Bài 10

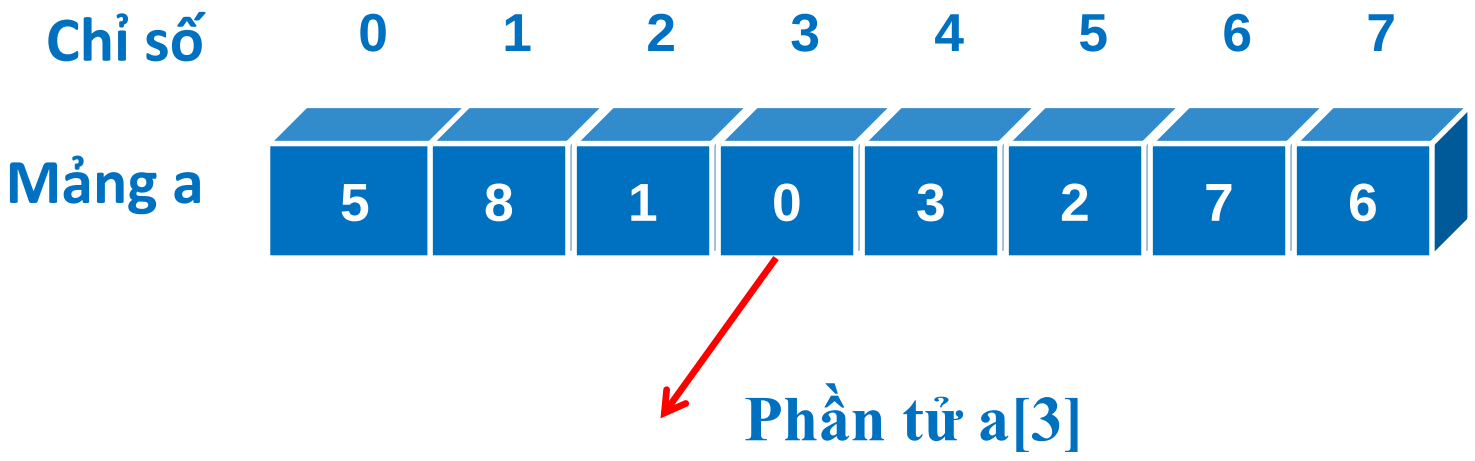
Làm việc với Array

Nội dung bài học

- Khái niệm
- Khai báo, cấp phát vùng nhớ
- Thao tác với mảng
- Hạn chế của mảng

Khái niệm

- Mảng là một **cấu trúc dữ liệu**, lưu trữ một **tập các phần tử** cùng kiểu dữ liệu và có kích thước cố định: **fixed-size sequential**.
- Thay vì khai báo individual variables var0, var1, …, var99. Ta có thể khai báo mảng có với các phần tử var[0], var[1], …, var[99].
- Mỗi phần tử trong mảng được truy xuất thông qua **chỉ số, giá trị**



Khai báo

❖ Khai báo: Array Variables

```
data_type arr[];     // works but not preferred way.
data_type[] arr;   // preferred way.

double elements[];          // works but not preferred way.
String[] elements;          // preferred way.
Employee[] employees;  // preferred way.
```

Khai báo

❖ Khởi tạo: Array Variables

```
dataType[] arrayRefVar = new dataType[arraySize];
```

Cú pháp:

dataType[] arrayRefVar = **new** dataType[arraySize];

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

Example:

String[] elements = new String[4];

Khái niệm

- Lợi ích của việc sử dụng mảng
  - Cho phép quản lý nhiều phần tử có cùng kiểu dữ liệu tại một thời điểm
  - Lưu trữ các phần tử liên tiếp nhau trong ô nhớ, tránh phân mãnh ổ đĩa.
  - Tạo ra sự tối ưu trong việc quản lý bộ nhớ so với việc sử dụng nhiều biến cùng kiểu dữ liệu
  - **Mảng là kiểu dữ liệu đối tượng.**
  - Bộ nhớ[HEAP] được cấp phép cho mảng chỉ khi mảng thực sự được sử dụng. Do đó, bộ nhớ không bị tiêu tốn ngay khi khai báo mảng .

Khái niệm

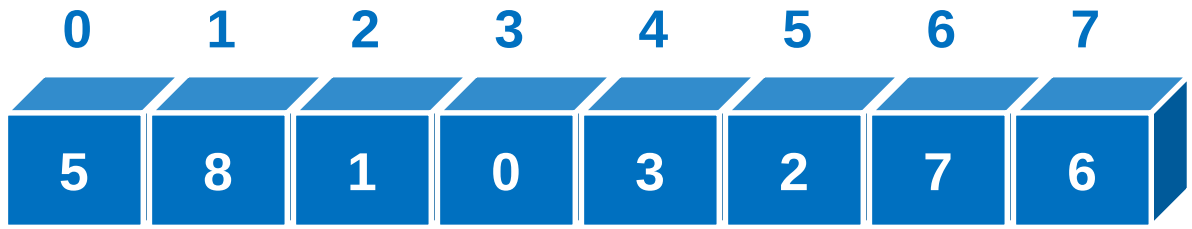
❖ Tên class của KDL mảng tương ứng

Array type	Corresponding class Name
int[]	[I
int[][]	[[I
double[]	[D
double[][]	[[D
byte[]	[B
boolean[]	[Z
Student[]	[package_name.Student

Thao tác với mảng

Thao tác: Arrays Variables

- Truy xuất giá trị các phần tử:
  - Phần tử đầu tiên có chỉ số là 0
  - Phần tử cuối cùng có chỉ số là length – 1
  - Truy cập ngoài phạm vi: ArrayIndexOutOfBoundException



Thao tác với mảng

❖ Duyệt mảng: Array Variables

```
for (int i = 0; i < TênMảng.length; i++){
    // Xử lý trên phần tử TênMảng[i]
}
```

Example: for-index

```
int[] a = {1,2,3,4,5};
for (int i = 0; i < a.length; i++){
    System.out.println(a[i]);
}
```

Thao tác với mảng

❖ **Duyệt mảng: Array Variables**

```
for (KDL tenbien: tenmang){
    // Xử lý trên phần tử tenbien
}
```

Example: for-each – ignore index

```
int[] array = {1,2,3,4,5};
for (int number: array){
    System.out.println(number);
}
```

Array

Đề bài:

Cho mảng số nguyên int[] numbers

Yêu cầu:

- In ra các phần tử lẻ trong mảng
- Tính tổng các phần tử chẵn trong mảng
- Gán giá trị cho phần tử đầu tiên = 199
- Tìm phần tử có giá trị lớn nhất trong mảng
- Tìm phần tử có giá trị lớn thứ hai trong mảng
- Sắp xếp mảng theo chiều tăng dần
- Thêm mới phần tử vào vị trí 0, k, n-1
- Xóa phần tử tại vị trí 0, k, n - 1

Mảng nhiều chiều

- Mảng nhiều chiều là mảng một chiều mà **mỗi phần tử của mảng** chứa **một mảng khác**, nói cách khác mỗi phần tử của mảng chứa 1 tập hợp nhiều phần tử bên trong
- Các phần tử này vẫn được truy cập thông thường thông qua các chỉ số.
- Cách khai báo:

  <Kiểu dữ liệu> <tên mảng>[][]….[];

  Hoặc

  <Kiểu dữ liệu>[][]…[] <tên mảng>;

Hạn chế

- Kích thước và số chiều cố định, khó khăn trong việc mở rộng ứng dụng.
- Các phần tử được đặt và tham chiếu một cách liên tiếp nhau trong bộ nhớ, khó khăn trong việc thao tác với các phần tử nằm ở giữa mảng.
- Khó khăn trong việc chèn, xóa một phần tử bất kì trong mảng

Tóm tắt bài học

❖ Trong những ngôn ngữ lập trình (ngay cả bán đối tượng C++) mảng là một kiểu dữ liệu cơ bản. Trong Java mảng là một kiểu dữ liệu đối tượng, phải dùng toán tử **new** để tạo mảng.

❖ Nên đặt notation [] phía sau kiểu dữ liệu: String[] arrRefName …

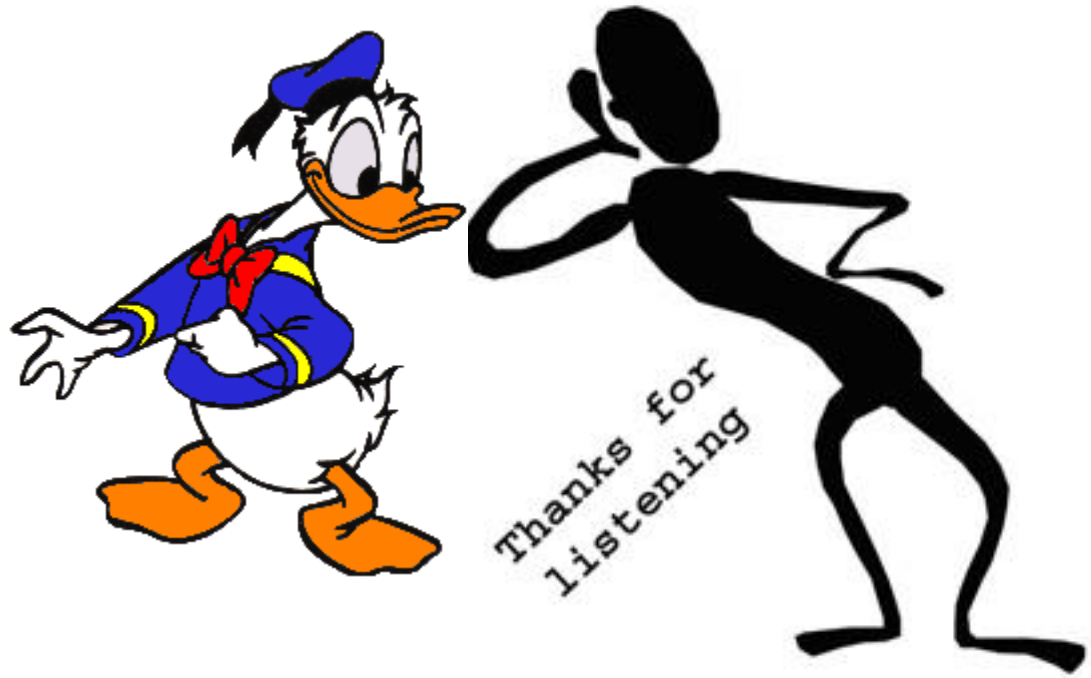
❖ Mảng có trường length để tìm kích thước của mảng theo đơn vị byte. Nhớ rằng đây là tổng số byte bị chiếm trong mảng chứ không phải là số các mục dữ liệu có trong mảng

Thư viện Arrays

❖ The java.util.Arrays class bao gồm một số phương thức static hỗ trợ sorting, searching, comparing và filling elements trong array.

❖ These methods are overloaded for all primitive types.

Method	Desciption
public static int binarySearch(Object[] a, Object key)	Tìm kiếm vị trí của phần tử có giá trị key trong mảng a. Sử dụng binary search algorithm. The array **must be sorted prior** to making this call. Return index of search key. Otherwise return -1
public static boolean equals(long[] a, long[] a2)	Return true nếu 2 mảng có cùng number of elements và all corresponding pairs of elements in the two arrays are equal
public static void fill(int[] a, int val)	Assigns the specified int value to each element of the specified array of ints
public static void sort(Object[] a)	Sorts the specified array of objects into an ascending order, using quick sort algorithm



END