

 Cập nhật tháng 8 năm 2024

[Bài đọc] Làm việc với mảng đơn chiều

1. Khái niệm về mảng

- Mảng (Array) là một tập hợp các phần tử cùng kiểu dữ liệu, được lưu trữ liên tiếp trong bộ nhớ và có thể truy cập thông qua chỉ số (index). Trong Java, chỉ số của mảng bắt đầu từ 0.

2. Khai báo và khởi tạo mảng

- Có hai cách phổ biến để khai báo và khởi tạo mảng trong Java:
- Cách 1: Khai báo trước, khởi tạo sau

```
int[] numbers;           // khai báo mảng
numbers = new int[5];    // khởi tạo mảng với 5 phần tử
```

- Cách 2: Khai báo và khởi tạo cùng lúc

```
int[] numbers = new int[5];           // tất cả phần tử có giá trị mặc định là 0
String[] names = {"An", "Bình", "Cường"}; // khởi tạo mảng với giá trị sẵn có
```

3. Truy cập phần tử trong mảng

- Dùng chỉ số để truy cập:

```
int[] numbers = {1, 2, 3, 4, 5, 6};
System.out.println(numbers[0]); // in phần tử đầu tiên
numbers[2] = 10;                // gán giá trị 10 cho phần tử thứ 3
```

4. Duyệt mảng bằng vòng lặp

- Sử dụng vòng lặp for:

```
int[] numbers = {1, 2, 3, 4, 5, 6};

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

- Sử dụng vòng lặp for-each:

```
int[] numbers = {1, 2, 3, 4, 5, 6};

for (int number : numbers)
{
    System.out.println(number);
}
```

5. Một số thao tác với mảng thường gặp với mảng

- Tính tổng các phần tử:

```
int[] numbers = {1, 2, 3, 4, 5, 6};

int sum = 0;
for (int number : numbers)
{
    sum += number;
}
```

- Tìm phần tử lớn nhất:

```
int[] numbers = {1, 2, 3, 4, 5, 6};

int max = numbers[0];
for (int i = 1; i < numbers.length; i++) {
    if (numbers[i] > max) {
        max = numbers[i];
    }
}
```

- Nhập dữ liệu cho mảng từ bàn phím:

```
int[] numbers = new int[5];

Scanner scanner = new Scanner(System.in);
int[] arr = new int[5];
for (int i = 0; i < arr.length; i++)
{
    System.out.print("Nhập phần tử thứ " + (i + 1) + ": ");
    arr[i] = scanner.nextInt();
}
```

6. Thao tác với chức năng Thêm, sửa, xóa

- Giới hạn của mảng trong Java
 - Mảng trong Java có kích thước cố định sau khi được khởi tạo
 - Do đó, không thể thêm hoặc xóa phần tử trực tiếp như danh sách (List)
 - Tuy nhiên, bạn có thể giả lập thao tác thêm/xóa bằng cách tạo mảng mới
- Sửa phần tử trong mảng
 - Đơn giản là gán lại giá trị tại chỉ số cần sửa:

```
int[] arr = {1, 2, 3, 4, 5};
arr[2] = 99; // sửa phần tử thứ 3 (index 2) thành 99
```

- Thêm phần tử vào mảng (giả lập)
 - Do mảng không tự co giãn, ta cần tạo mảng mới có kích thước lớn hơn, rồi sao chép dữ liệu cũ và thêm phần tử mới

- Thêm phần tử vào cuối:

```
int[] oldArr = {1, 2, 3};
int newElement = 4;

int[] newArr = new int[oldArr.length + 1];
for (int i = 0; i < oldArr.length; i++) {
    newArr[i] = oldArr[i];
}
newArr[newArr.length - 1] = newElement;
```

- Thêm phần tử vào vị trí bất kỳ:

```
int[] oldArr = {1, 2, 3};
int newElement = 99;
int position = 1; // thêm vào vị trí thứ 2 (index = 1)

int[] newArr = new int[oldArr.length + 1];
for (int i = 0; i < position; i++) {
    newArr[i] = oldArr[i];
}
newArr[position] = newElement;
for (int i = position; i < oldArr.length; i++) {
    newArr[i + 1] = oldArr[i];
}
```

- Xóa phần tử trong mảng (giả lập)
 - Tạo mảng mới **ít hơn 1 phần tử**, sao chép trừ phần tử muốn xóa

```
int[] oldArr = {1, 2, 3, 4};  
int deleteIndex = 2; // xóa phần tử thứ 3 (index = 2)  
  
int[] newArr = new int[oldArr.length - 1];  
for (int i = 0, j = 0; i < oldArr.length; i++) {  
    if (i != deleteIndex) {  
        newArr[j++] = oldArr[i];  
    }  
}
```

Danh sách các bài học

