



## Session 03:

# Mảng



- 1. Giới thiệu kiểu dữ liệu mảng, cách khai báo, khởi tạo**
- 2. Các thao tác làm việc với mảng 1 chiều**
- 3. Sử dụng được cú pháp Java để thao tác với phương thức**
- 4. Các kỹ thuật lập trình với mảng**

# 1. Kiểu dữ liệu mảng, cách khai báo, khởi tạo



## Câu hỏi:

“Nếu chương trình chỉ xử lý 1–2 dữ liệu thì ổn. Vậy nếu dữ liệu tăng lên 50, 100, 1.000 thì sao?”

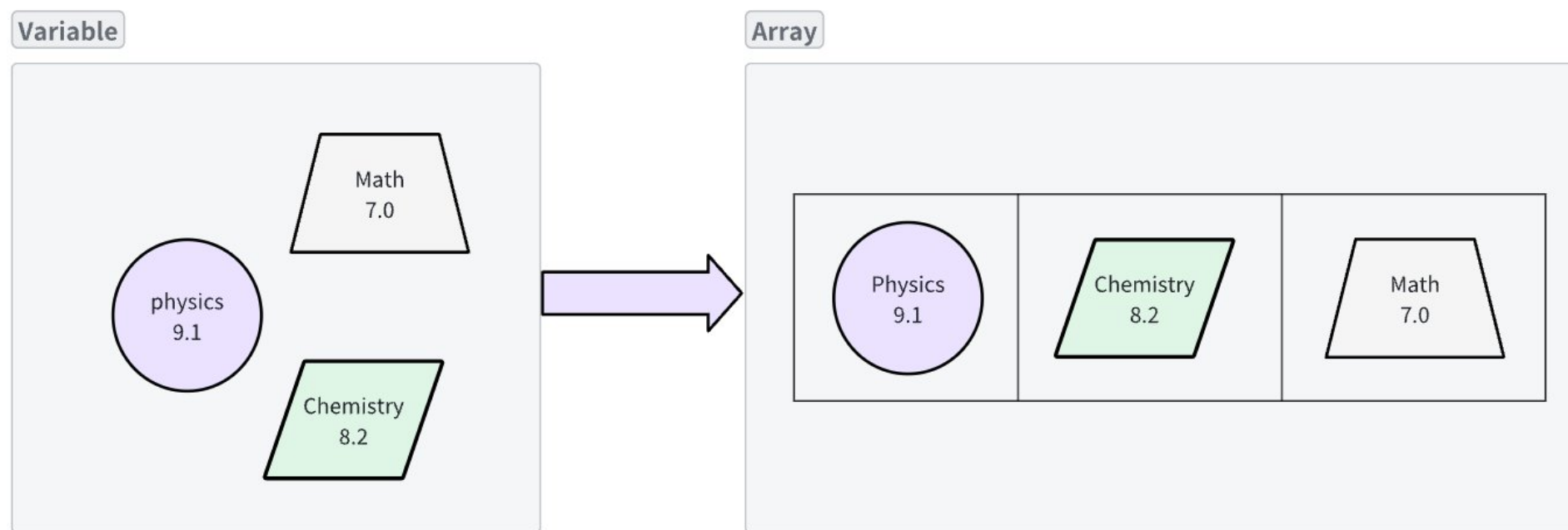
## Bài toán:

- Nhập điểm của 5 sinh viên
- Tính tổng và điểm trung bình

## Vấn đề:

- Phải khai báo **nhiều biến**
- Code dài và khó đọc
- Rất dễ nhầm lẫn

# 1. Kiểu dữ liệu mảng, cách khai báo, khởi tạo



→ Việc tạo ra từng biến cụ thể để lưu trữ điểm của sinh viên không thể giải quyết được yêu cầu trên → Mảng ra đời giải quyết bài toán lưu trữ 1 tập giá trị

# 1. Kiểu dữ liệu mảng, cách khai báo, khởi tạo

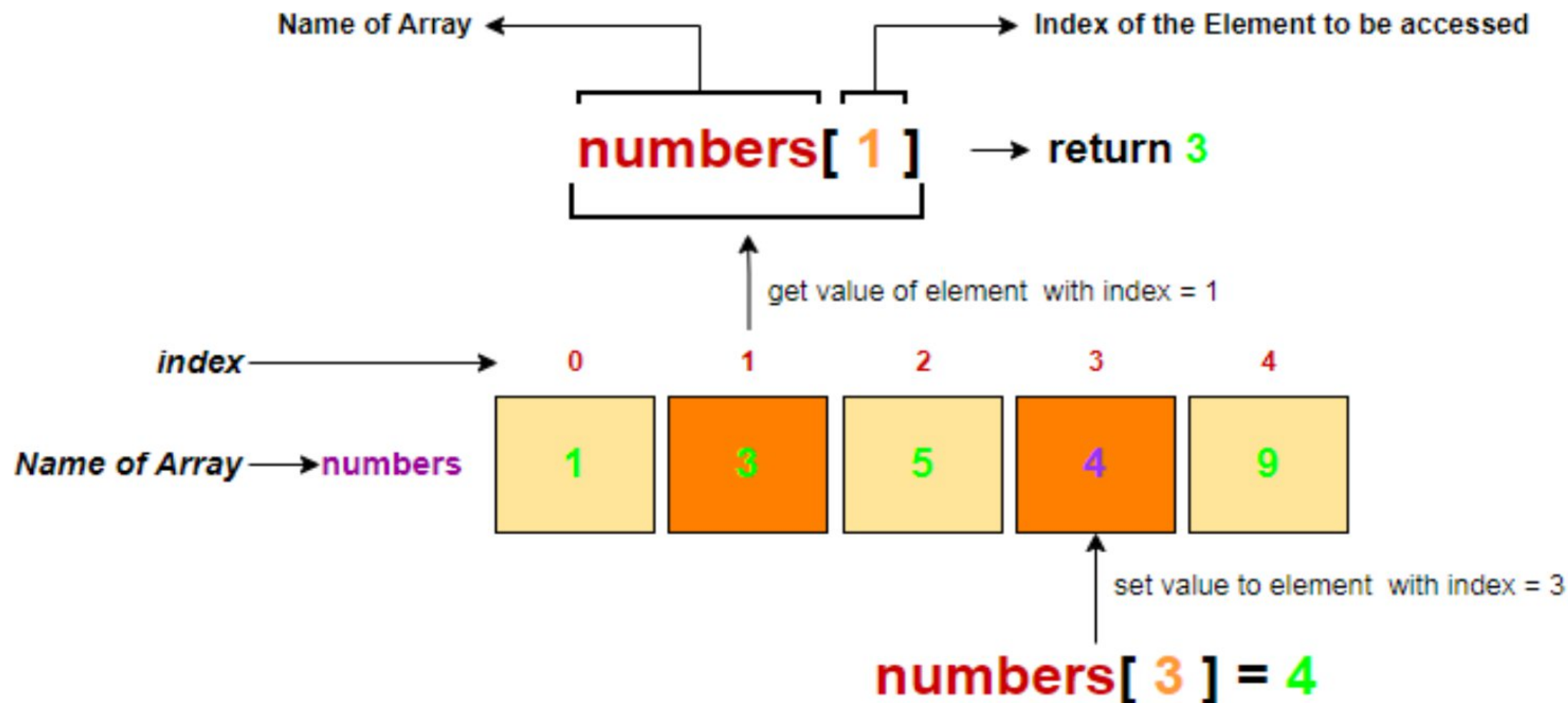


```
DataType[] arrayName = new DataType[numberOfElement]
```

## Trong đó:

- **DataType** - Kiểu dữ liệu mảng
- **arrayName** - Tên biến
- **numberOfElement** - Số lượng phần tử trong mảng
- **new** - Từ khoá khởi tạo đối tượng

# 1. Kiểu dữ liệu mảng, cách khai báo, khởi tạo



## 2. Các thao tác làm việc với mảng 1 chiều

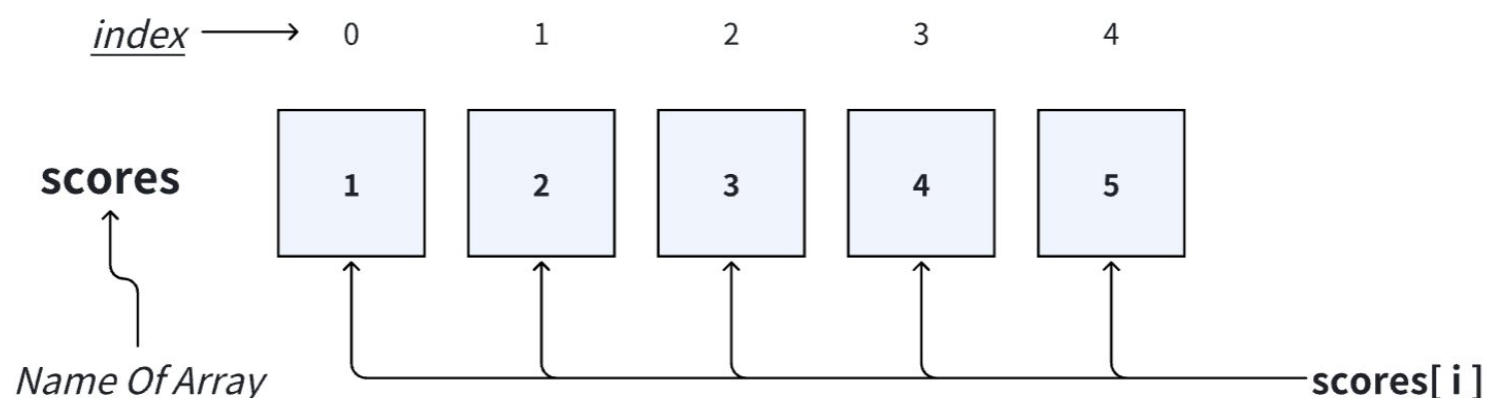


### Vậy thì:

- Làm sao để có thể hiển thị danh sách điểm sinh viên
- Làm sao để có thể thêm điểm mới cho sinh viên
- Làm sao để có thể cập nhật lại điểm của sinh viên
- Làm sao để có thể xóa điểm vừa thêm của sinh viên

## 2. Các thao tác làm việc với mảng 1 chiều

### Thao tác hiển thị danh sách điểm sinh viên

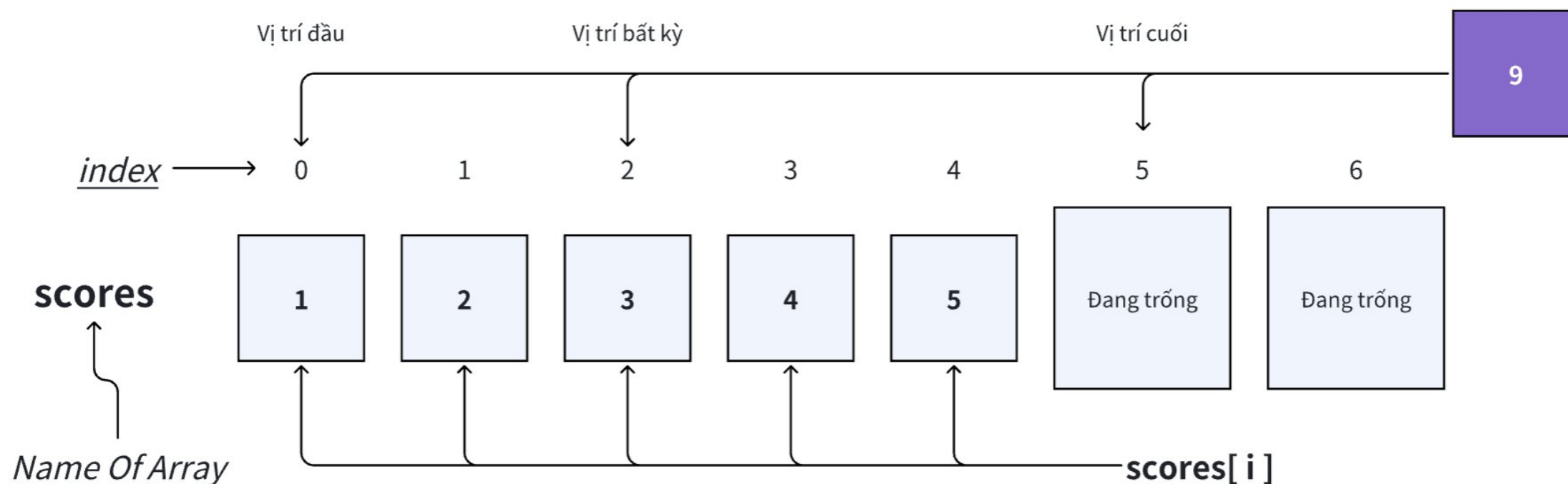


```
for (int i = 0 ; i < scores.length; i++) {  
    System.out.println(scores[i])  
}
```



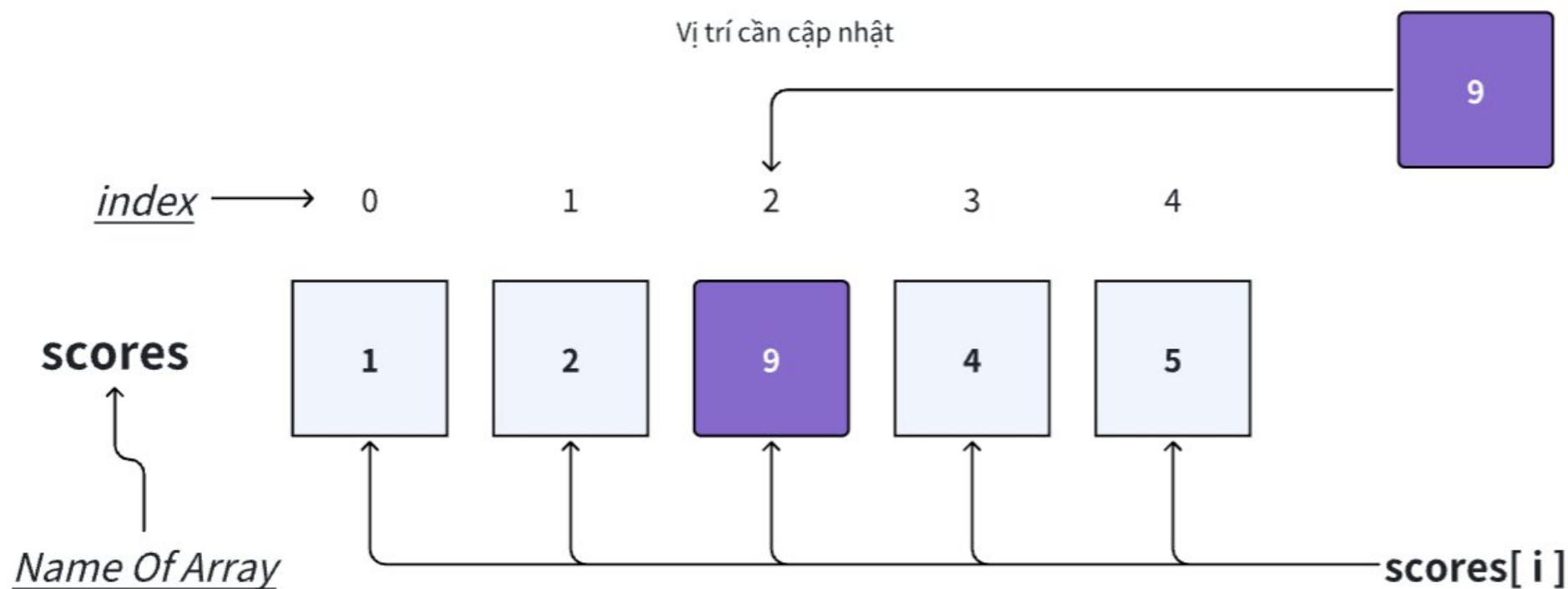
## 2. Các thao tác làm việc với mảng 1 chiều

### Thao tác thêm mới điểm cho sinh viên



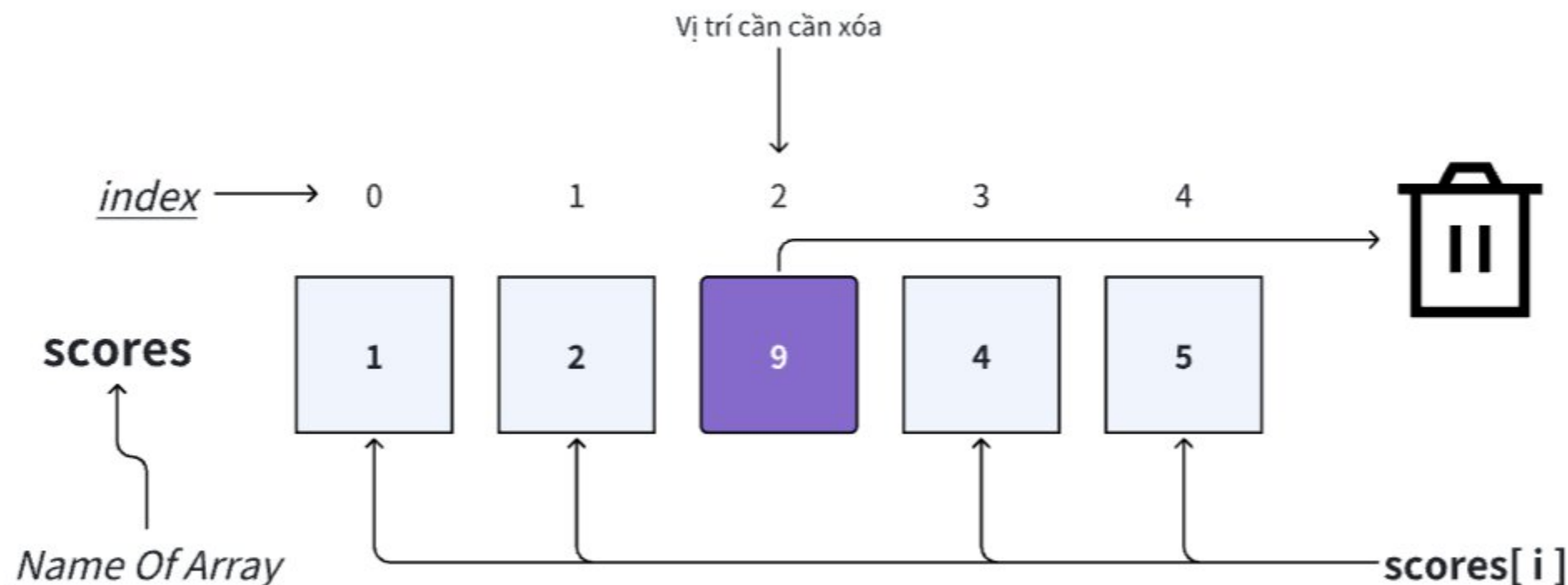
## 2. Các thao tác làm việc với mảng 1 chiều

### Thao tác cập nhật điểm cho sinh viên



## 2. Các thao tác làm việc với mảng 1 chiều

### Thao tác xóa điểm cho sinh viên

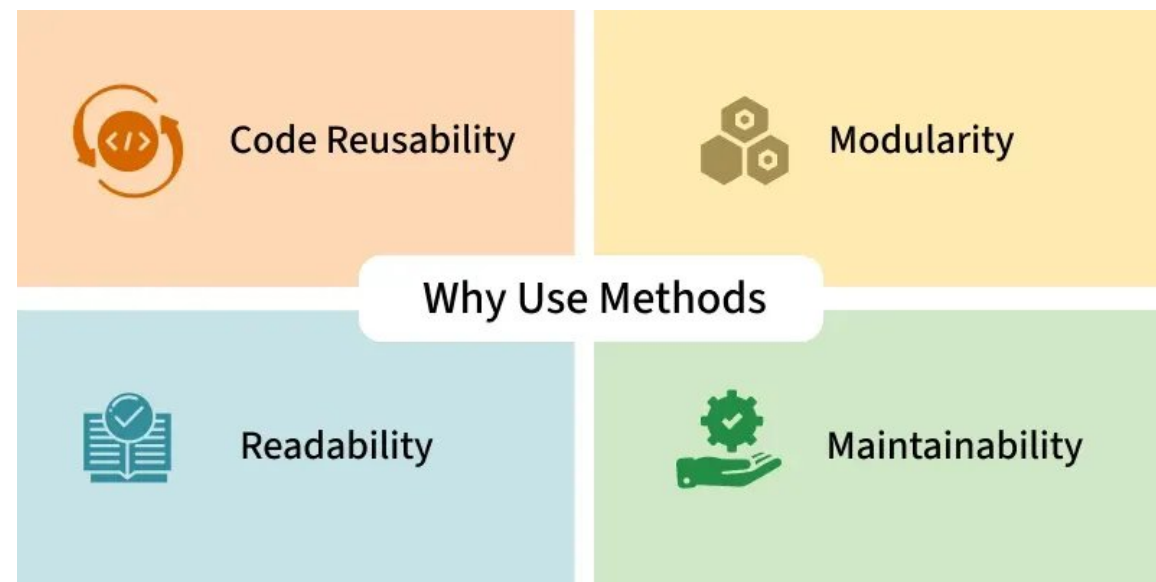


### 3. Sử dụng được cú pháp Java để thao tác với phương thức

Phương thức là một nhóm các câu lệnh dùng để thực hiện một nhiệm vụ nhất định

#### Mục đích:

- Giúp chương trình **gọn gàng**
- Tránh **lặp lại code**
- Dễ đọc, dễ bảo trì



### 3. Sử dụng được cú pháp Java để thao tác với phương thức

#### Những phương thức có sẵn trong Java

```
System.out.println('Nội dung cần in');  
// In nội dung ra màn hình  
Math.pow(2, 3);  
// Tính lũy thừa 2 mũ 3 → 8  
Math.sqrt(25);  
// Tính căn bậc 2 của 25 → 5  
// ...
```

### 3. Sử dụng cú pháp Java để thao tác với phương thức



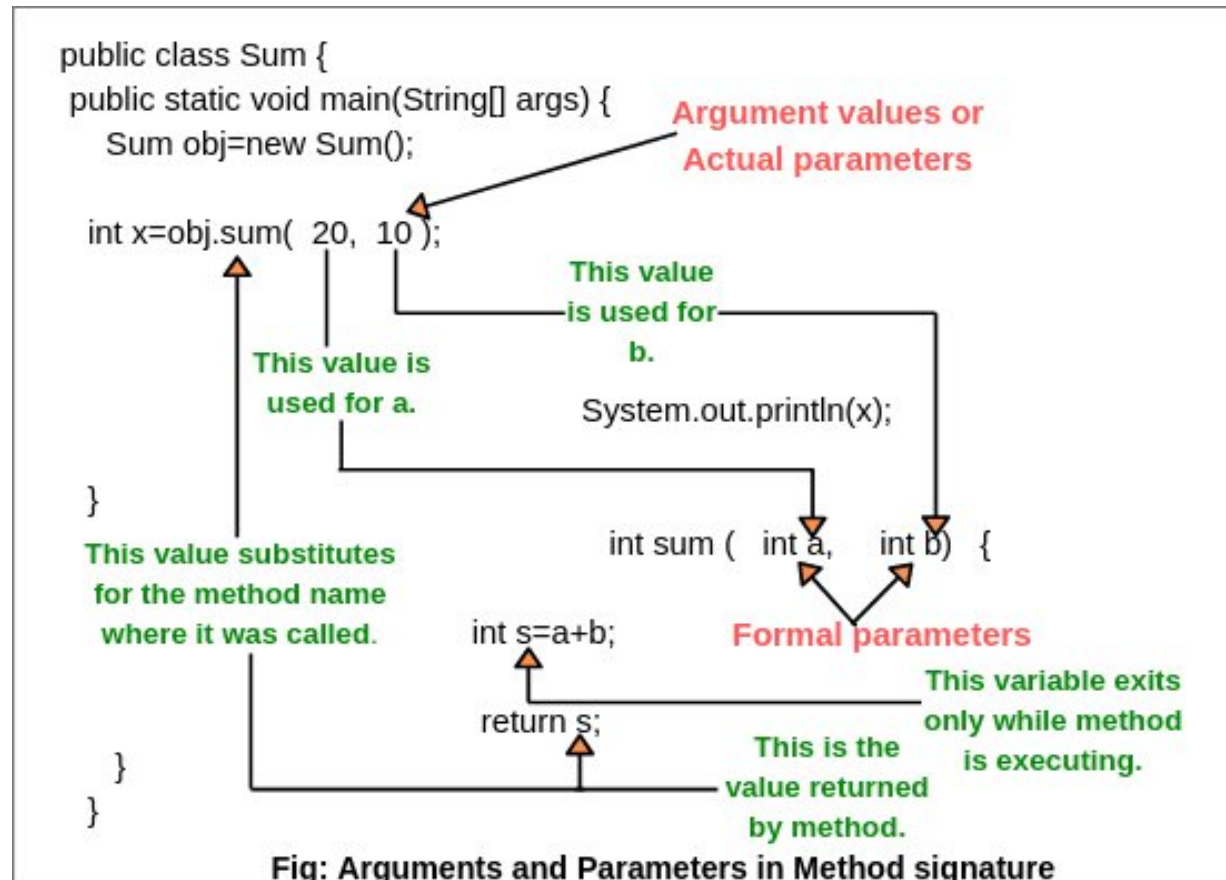
```
[modifier] [returnTypeValue] methodName( parameter ){  
    // Method Body  
}
```

#### Trong đó:

- **modifier**: phạm vi truy cập (ví dụ: **public**)
- **returnValueType**: kiểu dữ liệu trả về (**int**, **double**, **void**, ...)
- **methodName**: tên phương thức
- **parameter**: danh sách tham số (có thể rỗng)
- **method body**: các câu lệnh của phương thức

### 3. Sử dụng cú pháp Java để thao tác với phương thức

#### Tham số (parameters) và Đối số (Arguments)



### 3. Sử dụng được cú pháp Java để thao tác với phương thức

#### Gọi phương thức trong Java

```
// Defination Method
[modifier] [returnTypeValue] methodName( parameter ){
    // Method Body
}

// Call Method
methodName( parameter );
```



### 3. Sử dụng được cú pháp Java để thao tác với phương thức

#### Phương thức main() trong Java

```
public static void main(String[] args) {  
    // code bắt đầu chạy từ đây  
}
```

#### Trong đó:

- **main()** là phương thức đặc biệt
- Chương trình Java bắt đầu chạy từ main
- Từ **main()**, ta gọi các phương thức khác

### 3. Sử dụng được cú pháp Java để thao tác với phương thức

#### Ví dụ minh họa

```
public class Demo {  
  
    public static int sum(int a, int b) {  
        return a + b;  
    }  
  
    public static void main(String[] args) {  
        int result = sum(2, 3);  
        System.out.println(result);  
    }  
}
```

## 4. Các kỹ thuật lập trình với mảng



### Bài toán:

Cho danh sách điểm của một học sinh hãy thực hiện:

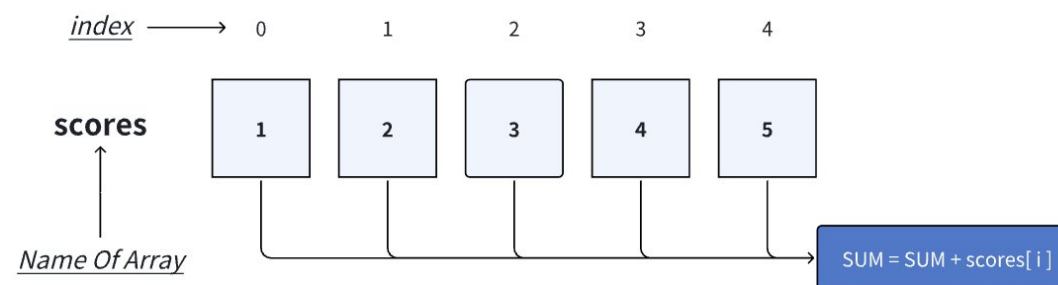
- Tính tổng điểm của sinh viên
- Đếm xem có bao nhiêu môn qua bao
- Kiểm tra xem có điểm tuyệt đối hay không

## 4. Các kỹ thuật lập trình với mảng

Kỹ thuật tính tổng các phần tử trong mảng:

**Ý tưởng:**

- Khởi tạo biến **sum** = 0
- Duyệt mảng
- Cộng từng phần tử vào **sum**



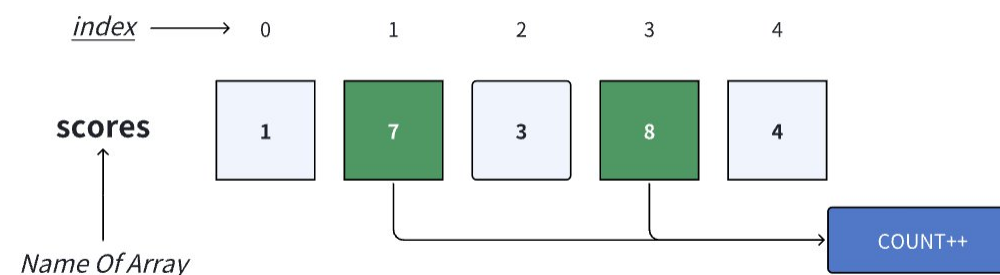
```
int sum = 0;
for (int i = 0; i < arr.length; i++) {
    sum += scores[i];
}
System.out.println("Sum = " + sum);
```

## 4. Các kỹ thuật lập trình với mảng

Kỹ thuật đếm các phần tử trong mảng:

**Ý tưởng:**

- Khởi tạo biến **count = 0**
- Duyệt mảng
- Nếu phần tử thỏa điều kiện → tăng **count**



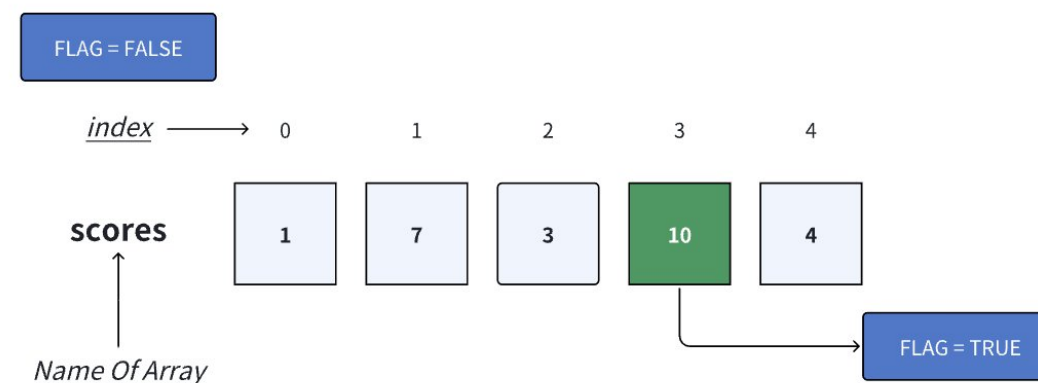
```
int count = 0;
for (int i = 0; i < scores.length; i++) {
    if (scores[i] >= 5) {
        count++;
    }
}
System.out.println("Số môn học đã qua môn = " + count);
```

## 4. Các kỹ thuật lập trình với mảng

Kỹ thuật cảm biến cò trong mảng:

Ý tưởng:

- Khởi tạo biến **flag = false**
- Duyệt mảng
- Nếu phần tử thỏa điều kiện → tăng **flag = true**



```

boolean found = false;
for (int i = 0; i < scores.length; i++) {
    if (scores[i] == 10) {
        found = true;
        break;
    }
}

if (found) {
    System.out.println("Found");
} else {
    System.out.println("Not found");
}

```

- ❑ **Nắm kiến thức về cú pháp và khai báo mảng 1 chiều**
- ❑ **Hiểu được cách thao tác cơ bản với mảng 1 chiều**
- ❑ **Hiểu và nắm được khai báo phương thức trong Java**
- ❑ **Nắm được các kỹ thuật lập trình với mảng**



# KẾT THÚC

HỌC VIỆN ĐÀO TẠO LẬP TRÌNH CHẤT LƯỢNG NHẬT BẢN