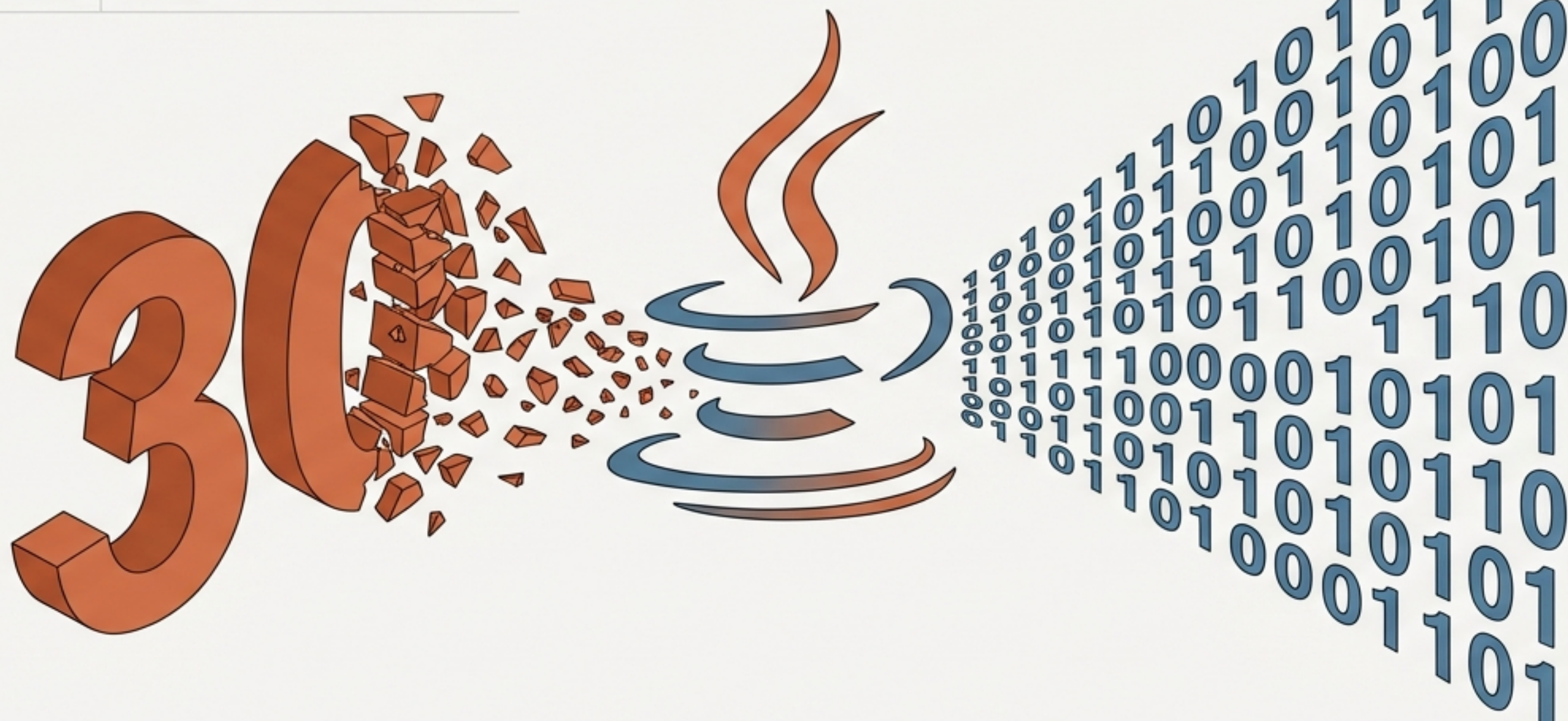


Chuyển đổi dữ liệu giữa hai thế giới: Thập phân và Nhị phân trong Java

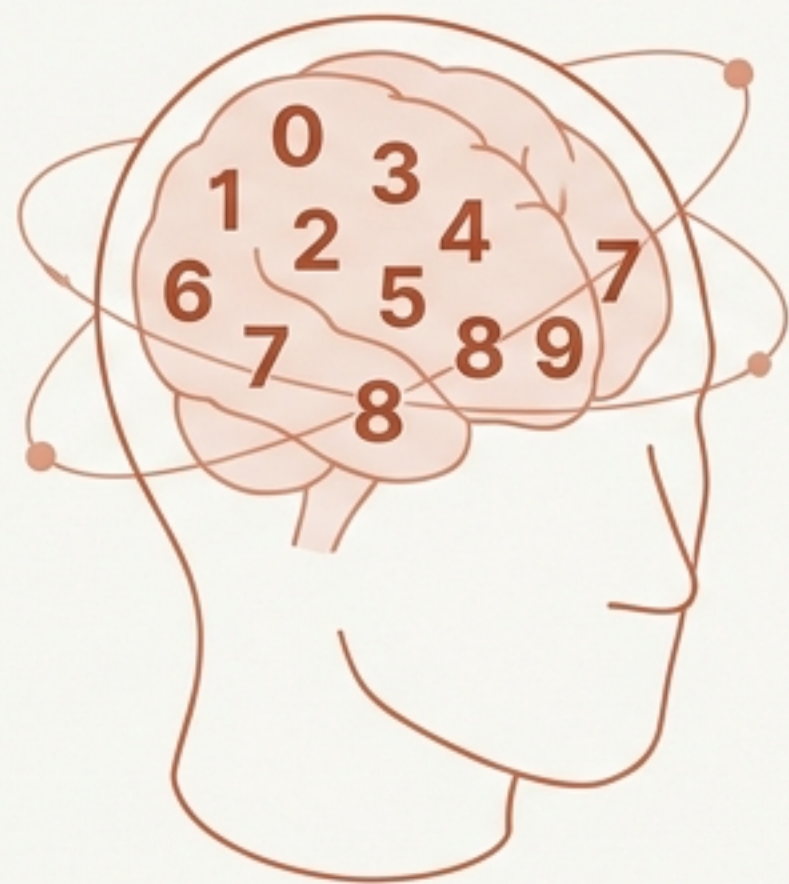
Học lập trình C/C++, Web, Game, Mobile & AI

Chuyên mục: R&D Programmer

Tác giả: Phạm Xuân Hoài

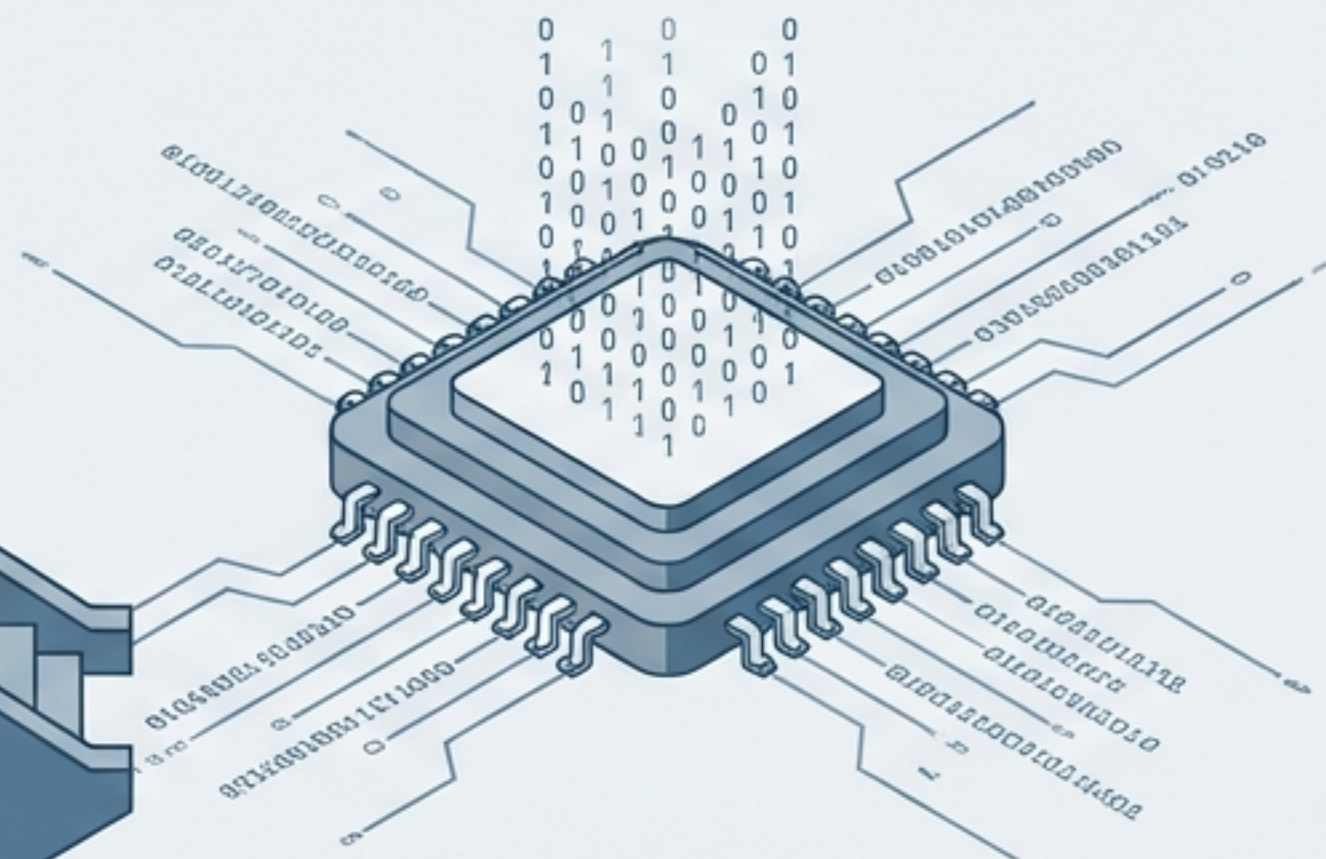
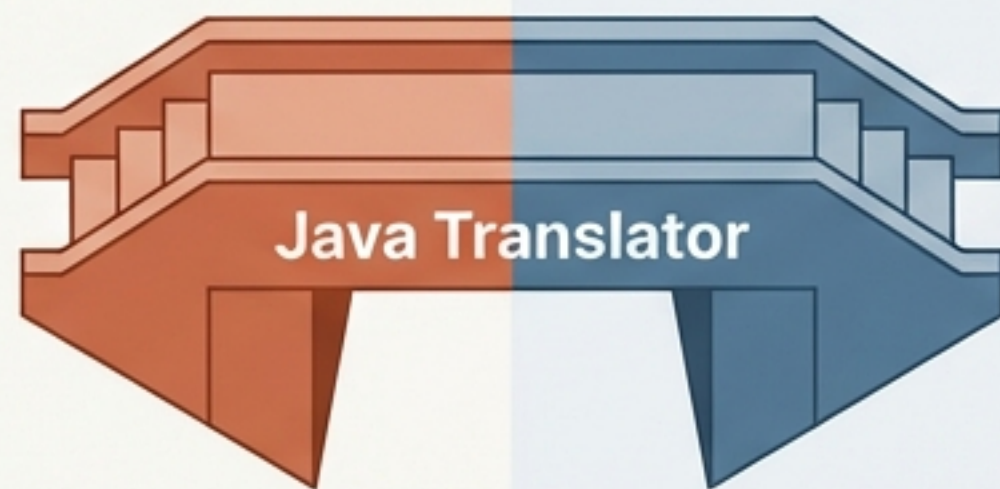


Rào cản ngôn ngữ giữa con người và máy tính



Thế giới con người

Tư duy và giao tiếp bằng hệ thập phân (Decimal - Cơ số 10).



Thế giới máy tính

Xử lý mọi dữ liệu qua các trạng thái điện, đại diện bằng hệ nhị phân (Binary - Cơ số 2).

Mục tiêu bài toán: Xây dựng một cổng dịch thuật bằng ngôn ngữ Java để chuyển đổi qua lại giữa hai hệ cơ số này.

Nguyên tắc dịch thuật từ Hệ 10 sang Hệ 2

Để chuyển một số (ví dụ: 30) sang hệ nhị phân, quy tắc cốt lõi là chia 2 liên tục. Tại mỗi bước chia, chúng ta chỉ quan tâm đến phần dư (0 hoặc 1).

$$30 \div 2 = 15$$

dư 0

$$15 \div 2 = 7$$

dư 1

$$7 \div 2 = 3$$

dư 1

$$3 \div 2 = 1$$

dư 1

$$1 \div 2 = 0$$

dư 1

Quy tắc đảo ngược phần dư

Các phần dư thu được từ quá trình chia liên tục không thể đọc từ trên xuống.

Để tạo thành chuỗi nhị phân chính xác, chúng ta bắt buộc phải đảo ngược lại (đọc từ dưới lên trên).

dư 0

dư 1

dư 1

dư 1

dư 1

11110



Đưa dữ liệu trở về định dạng Hệ 10

Quy tắc: Nhân từng bit nhị phân với các lũy thừa của 2 tương ứng với vị trí của nó, sau đó cộng tổng lại.

1	1	1	1	0
---	---	---	---	---



$\times 2^4$

$\times 2^3$

$\times 2^2$

$\times 2^1$

$\times 2^0$

16

+

8

+

4

+

2

+

0

=

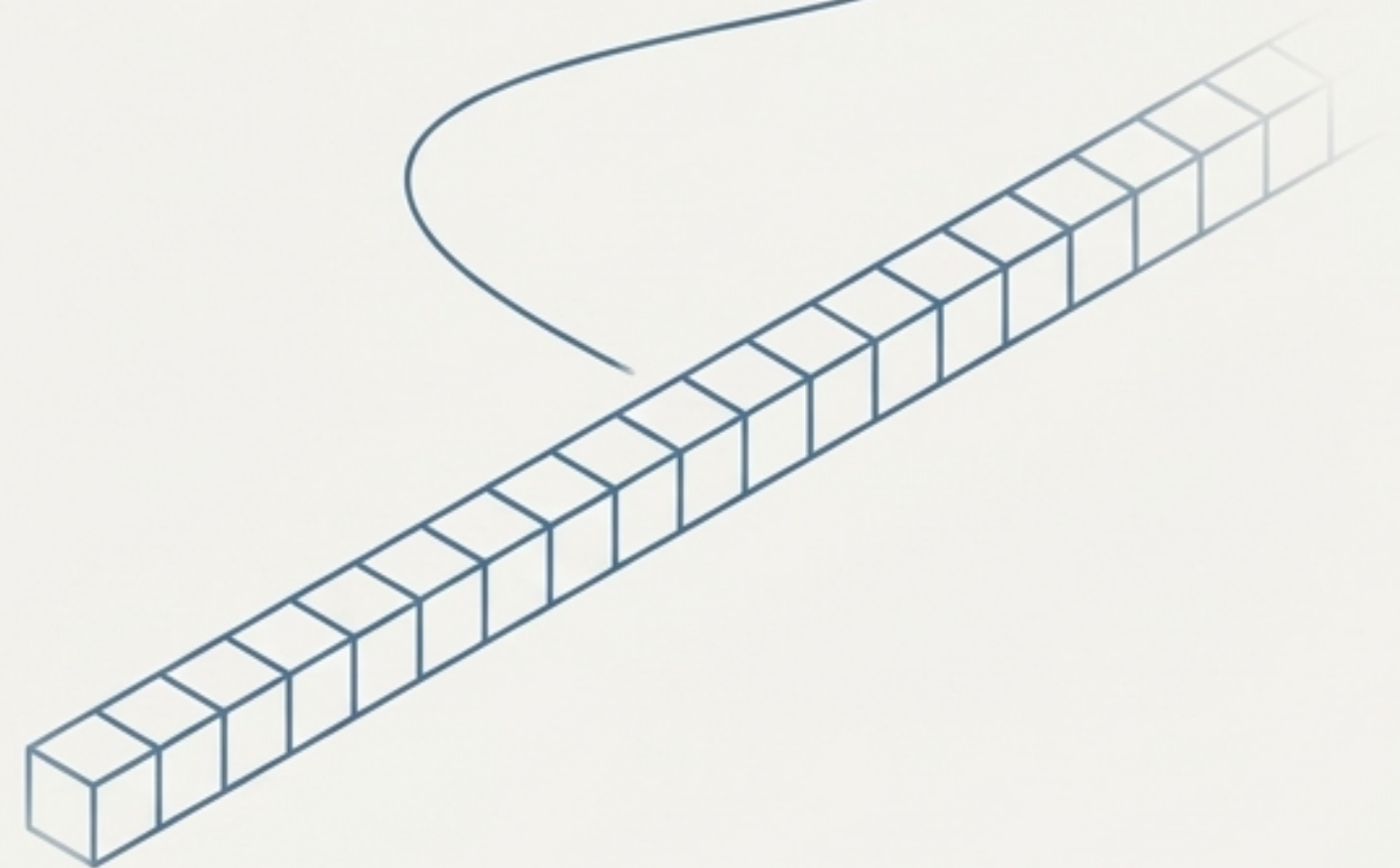
30

Chuẩn bị không gian lưu trữ dữ liệu trong Java

Khai báo biến: Xác định số thập phân cần chuyển đổi.

Mảng lưu các bit: Khởi tạo một mảng (Array) chuyên dụng để hứng và lưu trữ lần lượt các bit 0 và 1 sinh ra trong quá trình chia cho 2.

```
int number = 30;  
int[] binaryArray = new int[32];
```



Động cơ cốt lõi: Vòng lặp chia 2 liên tục

Sử dụng vòng lặp while để xử lý đến khi số ban đầu bằng 0.

```
while (number > 0) {  
    binaryArray[index++] = number % 2;  
    number = number / 2;  
}
```

Toán tử % 2: Lấy phần dư (trích xuất bit).

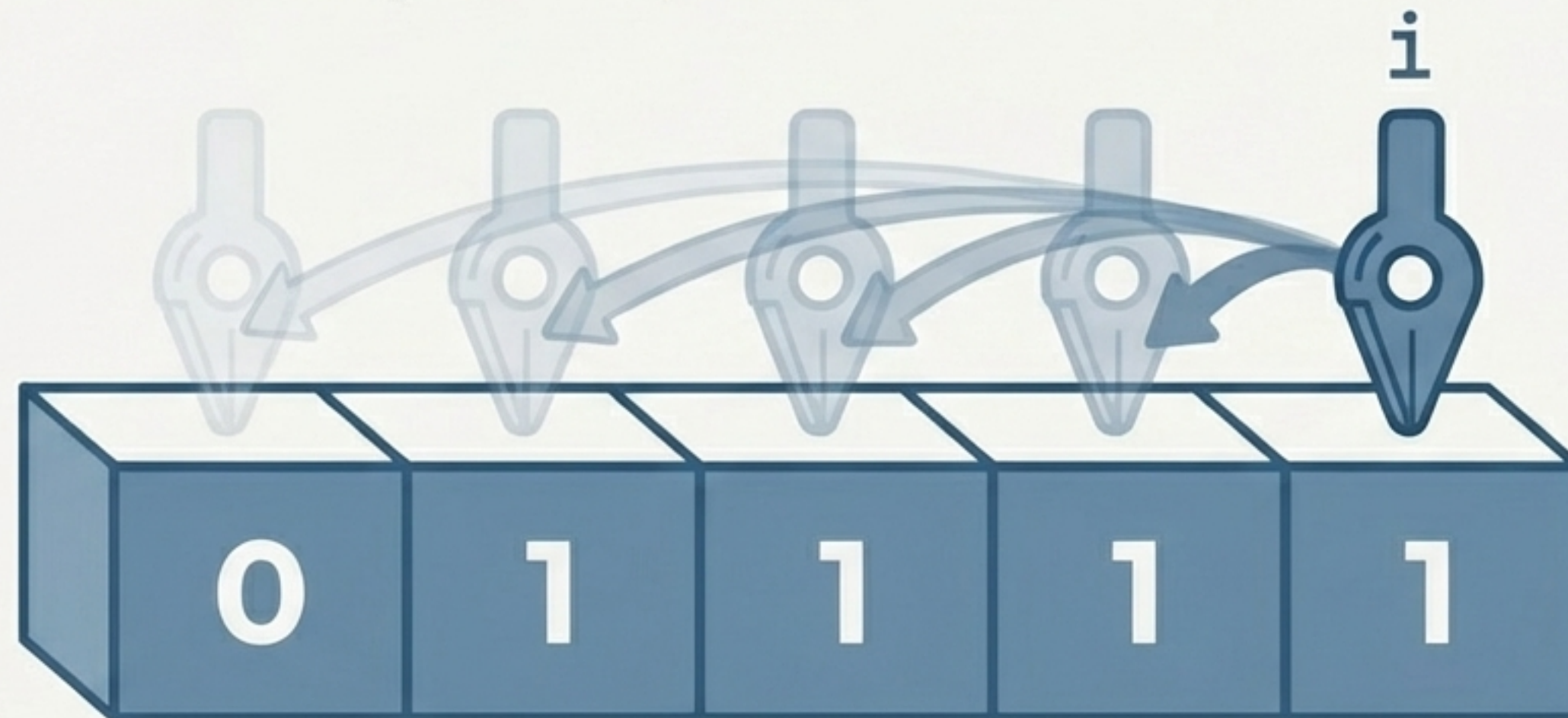
Toán tử / 2: Lấy phần nguyên để tiếp tục vòng lặp tiếp theo.

Xuất chuỗi nhị phân bằng vòng lặp ngược

Vì các phần dư được lưu vào mảng theo thứ tự sinh ra, việc đọc xuôi sẽ cho kết quả sai.

Giải pháp: Sử dụng vòng lặp for chạy từ cuối mảng về đầu để in ra số nhị phân chuẩn xác.

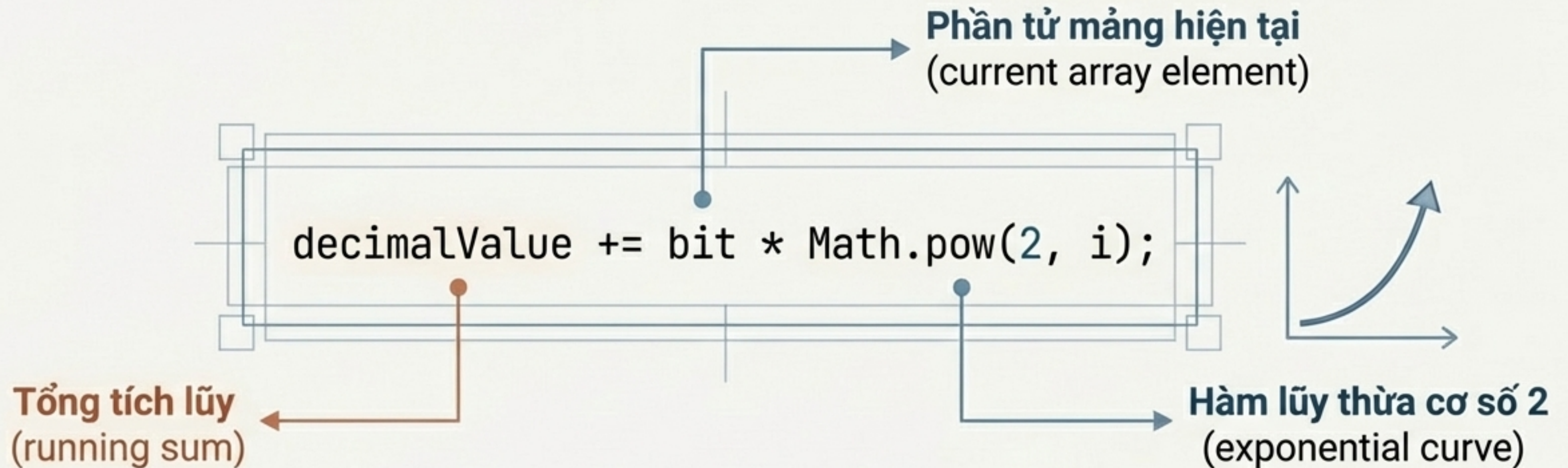
```
for(int i = index - 1; i >= 0; i--)
```



Mã hóa công thức chuyển đổi ngược về Hệ 10

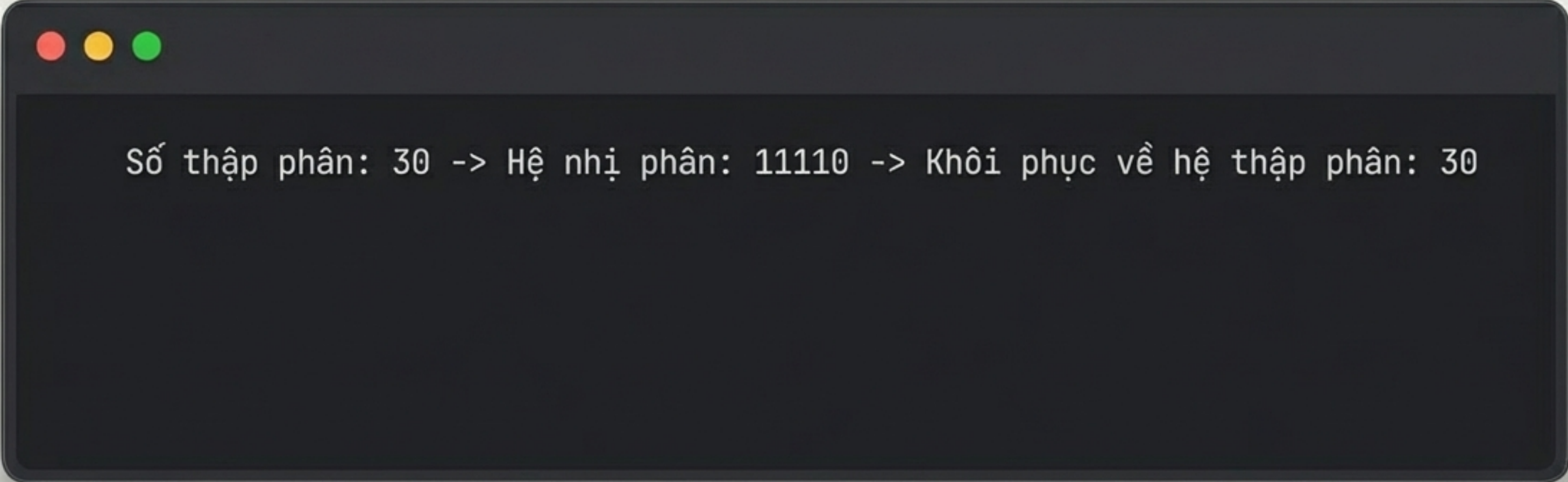
Áp dụng công thức lũy thừa cơ số 2 ngay trong mã lệnh.

Chạy vòng lặp qua từng bit, thực hiện phép nhân tương ứng với vị trí i , sau đó cộng dồn tất cả lại để khôi phục số thập phân ban đầu.



Kết quả biên dịch và thực thi

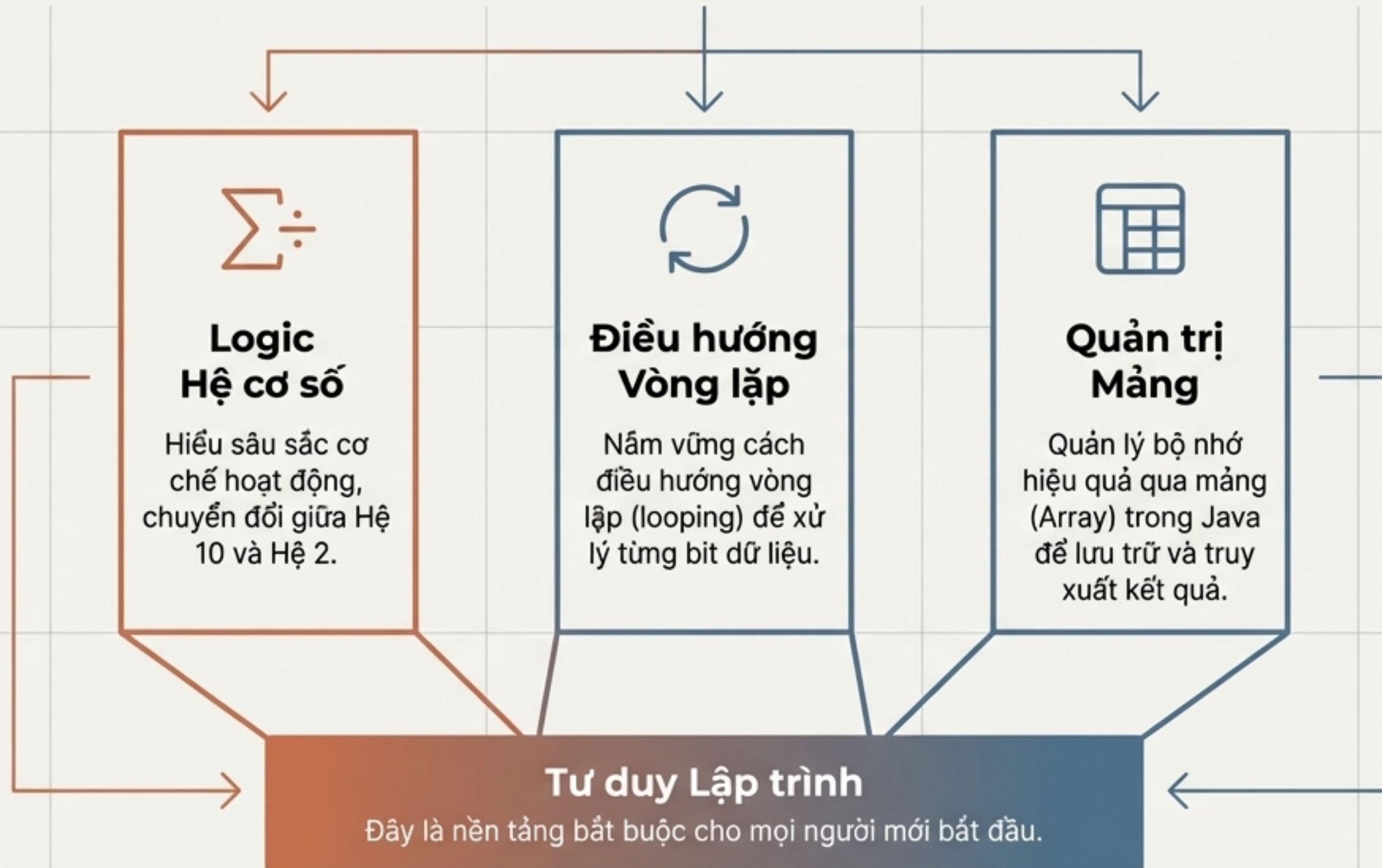
Toàn bộ quá trình chia, lưu trữ, đảo ngược và chuyển đổi ngược đã hoạt động trơn tru. Chương trình nhận đầu vào là hệ 10, in ra hệ 2, và khôi phục thành công về hệ 10.



```
Số thập phân: 30 -> Hệ nhị phân: 11110 -> Khôi phục về hệ thập phân: 30
```

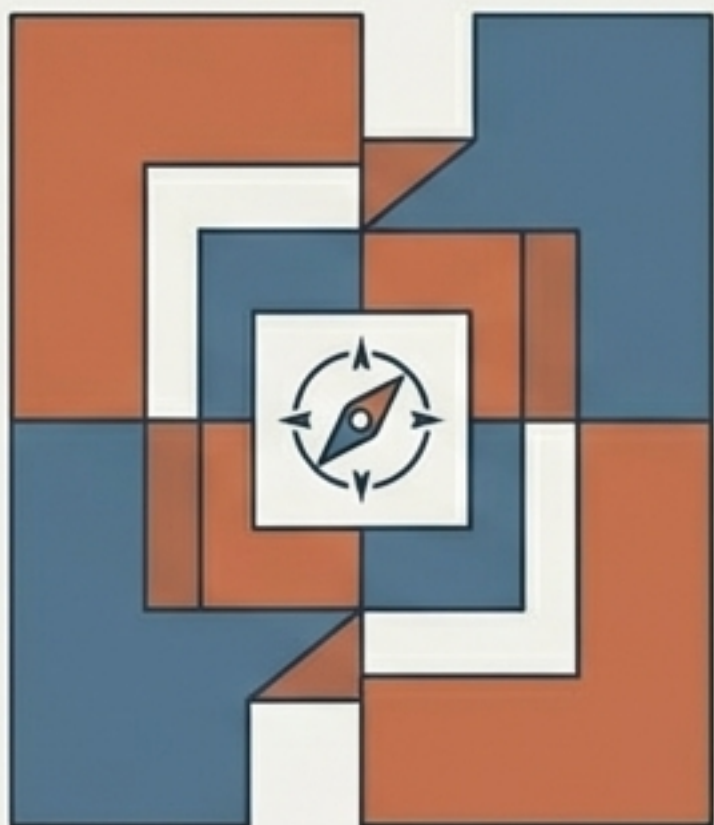
Bản chất là kỹ năng làm chủ cấu trúc dữ liệu cơ bản

Bài toán này không chỉ giải quyết việc chuyển đổi Hệ 10 sang Hệ 2 và ngược lại. Giá trị cốt lõi: Giúp lập trình viên hiểu sâu sắc cơ chế hoạt động, cách điều hướng vòng lặp và quản lý bộ nhớ qua mảng (Array) trong Java.

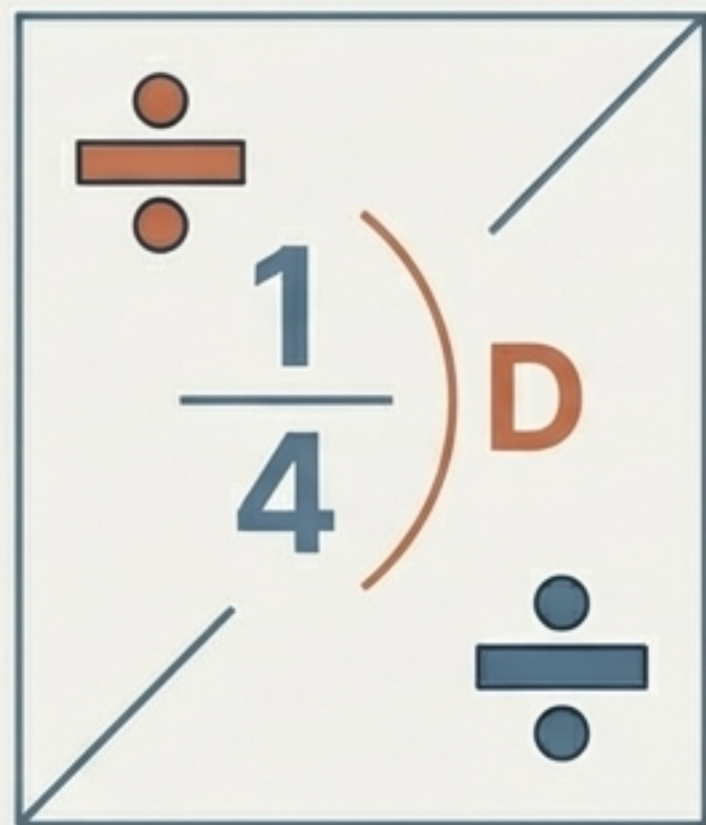


Tiếp tục hành trình R&D Programmer cùng Error404-Labs

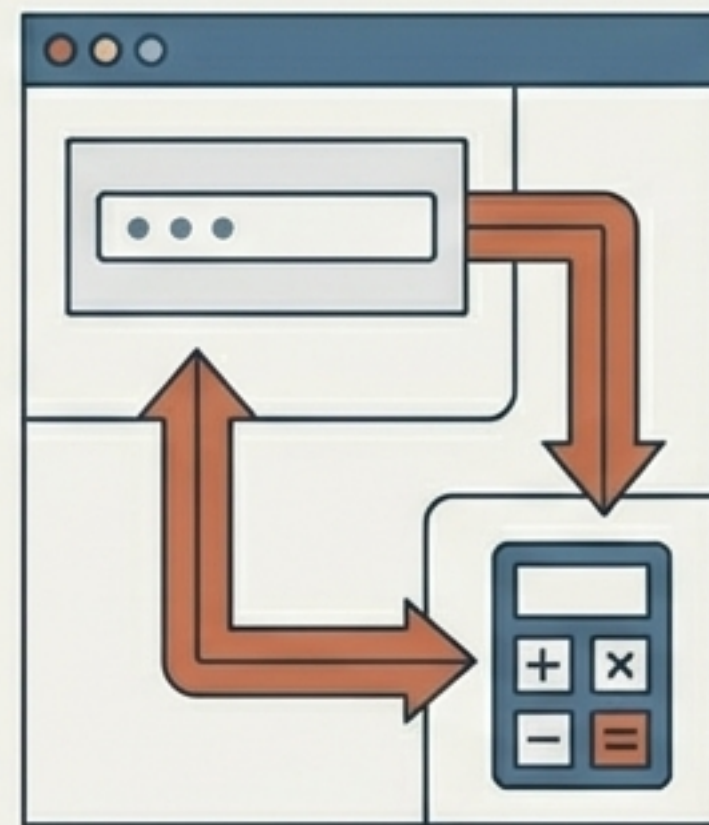
Củng cố nền tảng Java với các chuyên đề tiếp theo:



Enum trong Java – Khi Constant tiến hóa thành một thực thể hoàn chỉnh



Tối Giản Phân Số Bằng UCLN – Java Cơ Bản Nhưng Rất Chắc



Java Basic – Nhập dữ liệu an toàn & sử dụng Math