**Mảng (Array) trong Java**

**1. Khai Báo, Khởi Tạo và Truy Cập Phần Tử Mảng**

Mảng là cấu trúc dữ liệu lưu trữ nhiều phần tử có cùng kiểu dữ liệu.

**Khai Báo Mảng**

Khai báo mảng tạo ra một tham chiếu tới mảng, nhưng chưa khởi tạo bộ nhớ cho các phần tử.

int[] arr; // Khai báo mảng số nguyên

**Khởi Tạo Mảng**

Sau khi khai báo, cần phải khởi tạo mảng với kích thước cố định.

arr = new int[5]; // Khởi tạo mảng với 5 phần tử

Hoặc khai báo và khởi tạo đồng thời:

int[] arr = new int[5]; // Khai báo và khởi tạo mảng có 5 phần tử

Khởi tạo với giá trị cố định:

int[] arr = {1, 2, 3, 4, 5}; // Mảng chứa 5 phần tử

**Truy Cập Phần Tử Mảng**

Các phần tử mảng được truy cập qua chỉ số (index), bắt đầu từ 0.

int firstElement = arr[0]; // Truy cập phần tử đầu tiên

arr[1] = 10; // Gán giá trị 10 cho phần tử thứ hai

**2. Mảng Một Chiều**

Mảng một chiều chứa một danh sách các phần tử, tất cả có cùng kiểu dữ liệu.

int[] arr = {1, 2, 3, 4, 5}; // Mảng số nguyên một chiều

System.out.println(arr[2]); // In ra phần tử thứ 3 (giá trị là 3)

**3. Mảng Nhiều Chiều**

Mảng nhiều chiều lưu trữ các mảng trong một mảng khác. Phổ biến nhất là mảng hai chiều.

**Mảng Hai Chiều**

Mảng hai chiều thường được coi như là một ma trận.

int[][] matrix = {

{1, 2, 3},

{4, 5, 6},

{7, 8, 9}

};

Truy cập phần tử trong mảng hai chiều:

int value = matrix[1][2]; // Truy cập phần tử hàng 1, cột 2 (giá trị là 6)

**Chuỗi (String) trong Java**

**1. Lớp String**

Lớp String được sử dụng để lưu trữ chuỗi các ký tự. Chuỗi là bất biến (immutable), nghĩa là một khi được tạo, không thể thay đổi giá trị của nó.

**Khai Báo Chuỗi**

Có thể khai báo chuỗi trực tiếp bằng cách sử dụng dấu ngoặc kép hoặc từ khóa new.

String str1 = "Hello";

String str2 = new String("World");

**2. Các Phương Thức Chuỗi**

**length()**

Trả về độ dài (số ký tự) của chuỗi.

String str = "Hello";

int length = str.length(); // length = 5

**charAt()**

Trả về ký tự tại vị trí chỉ định trong chuỗi (chỉ số bắt đầu từ 0).

char ch = str.charAt(1); // ch = 'e'

**substring()**

Trả về chuỗi con từ chuỗi ban đầu. Có hai phiên bản:

* substring(int beginIndex) - Trả về chuỗi con từ vị trí beginIndex.
* substring(int beginIndex, int endIndex) - Trả về chuỗi con từ beginIndex đến endIndex (không bao gồm endIndex).

String sub1 = str.substring(1); // sub1 = "ello"

String sub2 = str.substring(1, 4); // sub2 = "ell"

**indexOf()**

Tìm vị trí xuất hiện đầu tiên của ký tự hoặc chuỗi con trong chuỗi. Nếu không tìm thấy, trả về -1.

int index1 = str.indexOf('e'); // index1 = 1

int index2 = str.indexOf("lo"); // index2 = 3

**3. Lớp StringBuilder và StringBuffer**

Cả hai lớp StringBuilder và StringBuffer đều cho phép tạo chuỗi có thể thay đổi (mutable). Khác với String, các đối tượng của hai lớp này có thể thay đổi nội dung mà không tạo ra đối tượng mới trong bộ nhớ.

**StringBuilder**

Lớp này được sử dụng khi không cần đồng bộ hóa (synchronized). Nó nhanh hơn StringBuffer.

StringBuilder sb = new StringBuilder("Hello");

sb.append(" World"); // Thêm chuỗi " World" vào cuối chuỗi ban đầu

System.out.println(sb.toString()); // In ra "Hello World"

**StringBuffer**

Tương tự như StringBuilder, nhưng StringBuffer được đồng bộ hóa, an toàn hơn trong các ứng dụng đa luồng.

StringBuffer sb = new StringBuffer("Hello");

sb.append(" World");

System.out.println(sb.toString()); // In ra "Hello World"

Cả StringBuilder và StringBuffer đều hỗ trợ các phương thức như:

* append(): Thêm chuỗi vào cuối chuỗi hiện tại.
* insert(): Chèn chuỗi vào vị trí chỉ định.
* delete(): Xóa các ký tự từ vị trí chỉ định.

Trên đây là các kiến thức cơ bản về mảng, chuỗi và các lớp hỗ trợ trong Java.