**Chuẩn bị:**

1. **Mảng là gì? Vì sao phải có mảng?**

**Mảng** là một cấu trúc dữ liệu đại diện cho một tập hợp các loại dữ liệu giống nhau.

Khi một mảng được tạo, kích thước của nó là cố định. Biến tham chiếu mảng được sử dụng để truy cập các phần tử trong mảng bằng cách sử dụng chỉ mục.

Mảng được sử dụng để lưu trữ một tập hợp dữ liệu, nhưng chúng ta thường thấy hữu ích hơn khi coi mảng là một tập hợp các biến cùng loại. Thay vì khai báo các biến riêng lẻ như number0, number1, ... . và number99, bạn khai báo 1 biến mảng như number và sử dụng number[0], number[1], . . . và number[99] để biểu thị các biến riêng lẻ. Phần này giới thiệu cách khai báo biến mảng, tạo mảng và xử lý mảng bằng chỉ mục.

**Phải có mảng vì**: Mảng giúp giải quyết nhiều vấn đề trong lập trình, bao gồm:

* Lưu trữ nhiều dữ liệu cùng loại: Mảng cho phép bạn lưu trữ nhiều giá trị có cùng kiểu dữ liệu trong một biến.
* Dễ dàng truy cập dữ liệu: Mảng sử dụng chỉ số để truy cập các phần tử, giúp bạn dễ dàng thao tác với dữ liệu.
* Hiệu quả về mặt bộ nhớ: Mảng lưu trữ các phần tử liên tiếp trong bộ nhớ, nên tiết kiệm bộ nhớ so với việc sử dụng nhiều biến riêng lẻ.

1. **Các cách khai báo, khởi tạo mảng.**

**Cách khai báo mảng:** Để sử dụng mảng trong chương trình, bạn phải khai báo một biến để tham chiếu mảng và chỉ định loại phần tử của mảng. Cú pháp khai báo một biến mảng:

elementType[] arrayRefVar;

hoặc:

elementType arrayRefVar[]; // Được phép, nhưng không được ưu tiên

ElementType có thể là bất kỳ kiểu dữ liệu nào và tất cả các phần tử trong mảng sẽ có cùng kiểu dữ liệu.

Tạo mảng Không giống như việc khai báo các biến kiểu dữ liệu nguyên thủy, việc khai báo một biến mảng không phân bổ bất kỳ khoảng trống nào trong bộ nhớ cho mảng. Nó chỉ tạo ra một vị trí lưu trữ để tham chiếu đến một mảng. Nếu một biến không chứa tham chiếu đến một mảng thì giá trị của biến đó là null. Bạn không thể gán các phần tử cho một mảng trừ khi nó đã được tạo trước đó. Sau khi khai báo một biến mảng, bạn có thể tạo một mảng bằng cách sử dụng toán tử mới và gán tham chiếu của nó cho biến đó theo cú pháp sau:

arrayRefVar = new elementType[arraySize];

Câu lệnh này thực hiện hai việc: (1) nó tạo một mảng bằng cách sử dụng new elementType[arraySize] và (2) nó gán tham chiếu của mảng mới được tạo cho biến arrayRefVar.

Khai báo một biến mảng, tạo một mảng và gán tham chiếu của mảng cho biến đó có thể được kết hợp trong một câu lệnh như

elementType[] arrayRefVar = new elementType[arraySize];

hoặc

elementType arrayRefVar[] = new elementType[arraySize];

**Khởi tạo mảng:** Java có một ký hiệu viết tắt, được gọi là trình khởi tạo mảng, kết hợp việc khai báo, tạo và khởi tạo một mảng trong một câu lệnh bằng cú pháp sau:

elementType[] arrayRefVar = {value0, value1, ..., valuek};

1. **Phương thức/ thao tác cơ bản với mảng.**

* Truy cập phần tử: Sử dụng chỉ số (index) để truy cập giá trị trong mảng, ví dụ: int x = numbers[2]; (lấy giá trị thứ ba của mảng numbers).
* Cập nhật phần tử: Sử dụng chỉ số để thay đổi giá trị của một phần tử, ví dụ: numbers[1] = 10; (cập nhật phần tử thứ hai thành 10).
* Tìm kích thước mảng: Sử dụng thuộc tính length, ví dụ: int size = numbers.length; (lấy kích thước của mảng numbers).
* Duyệt mảng: Sử dụng vòng lặp để duyệt qua từng phần tử của mảng.

1. **Mảng đa chiều là gì? Cách khai báo, sử dụng mảng đa chiều.**

**Mảng đa chiều** là một mảng chứa mảng khác. Nó được sử dụng để lưu trữ dữ liệu có cấu trúc hơn, chẳng hạn như ma trận hoặc bảng.

**Cách khai báo, sử dụng mảng đa chiều**

// Khai báo mảng refvar

dataType[][] refVar;

// Tạo mảng và gán tham chiếu của nó cho biến

refVar = new dataType[10][10];

// Kết hợp khai báo và tạo trong một câu lệnh

dataType[] [] refVar = new dataType[10] [10];

// Cú pháp thay thế

dataType refVar[] [] = new dataType[10] [10];