**BIẾN MẢNG VÀ ĐỐI TƯỢNG MẢNG**

Đối tượng mảng(array object):Là một đối tượng thực sự trong bộ nhớ,được cấp phát bộ nhớ động khi khởi tạo mảng bằng cú pháp new.Nó chứa dữ liệu và các phương thức để truy cập và thao tác với dữ liệu đó.  
  
Biến mảng(array variable):Là một biến tham chiếu(reference variable)đến đối tượng mảng trong bộ nhớ.Biến mảng chỉ chứa địa chỉ bộ nhớ của đối tượng mảng,không chứa dữ liệu thực tế của mảng đó.Biến mảng được sử dụng để tham chiếu đến đối tượng mảng và thực hiện các thao tác.

VD: int arr = new int[10];  
arr là biến mảng chỉ lưu địa chỉ vùng nhớ của đối tượng mảng mà không lưu trữ dữ liệu  
new int[10] là đối tượng mảng có vùng nhớ chưa 10 giá trị trên đối tượng mảng đó.

**Array**

- Biến đặc biệt, lưu trữ được nhiều giá trị;  
 - Các giá trị cùng kiểu dữ liệu, khác nhau về chỉ số;  
 - Tên mảng, các phần tử, chỉ số, độ dài;

**Khai báo mảng**

elementType[] arrayRefVar;  
elementType: kiểu dữ liệu các phần tử trong mảng;  
arrayRefVar: tên mảng;

- Khi khai báo biến mảng thì sẽ không có việc cấp phát bộ nhớ ngay cho các phần tử của mảng. Chỉ có việc cấp phát bộ nhớ cho tham chiếu đến mảng.  
 - Nếu không gắn tham chiếu đến mảng thì giá trị của biến mảng là null

Ví dụ: khai báo mảng số thực myList:  
 double[] myList;  
myList là 1 biến chứa tham chiếu đến 1 mảng  
Tuy nhiên tường gọi myList là 1 mảng

**Khởi tạo mảng**

arrayRefVar = new elementType[arraySize];

**Khai báo, khởi tạo và gắn tham chiếu cho biến mảng**

elementType[] arrayRefVar = new elementType[arraySize];  
hoặc  
elementType arrayRefVar[] = new elementType[arraySize];

**Gán giá trị cho các phần tử mảng**

arrayRefVar[index] = value;  
Ví dụ: khai báo String gồm 10 phần tử:  
str là biến mảng  
new String[10] là đối tượng mảng  
String[] str = new String[10];  
int[] array = new int[5];  
array[0] = 0;  
array[1] = 1;  
array[2] = 2;  
array[3] = 3;  
array[4] = 5;

**Khởi tạo nhanh mảng**

String[] str = {"h","e","l","l","o"};  
int[] nums = {1,2,3,4};

Cách khác  
double[] arr;  
int size = 5;  
arr = new double[size];

**Nhập liệu cho mảng**

System.out.println("Input data for array");  
for (int i = 0; i < arr.length; i++){  
 System.out.printf("arr[%d] = ", i);  
 arr[i] = scanner.nextDouble();  
}

**Hiển thị dữ liệu mảng**

System.out.println("Show data of array:");  
for (double i: arr){ for-each: Dùng trong trường hợp không thay đổi dữ liệu mảng  
System.out.println(i);

}

**Sao chép mảng**

int[] a = {1,2,3,5,4};

int[] b = new int[a.length];

System.arraycopy(a, 0, b, 0, a.length);

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

System.out.printf("b[%d] - %d \n", i, b[i]);

**String Format trong Java**

%s : strings (Định dạng chuỗi)

String str = "Java programming language";

System.out.printf("Tôi đang học ngôn ngữ lập trình %s", str);

Output: Tôi đang học ngôn ngữ lập trình Java programming language

%d : decimal integers (Định dạng số nguyên)

int age = 25;

System.out.printf("Tuổi của tôi là %d", age);

Output: Tuổi của tôi là 25

%f : floating-point numbers (Định dạng số thực)

double price = 10.50;

System.out.printf("Giá của sản phẩm là: %.3f", price);

Output: Giá của sản phẩm là: 10.500

%b : booleans (true/false)

boolean isJavaFun = true;

System.out.printf("Có phải Java là ngôn ngữ lập trình thú vị? %b", isJavaFun);

Output: Có phải Java là ngôn ngữ lập trình thú vị? true

%c : characters (Định dạng ký tự, đặt trong dấu nháy đơn)

char myGrade = 'A';

System.out.printf("Điểm số của tôi là: %c", myGrade);

Output: Điểm số của tôi là: A

%tT : date/time values. (Định dạng thời gian thực, đầy đủ giờ, phút, giây)

import java.util.Date;

Date now = new Date();

System.out.printf("Bây giờ là: %tT", now);

Output: Bây giờ là: 14:26:42

%n : line separator (xuống dòng)

System.out.printf("Chào mừng đến với Java.%n");

System.out.printf("Đây là ví dụ về sử dụng để thêm dòng mới");

Output: Chào mừng đến với Java.

Đây là ví dụ về sử dụng để thêm dòng mới.

**Chống trôi lệnh trong Java**

Xảy ra khi chúng ta nhập vào trước 1 số rồi sau đó nhập vào 1 chuỗi thì lúc này có hiện tượng trôi lệnh.

Giải thích: Khi nhập vào một số trước và chúng ta nhấn Enter thì ở dòng sau dòng nhập số phần nhập chuỗi sẽ bị thế bởi lệnh gõ phím Enter, từ đó gây ra trôi lệnh.

VD:

import java.util.Scanner;

Scanner scanner = new Scanner(System.in);

System.out.print("Nhập một số nguyên: ");

int myNumber = scanner.nextInt();

System.out.print("Nhập một chuỗi: "); //Trôi lệnh bắt đầu xuất hiện ở đây

String myString = scanner.nextLine();

**Khắc phục:**

import java.util.Scanner;

Scanner scanner = new Scanner(System.in);

System.out.print("Nhập một số nguyên: ");

int myNumber = scanner.nextInt();

**scanner.nextLine();**  // Chống trôi lệnh

System.out.print("Nhập một chuỗi: ");

String myString = scanner.nextLine();

**hoặc:**

import java.util.Scanner;

Scanner scanner = new Scanner(System.in);

System.out.print("Nhập một số nguyên: ");

**int myNumber = Integer.parseInt(scanner.nextLine());** // Chống trôi lệnh

System.out.print("Nhập một chuỗi: ");

String myString = scanner.nextLine();

**Wrapper class**

Chúng ta đã được biết về các kiểu dữ liệu nguyên thuỷ trong java như int, long, float etc. Các kiểu dữ liệu nguyên thuỷ này không có tác dụng gì ngoài lưu trữ các giá trị, nếu chúng ta muốn chuyện một số int sang String, hoặc là khởi tại một số nguyên bởi một chuỗi số thì điều này là không thể.

Vì thế java đã cung cấp cho chúng ta Wrapper class tương ứng với từng kiểu dữ liệu. Các Wrapper class này chứa các kiểu tương ứng đồng thời cung cấp cho chúng ta vô số method hữu ích.

Wrapper class gồm:(kiểu nguyên thủy - kiểu đối tượng)

boolean - Boolean

char - Character

byte - Byte

short - Short

int - Integer

long - Long

float - Float

double - Double

Ngoài ra, autoboxing sẽ tự động chuyển từ kiểu nguyên thủy thành kiểu đối tượng và ngược lại unboxing chuyển từ kiểu đối tượng sang kiểu nguyên thủy.

VD về việc sử dụng Wrapper class:

Integer x = new Integer(10); // đóng gói một số nguyên vào đối tượng Integer

Integer y = new Integer(5);

System.out.println("x = " + x.toString()); // in giá trị của x

System.out.println("y = " + y.toString()); // in giá trị của y

Integer sum = x + y; // thực hiện phép cộng hai đối tượng Integer

System.out.println("Tổng của x và y = " + sum.toString()); // in giá trị tổng của x và y