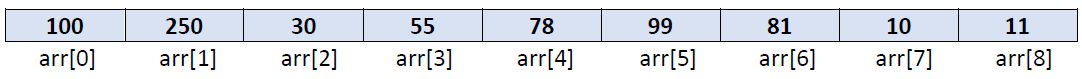
**Định nghĩa**

* Mảng là cấu trúc dữ liệu dùng để lưu trữ tập hợp của các giá trị cùng kiểu.
* Các giá trị đơn lẻ trong mảng được gọi là phần tử của mảng, nói gọn là phần tử hoặc phần tử mảng.
* Mỗi phần tử mảng được truy xuất, tức gán hoặc đọc giá trị ra thông qua một giá trị nguyên gọi là chỉ số mảng.
* Chỉ số mảng là số nguyên. Chỉ số phần tử đầu tiên trong mảng là 0. Chỉ số phần tử cuối trong mảng là n-1 với n là số phần tử tối đa có thể chứa của mảng.
* Như vậy:
  + Kiểu mảng là kiểu tập hợp, ta có thể dùng mảng để lưu một tập hợp các giá trị cùng kiểu nào đó
  + Thay vì dùng n biến lưu n giá trị khác nhau của cùng 1 kiểu nào đó ta dùng 1 biến mảng lưu trữ n phần tử
* Ví dụ: mảng dùng để lưu trữ danh sách sinh viên, danh sách môn học, danh sách các tỉ phú thế giới, danh sách người yêu cũ…
* Hình ảnh về mảng kiểu int tên arr. Dòng trên là từng giá trị của phần tử mảng, dòng dưới là cú pháp truy cập phần tử tương ứng:

[](https://braniumacademy.net/wp-content/uploads/2020/08/example-array.png)

**Cú pháp khai báo tổng quát**

Sau đây là cú pháp tổng quát của khai báo mảng:

type[] name;

Trong đó:

* Type là kiểu dữ liệu của mảng, thường gọi tắt là kiểu của mảng. Kiểu của mảng là các kiểu hợp lệ trong ngôn ngữ lập trình Java như các kiểu nguyên thủy: int, float boolean; các kiểu tham chiếu: String, các class, interface…
* Sau kiểu của mảng ta dùng cặp ngoặc vuông [] đi liền. Đây là cú pháp để nhận biết một biến kiểu mảng với một biến thông thường. Bạn có thể để [] sau tên mảng tuy nhiên khuyến nghị đặt [] ngay sau kiểu của mảng.
* Sau cặp móc vuông là 1 dấu cách, tiếp đó đến phần name. Name chính là tên mảng. Quy ước đặt tên như đặt tên biến nhưng ở dạng số nhiều.
* Kết thúc cú pháp khai báo mảng luôn có dấu chấm phẩy
* Ví dụ:

int[] myNumbers; *// khai báo biến mảng kiểu int*

float[] avgGrades; *// khai báo mảng kiểu float*

String[] brands; *// khai báo mảng kiểu String*

String[] students; *// khai báo mảng lưu danh sách các sinh viên*

String[] friends; *// khai báo mảng lưu danh sách những người bạn*

**Khởi tạo mảng**

Mảng khai báo như trên chưa sử dụng được do chúng ta chỉ mới khai báo, chưa khởi tạo giá trị cho nó. Mọi mảng trước khi đem sử dụng đều phải được khởi tạo hoặc gán giá trị.

* Cú pháp khởi tạo mảng:

type[] name = new type[num\_of\_elements];

type[] name = new type[]{elements};

type[] name ={elements};

* Trong đó giống cú pháp khai báo mảng và cộng thêm:
  + Dấu bằng = và vế bên phải dấu = là tổ hợp new …
  + Phần num\_of\_elements là số lượng phần tử của mảng mà ta muốn lưu trữ. Nhớ đặt trong ngoặc vuông. Số phần tử tối đa luôn là số nguyên dương.
  + Cách thứ nhất sẽ mặc định tạo ra một mảng kiểu type có num\_of\_elements phần tử. Tất cả các phần tử này mặc định sẽ là 0 nếu là các kiểu số, false nếu là boolean và null nếu là các kiểu tham chiếu như String, các class, interface khác
  + Cách thứ hai tạo ra một mảng kiểu type có số phần tử xác định qua số lượng phần tử được khởi tạo trong cặp {}.
  + Cách thứ ba là rút gọn của cách thứ 2, lúc này số lượng phần tử liệt kê trong cặp ngoặc {} sẽ là số lượng phần tử của mảng.
  + Các phần tử trong phần elements là các giá trị cùng kiểu với kiểu của mảng và phân tách nhau bởi dấu phẩy.
  + Kết thúc cú pháp khởi tạo luôn có dấu chấm phẩy.
* Ví dụ:

int[] myNumbers = new int[10] {1, 2, 3, 5}; *// error*

float[] avgGrades = {1.25f, 3.24, 5.66f, 7.49}; *// error*

String[] brands = new String[]{"Apple", "Samsung", "Huawei"}; *// ok*

String[] students = {"Loan", "Hoa", "Nhung", "Phương", "Thanh"}; *// ok*

**Truy xuất giá trị trong mảng**

* Để sử dụng giá trị trong mảng, tức đọc giá trị ra để tính toán, thao tác; hoặc gán, thay đổi giá trị một phần tử nào đó, ta sử dụng tên mảng và chỉ số mảng.
* Cú pháp tổng quát gán giá trị cho 1 phần tử là: **name[index] = value;**
* Cú pháp gán giá trị của một phần tử mảng cho một biến cùng kiểu: **variable = arrName[index];**
  + Trong đó name hoặc arrName là tên mảng
  + Value, variable phải cùng kiểu với kiểu của mảng
  + Index là chỉ số mảng luôn nằm trong đoạn [0, n-1] với n là số phần tử của mảng
* Cú pháp lấy số phần tử của mảng là: **name.length**
* Ta có thể gán hai biến mảng cùng kiểu cho nhau. Khi đó chúng cùng tham chiếu tới một mảng. Lý do là kiểu mảng là kiểu đối tượng.
* Ví dụ:

String[] brands = new String[]{"Apple", "Samsung", "Huawei", "Oppo"};

int size = brands.length; *// lấy số phần tử mảng*

System.*out*.println("Số phần tử của mảng là: " + size);

*// gán giá trị cho một phần tử mảng:*

brands[2] = "Xiaomi";

System.*out*.println("Phần tử tại chỉ số 2 của mảng là: " + brands[2]);

*// gán mảng cho một biến mảng khác cùng kiểu*

String[] other = brands; *// gán mảng brands cho mảng khác cùng kiểu String*

System.*out*.println(brands[0]); *// hiển thị giá trị phần tử đầu tiên trong brands*

System.*out*.println(other[0]); *// hiển thị giá trị phần tử đầu tiên trong other*

*// kết quả cùng là chuỗi kí tự "Apple"*

Kết quả:

Số phần tử của mảng là: 4

Phần tử tại chỉ số 2 của mảng là: Xiaomi

Apple

Apple

**Copy mảng**

* Nếu muốn copy nội dung từ một mảng sang 1 mảng khác ta dùng *Arrays*.*copyOf(source, new\_size);*
* Trong đó:
  + Tham số thứ nhất là mảng nguồn cần copy
  + Tham số thứ hai là kích thước mảng đích
  + Phương thức trên trả về mảng mới cùng kiểu mảng gốc trong tham số thứ nhất
* Ví dụ:

String[] brands = new String[]{"Apple", "Samsung", "Huawei", "Oppo"};

var copyArr = Arrays.*copyOf*(brands, 2);

System.*out*.println(copyArr[1]); *// kết quả in ra: Samsung*

**Truy xuất mảng**

* Truy xuất tất cả hoặc một đoạn phần tử trong mảng, ta thường sử dụng kết hợp với vòng lặp for:

String[] brands = new String[]{"Apple", "Samsung", "Huawei", "Oppo"};

*// duyệt toàn bộ mảng*

for (int i = 0; i < brands.length; i++) {

    System.*out*.println(brands[i]);

}

Apple

Samsung

Huawei

Oppo

* Lưu ý rằng nếu chỉ số của phần tử mảng mà nằm ngoài đoạn [0, length-1] với length là độ dài hay số phần tử mảng. Thì bạn sẽ nhận được một ngoại lệ:

public class Bai22 {

    public static void main(String[] args) {

        String[] brands = new String[]{"Apple", "Samsung", "Huawei", "Oppo"};

*// duyệt toàn bộ mảng*

for (int i = 0; i <= brands.length; i++) {

**System.*out*.println(brands[i]); // dòng xuất hiện ngoại lệ**

        }

    }

}

Apple

Samsung

Huawei

Oppo

Exception in thread "main" **java.lang.ArrayIndexOutOfBoundsException:** Index 4 out of bounds for length 4

               at Other.main(Other.java:6)

Do đó luôn đảm bảo rằng sử dụng chỉ số mảng đúng trong đoạn [0, length-1].

**Vòng lặp foreach**

* Khi sử dụng vòng lặp for, ta có thể gặp vấn đề đó là chỉ số phần tử vượt biên của mảng. Điều này làm chương trình bị crash và kết thúc không mong muốn.
* Để giải quyết vấn đề trên, ta sử dụng vòng lặp foreach.
* Đọc là for-each.
* Vòng lặp này sẽ lần lượt duyệt từ đầu đến hết tập hợp hiện có.
* Cú pháp của vòng lặp foreach là:

for (type element : collection) {

*// do something*

}

* Trong đó:
  + Type là kiểu của biến, cùng kiểu với kiểu mảng cần duyệt, có thể sử dụng var
  + Element là tên biến nào đó ngắn gọn dùng để đại diện cho từng phần tử của mảng
  + Có dấu hai chấm phân tách ở giữa element và collection
  + Phần collection là tên một tập hợp nào đó như tên mảng chẳng hạn
  + Trong phần thân của foreach là các câu lệnh cần thực hiện nào đó
* Ta sử dụng foreach khi duyệt mảng là chủ yếu. Còn nếu muốn thao tác sâu hơn như thay đổi giá trị phần tử, duyệt một đoạn thay vì toàn bộ mảng,… ta dùng vòng for thường.
* Để sử dụng foreach trong IntelliJ bạn gõ foreach rồi ấn phím Tab.
* Ví dụ:

String[] brands = new String[]{"Apple", "Samsung", "Huawei", "Oppo"};

*// duyệt toàn bộ mảng*

for (int i = 0; i < brands.length; i++) {

    System.*out*.println(brands[i]); *// hiển thị p.tử thứ i*

}

*// tương đương:*

for (var item : brands) { *// sử dụng kiểu tự suy luận var*

System.*out*.println(item);

}

*// hoặc:*

for (String item : brands) { *// sử dụng kiểu tường minh: String*

System.*out*.println(item);

}

Kết quả:

Apple

Samsung

Huawei

Oppo

* Foreach không làm thay đổi giá trị của phần tử mảng:

String[] brands = new String[]{"Apple", "Samsung", "Huawei", "Oppo"};

*// duyệt toàn bộ mảng*

for (var item : brands) { *// sử dụng kiểu tự suy luận var*

item = item + item.toString();

}

*// duyệt lại xem có thay đổi gì không*

for (String item : brands) { *// sử dụng kiểu tường minh: String*

System.*out*.println(item);

}

Apple

Samsung

Huawei

Oppo

* Vòng lặp thường thì có:

String[] brands = new String[]{"Apple", "Samsung", "Huawei", "Oppo"};

*// duyệt toàn bộ mảng*

for (int i = 0; i < brands.length; i++) {

    brands[i] = brands[i] + brands[i];

}

*// duyệt lại xem có thay đổi gì không*

for (String item : brands) { *// sử dụng kiểu tường minh: String*

System.*out*.println(item);

}

AppleApple

SamsungSamsung

HuaweiHuawei

OppoOppo

**Ví dụ minh họa**

Ví dụ 1: Nhập vào danh sách n > 0 người bạn của bạn, lưu vào mảng. Tìm người tên dài nhất và hiển thị ra màn hình.

import java.util.Scanner;

public class Vidu1 {

    public static void main(String[] args) {

        var input = new Scanner(System.*in*);

        System.*out*.println("Nhập số người bạn của bạn: ");

        var size = input.nextInt();

        var friends = *getFriends*(size, input);

        var maxLength = *findMaxLength*(friends);

*// hiện kết quả:*

*showFriendList*(friends);

*showResult*(friends, maxLength);

    }

*// phương thức hiển thị danh sách những người bạn*

private static void showFriendList(String[] friends) {

        System.*out*.println("Danh sách những người bạn: ");

        for (var friend : friends) {

            System.*out*.print(friend + " ");

        }

    }

*// phương thức hiển thị danh sách những người có tên dài nhất*

private static void showResult(String[] friends, int maxLength) {

        System.*out*.println("\nDanh sách những người có tên dài nhất là: ");

        for (var friend : friends) {

            if (friend.length() == maxLength) {

                System.*out*.print(friend + " ");

            }

        }

    }

*// phương thức tìm giá trị độ dài lớn nhất của tên*

private static int findMaxLength(String[] friends) {

        var max = 0;

        for (var friend : friends) {

            if (friend.length() > max) { *// tìm tên có độ dài max*

max = friend.length();

            }

        }

        return max;

    }

*// phương thức nhập vào một danh sách những người bạn và trả về ds này*

private static String[] getFriends(int size, Scanner input) {

        var friends = new String[size];

        for (int i = 0; i < size; i++) {

            System.*out*.println("Nhập vào tên người bạn thứ " + (i + 1) + ": ");

            friends[i] = input.next(); *// chỉ nhập tên*

}

        return friends;

    }

}

Nhập số người bạn của bạn:

6

Nhập vào tên người bạn thứ 1:

Nam

Nhập vào tên người bạn thứ 2:

Nhân

Nhập vào tên người bạn thứ 3:

Hoa

Nhập vào tên người bạn thứ 4:

Nhung

Nhập vào tên người bạn thứ 5:

Phương

Nhập vào tên người bạn thứ 6:

Linh

Danh sách những người bạn:

Nam Nhân Hoa Nhung Phương Linh

Danh sách những người có tên dài nhất là:

Phương

Process finished with exit code 0

Ví dụ 2: Cho mảng nguyên n > 0 phần tử nhập vào từ bàn phím. Liệt kê các số nguyên tố trong mảng này.

import java.util.Scanner;

public class Vidu2 {

    public static void main(String[] args) {

        var input = new Scanner(System.*in*);

        System.*out*.println("Nhập số phần tử mảng > 0: ");

        var n = input.nextInt();

        if (n > 0) {

*// tạo mảng chứa dữ liệu nhập vào:*

var numbers = *getNumbers*(n, input);

*// hiển thị kết quả:*

*showRessult*(numbers);

        } else {

            System.*out*.println("Vui lòng nhập số phần tử mảng > 0");

        }

    }

*// phương thức hiển thị kết quả*

private static void showRessult(int[] numbers) {

        System.*out*.println("Các số nguyên tố trong mảng: ");

        for (int i = 0; i < numbers.length; i++) {

            if (*isPrime*(numbers[i])) { *// nếu số tại vị trí i là số ng.tố*

System.*out*.print(numbers[i] + " "); *// in nó ra*

}

        }

    }

*// phương thức nhập vào danh sách các số nguyên cần kiểm tra*

private static int[] getNumbers(int n, Scanner input) {

        var numbers = new int[n];

        for (int i = 0; i < n; i++) {

            System.*out*.print("Nhập phần tử thứ " + (i + 1) + ": ");

            numbers[i] = input.nextInt();

        }

        return numbers;

    }

*// phương thức kiểm tra n có phải số nguyên tố hay không*

private static boolean isPrime(int n) {

        if (n < 2) {

            return false;

        }

        for (int i = 2; i <= Math.*sqrt*(n); i++) {

            if (n % i == 0) {

                return false;

            }

        }

        return true;

    }

}

Nhập số phần tử mảng > 0:

10

Nhập phần tử thứ 1: 1

Nhập phần tử thứ 2: 2

Nhập phần tử thứ 3: 5

Nhập phần tử thứ 4: 4

Nhập phần tử thứ 5: 8

Nhập phần tử thứ 6: 11

Nhập phần tử thứ 7: 7

Nhập phần tử thứ 8: 9

Nhập phần tử thứ 9: 21

Nhập phần tử thứ 10: 101

Các số nguyên tố trong mảng:

2 5 11 7 101

Process finished with exit code 0