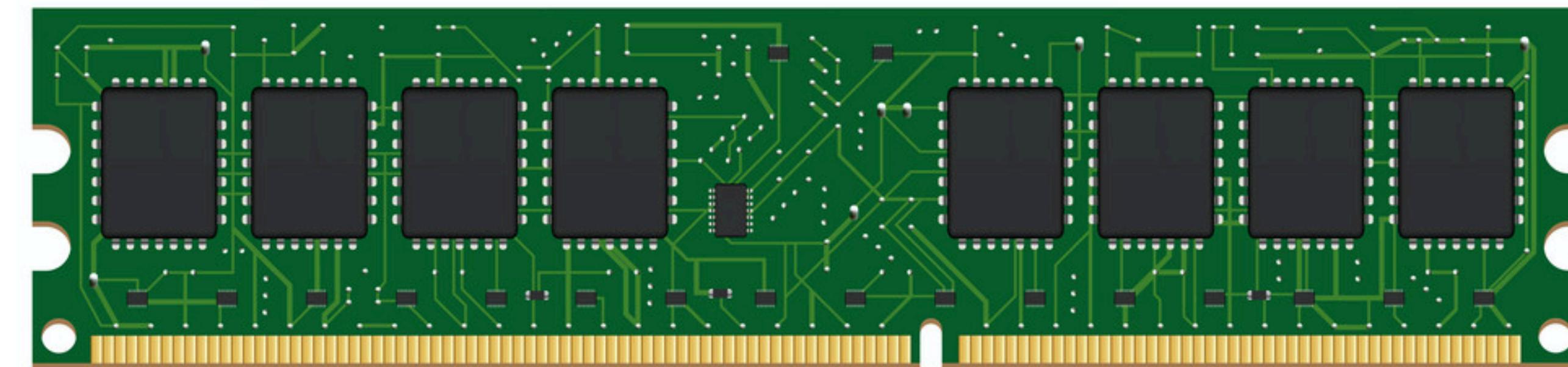
```
int n1 = 1;  
int n2 = ++n1;  
int n3 = n1++;  
int n4 = n1++;
```



Thảo luận sâu về biến & lưu trữ bộ nhớ



Bộ nhớ RAM

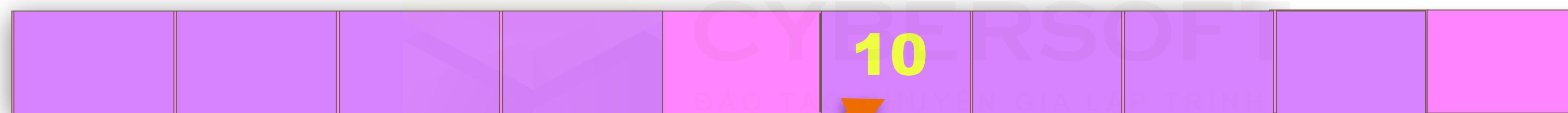


1004

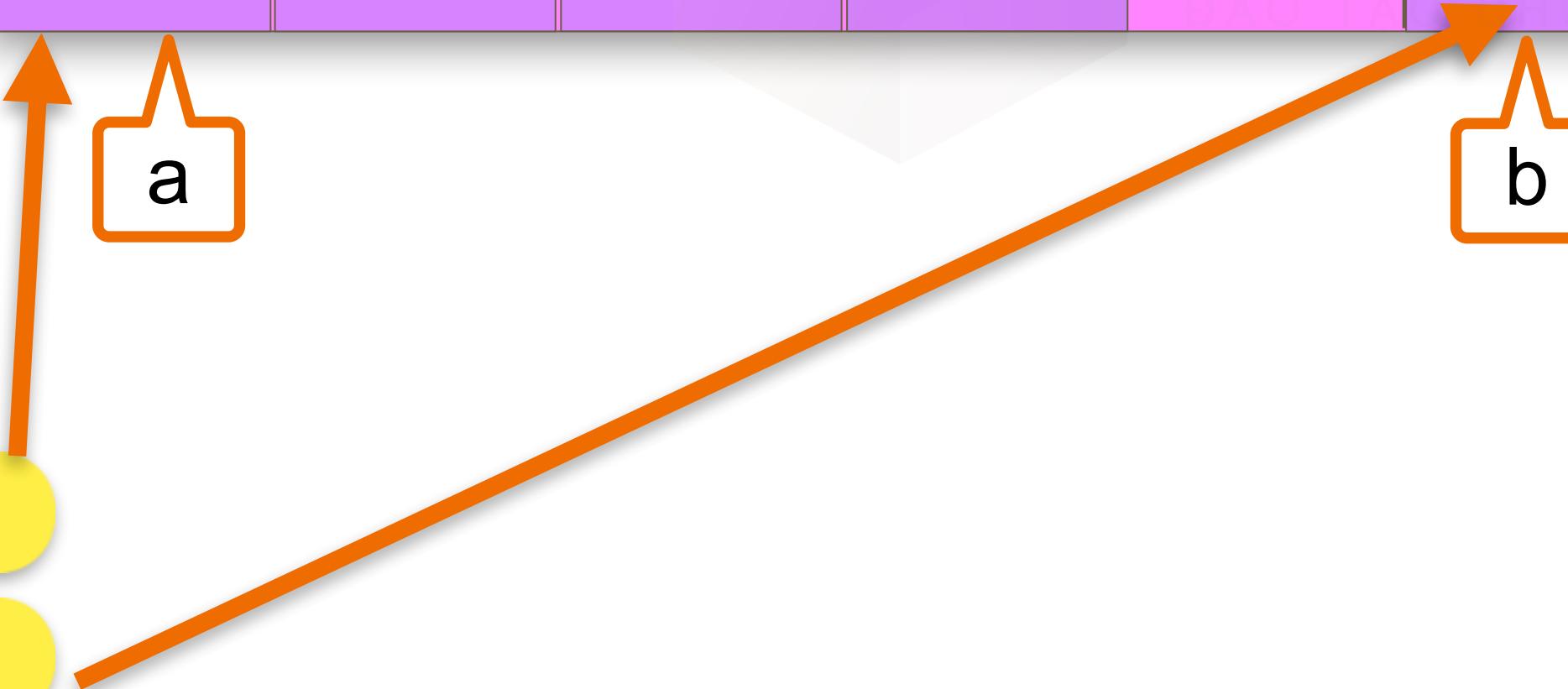
1005

1007

1009



`int a;`



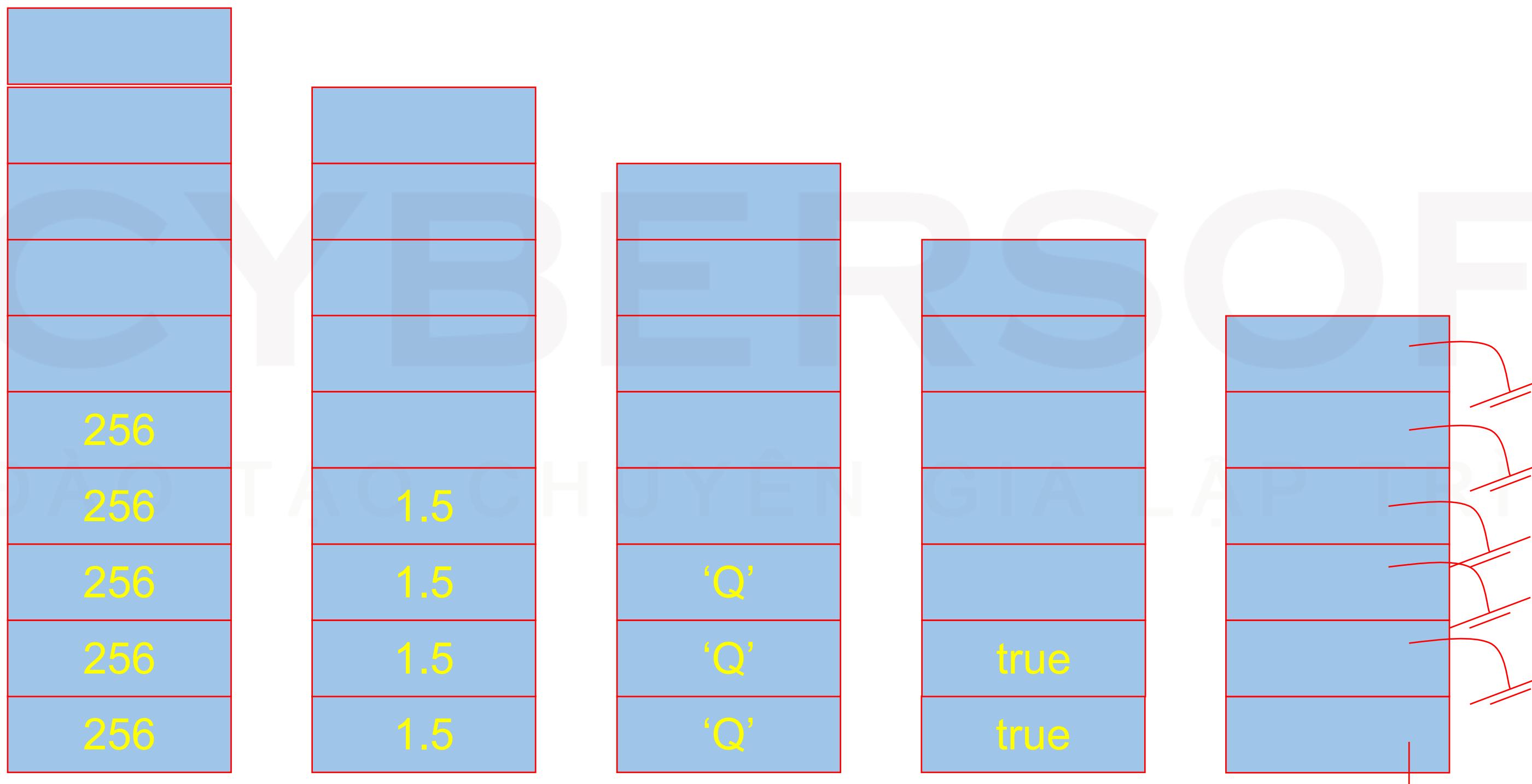
`int b = 10;`

Quá trình cấp phát bộ nhớ cho biến



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

x d c b s



Quá trình cấp phát bộ nhớ cho biến



```
int num1;  
int num2;  
int num3;  
num1 = 18;  
num1 = num1 + 27;  
num2 = num1;  
num3 = num2 / 5;  
num3 = num3 / 4;
```

CYBERSOFT
ĐÀO TẠO CHUYÊN GIA LẬP TRÌNH

45

num1

45

num2

2

num3

Hằng số



- Lý do ? LIVE DEMO TÍNH DIỆN TÍNH, CHU VI HÌNH TRÒN

- Cú pháp trong Java :

- **final [kieu_du_lieu] [ten_hang_so] = [gia_tri];**
- Ví dụ:
 - **final double PI = 3.14;**
 - **final int SO_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



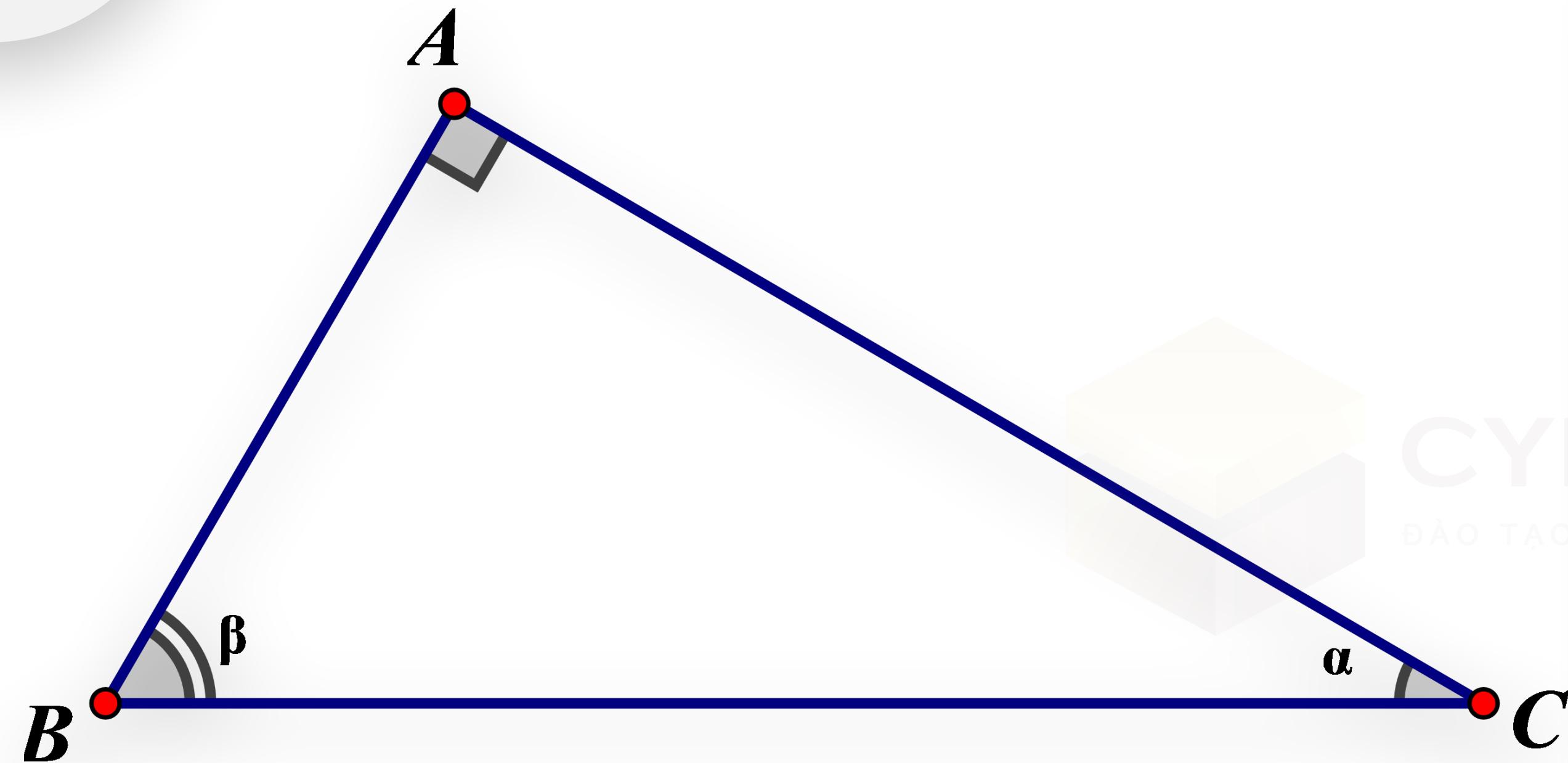
MÔ HÌNH 3 KHỐI



ĐẦU VÀO
Người dùng nhập vào



ĐẦU RA
Kết quả hiển thị



Bài tập luyện 1

Viết chương trình tính chiều dài cạnh huyền của một tam giác vuông khi biết 2 cạnh góc vuông.

MÔ HÌNH 3 KHỐI

- Cạnh góc vuông 1
- Cạnh góc vuông 2

- Bước 1: Tạo 2 biến **edge1** và **edge2**
- Bước 2: In thông báo nhập cạnh 1
- Bước 3: Cho người dùng nhập cạnh 1 và gán **edge1**
- Bước 4: In thông báo nhập cạnh 2
- Bước 5: Cho người dùng nhập cạnh 2 và gán **edge2**
- Bước 6: Tạo biến cạnh huyền **edge3**
- Bước 7: Sử dụng công thức **Pytago**
$$\text{edge3} * \text{edge3} = \text{edge1} * \text{edge1} + \text{edge2} * \text{edge2}$$
$$\Rightarrow \text{edge3} = \text{Math.sqrt}(\text{edge1} * \text{edge1} + \text{edge2} * \text{edge2});$$
- Bước 8: In kết quả sum ra màn hình

Kết quả cạnh huyền



Bài tập luyện 2

Viết chương trình tính nhập vào số nguyên dương n với 3 ký số, tính và xuất tổng 3 ký số của n.

MÔ HÌNH 3 KHỐI

- Số nguyên dương n có 3 ký số
(ví dụ : 586, 219,...)
- $586 \rightarrow 5 + 8 + 6 = 19$
- $219 \rightarrow 2 + 1 + 9 = 12$

- Bước 1: Tạo biến ***n, unit, ten, hundred, sum***
- Bước 2: In thông báo nhập n
- Bước 3: Cho người dùng nhập vào n
- Bước 3: Tách số hàng trăm theo công thức:
$$\text{hundred} = n/100;$$
- Bước 4: Tách số hàng chục theo công thức
$$\text{ten} = ((n \% 100 / 10));$$
- Bước 5: Tách số hàng đơn vị theo công thức
$$\text{unit} = (n \% 10);$$
- Bước 6: Tính $\text{sum} = \text{hundred} + \text{ten} + \text{unit};$
- Bước 7: In kết quả sum ra màn hình

In kết quả sum ra màn hình

CASE STUDY (THẢO LUẬN NHÓM)

Một bộ phim phát trong rạp chiếu địa phương thu hút rất nhiều người. Để giúp một tổ chức từ thiện địa phương, chủ rạp đã quyết định quyên góp cho tổ chức từ thiện một phần trong tổng số tiền được tạo ra từ bộ phim. Hãy thực hiện một chương trình cho người dùng nhập tên phim, giá vé người lớn, giá vé trẻ em, số vé người lớn đã bán, số vé trẻ em được bán và tỷ lệ phần trăm của tổng số tiền quyên góp cho tổ chức từ thiện.

Yêu cầu: Phân tích mô hình 3 khối, viết thuật toán xử lý và cài đặt chương trình

Tên phim:Zero to Hero at CyberSoft

Số vé đã bán: 2650

Doanh thu: 9150

Trích % từ thiên: 10%

Tổng tiền trích từ thiên: 915

Tổng thu được sau khi trừ: 8235

MÔ HÌNH 3 KHỐI

- Tên phim
- Giá vé người lớn
- Giá vé em bé
- Số vé người lớn
- Số vé em bé
- % làm từ thiện

- Bước 1: Khai báo các biến *tenPhim, giaVeLon, giaVeEm, soVeLon, soVeEm, phanTramTuThien, tongThu, tuThien, conLai*
- Bước 2: Thông báo nhập tên phim
- Bước 3: Lấy dữ liệu nhập vào cho biến tên phim
- Bước 4: Thông báo nhập giá vé người lớn
- Bước 5: Lấy dữ liệu nhập cho biến giá vé người lớn
- Bước 6: Thông báo nhập giá vé trẻ em
- Bước 7: Lấy dữ liệu nhập cho biến giá vé trẻ em
- Bước 8: Thông báo nhập số vé người lớn đã bán
- Bước 9: Lấy dữ liệu nhập cho biến số vé người lớn đã bán
- Bước 10: Thông báo nhập số vé trẻ em đã bán
- Bước 11: Lấy dữ liệu nhập cho biến số vé trẻ em đã bán
- Bước 13: Thông báo nhập % tiền làm từ thiện
- Bước 14: Lấy dữ liệu nhập cho % làm từ thiện
- Bước 15: Tính tổng tiền thu được
- Bước 16: Tính tiền cho từ thiện
- Bước 17: Tính tiền còn lại sau khi trừ đi tiền từ thiện
- Bước 18: Xuất kết quả

In kết quả như biểu mẫu ra màn hình

```
String text = String.format("%20s", "CyberLearn");
```

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

```
String text = String.format("%-20s", "CyberLearn");
```

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

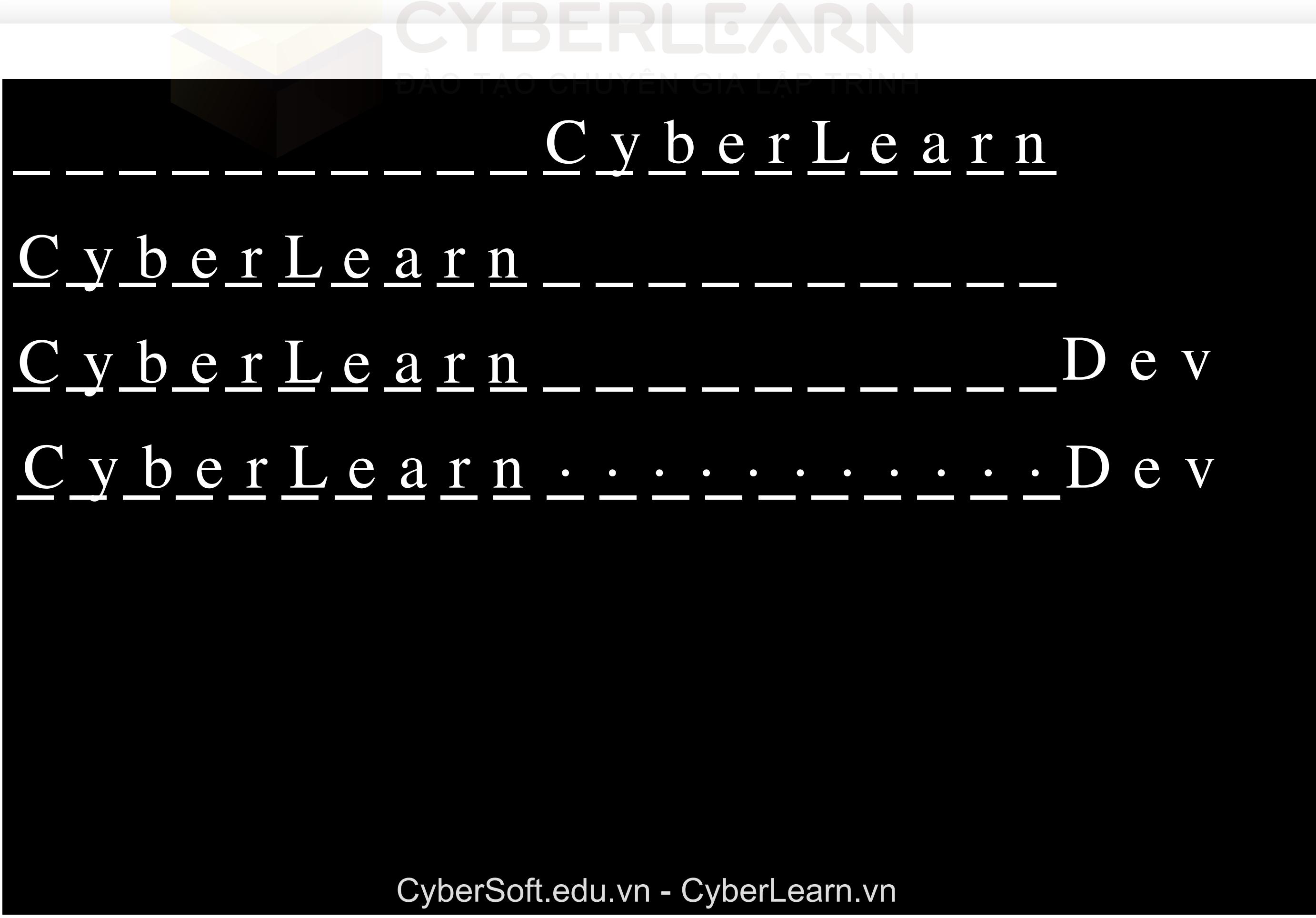
```
String text = String.format("%-20s", "CyberLearn") + "Dev";
```

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

```
String text = String.format("%-20s", "CyberLearn").replace(' ', '.') + "Dev";
```

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

String.format



Tăng - giảm biến - Trắc nghiệm



```
int a;  
int b = 2;  
a = 4;  
b = (++b) + (++a);
```

```
int a = 3;  
int b = 1;  
a += b;  
b = (b++) + (a++)
```

a = 6; b = 7
a = 5; b = 7
a = 5; b = 8
a = 6; b = 8

a = 6; b = 7
a = 5; b = 5
a = 5; b = 8
a = 6; b = 8

Tăng - giảm biến - Trắc nghiệm



```
int a = 3;  
int b = 4;  
a = b--;  
b = (--b) + (++a);
```

```
int a = 3;  
int b = 1;  
a -= b;  
b = (b--) + (-a);
```

a = 6; b = 7
a = 5; b = 7
a = 5; b = 8
a = 6; b = 8

a = 1; b = 2
a = 3; b = 1
a = 1; b = 3
a = 2; b = 1

Tăng - giảm biến - Trắc nghiệm



```
int a = 5;  
int b = 6;  
a = (b++) + 3;  
c = 2 * a + (++b);  
b = 2 * (++c) - (a++);
```

a = 10; b = 44; c = 27

a = 9; b = 54; c = 26;

a = 10; b = 45; c = 27

a = 10; b = 45; c = 25

```
int a = 3;  
int b = 5;  
double c = 14.1;  
double sum;  
sum = a + b + c;  
c /= a; b += c - a;  
a *= 2 * b + c;
```

a = 50; b = 6; c = 4.2; sum =21

a = 50; b = 8; c = 4.1; sum =21.1

a = 48; b = 6; c = 4.7; sum =22.1

a = 50; b = 6; c = 4.7; sum =22.1

Một số điểm cần lưu ý

Biến để lưu trữ dữ liệu

01

Không được xài hardcoded, tạo HÀNG SỐ khi cần

02

Chú ý toán tử ++, -- trước và sau

03

Luôn phân tích, viết thuật toán trước khi code

04

