BÀI GIẢNG

NGÔN NGỮ LẬP TRÌNH JAVA



MÅNG TRONG JAVA

TS. Nguyễn Sĩ Thìn (nsthin@vku.udn.vn)



Chương 4. Mảng trong Java



Mục tiêu

- Hiểu và sử dụng được mảng trong Java
- Hiểu và sử dụng được lớp ArrayList
- Hiểu và sử dụng được lớp Wrapper







Nội dung



Chương 4: Mảng trong Java

- 1. Mång
- 2. ArrayList
- 3. Wrapper



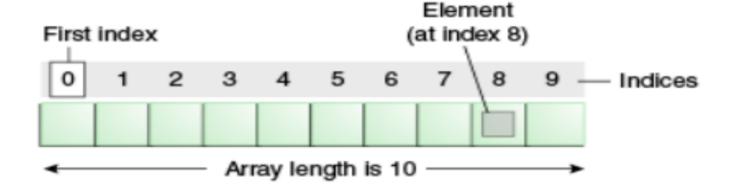
1. Khái niệm mảng



 Khái niệm: tập hợp các đối tượng có cùng kiểu dữ liệu và được lưu trữ gần nhau trong bộ nhớ.

Lưu ý:

- Có độ dài mảng (length)
- Các phần tử có thứ tự (index từ 0; a[index])
- · Các phần tử cùng kiểu dữ liệu
- Một chiều, đa chiều





1.1 Mảng một chiều



Cú pháp khai báo mảng:

- <kiểu dữ liệu>[] <tên mảng> → int[] a;
- <kiểu dữ liệu> []<tên mảng> → int []a;
- <kiểu dữ liệu> <tên mảng>[]
 → int a[];

Cấp phát mảng:

<tên mảng> = new <kiểu dữ liệu> [kích thước mảng]
 a=new int[5];

Khởi tạo mảng:

- a[0]=1; // tiếp tục khởi tạo cho a[1]=8, a[2]=6, a[3]=4, a[4]=9
- int[] a= {1,8,6,4,9};



Thao tác với các phần tử mảng



- Thao tác với các phần tử mảng
 - Vòng lặp for:
 for(int i=0;i<a.length;i++) {
 // làm việc với các phần tử mảng thông qua a[i]
 }
 - Vòng lặp for-each:
 for(int i: a) {
 // làm việc với các phần tử mảng thông qua a[i]
 }
 - Phương thức toString:
 System.out.println(Arrays.toString(a));



Các phép toán, phương thức sử dụng mảng



```
Gán giá trị cho mảng:
 public static void main(String[] args)
       int[] a= {1,8,6,4,9};
       int[] b=a;
       a[4]=6;
       System.out.println(b[4]); // result: 6
Lưu ý :
   - Mảng a và b cùng trỏ đến cùng một mảng

    Copy giá trị: int[] b=Arrays.copyOf(a, a.length);

                   a =
                   b =
                                                     4
                                                     9
```



Các phép toán, phương thức sử dụng mảng (tt)



	STT	java.util.Arrays	Ý nghĩa
	1	<pre>static String toString (type[] a)</pre>	Trả về chuỗi các phần tử của mảng a
	2	<pre>static type[] copyOf (type[] a)</pre>	Trả về một mảng giống mảng a
	3	<pre>Static type[] copyOfRange toString (type[] a, int start, int end)</pre>	Trả về mảng giống với mảng a nhưng chỉ tính từ vị trí "start" đến vị trí "end"
	4	<pre>static void sort (type[] a)</pre>	Sắp xếp lại mảng a
•	5	<pre>static int binarySearch (type[] a, type v)</pre>	Trả về vị trí của giá trị v trong mảng đã được sắp xếp a
	6	<pre>static int binarySearch (type[] a, int start, int end, type v)</pre>	Trả về vị trí của giá trị v trong mảng đã được sắp xếp a từ vị trí thứ "start" đến vị trí thứ "end -1"
	7	<pre>static void fill (type[] a, type v)</pre>	Thiết lập tất cả phần tử mảng a với giá trị v
	8	<pre>static boolean equals (type[] a, type[] b)</pre>	Trả về true nếu 2 mảng bằng nhau (cùng độ dài và các phần tử giống nhau)



1.2 Mảng đa chiều

Chỉ số dòng



- Mảng đa chiều: tăng số chiều lưu trữ lên nhiều chiều (>=2)
- o Lưu ý:
 - · Là mảng 1 chiều với các phần tử mảng
 - Mảng 2 chiều dùng 2 index để tham chiếu đến phần tử mảng

```
Côt 0
                                                         Côt 1
                                                                     Côt 2
                                                                                 Côt 3
                                                                  M[0][2]
                                                                              M[0][3]
                                         M[0][0]
                                                      M[0][1]
                                  Dòng 0
                                                                                  0
Mång a = {
             \{7, 2, 9, 0\},\
             {9, 5, 4, 1},
                                        M[1][0]
                                                                  M[1][2]
                                                                              M[1][3]
                                                      M[1][1]
                                  Dòng 1
             \{8, 0, 3, 6\},\
                                                          5
                                        M[2][0]
                                                                  M[2][2]
                                                                              M[2][3]
                                                       M[2][1]
                                  Dòng 2
                                                                      3
                                                                                  6
                                                                    Chỉ số cột
                                     Tên mảng
```



Khai báo cú pháp mảng đa chiều



- Cú pháp khai báo mảng: (ví dụ cho mảng 2 chiều)
 - <kiểu dữ liệu>[][] <tên mảng>
 <kiểu dữ liệu> [][]<tên mảng>
 <kiểu dữ liệu> <tên mảng>[][]
 → int [][]a;
 int a[][];
- Cấp phát mảng:
 - <tên mảng> = new <kiểu dữ liệu> [kích thước hàng] [kích thước cột] a=new int[3][5];
- Khởi tạo mảng:



Thao tác các phần tử mảng đa chiều



Thao tác các phần tử mảng

Phương thức deepToString:

System.out.println(Arrays.deepToString(a));



2. ArrayList



- Nhược điểm của mảng array: kích thước mảng là cố định
 - → Không bổ sung thêm hoặc xóa bớt các phần tử
 - → Lãng phí bộ nhớ
- Giải pháp: collections: Vector, ArrayList,...
 - → Mảng động, thêm bớt phần tử mềm dẻo
- Cú pháp:

ArrayList<kiểu đối tượng> <tên biến> = new ArrayList<>;

→ T: kiểu của các phần tử; KHÔNG được là kiểu nguyên thủy



Các phương thức trong ArrayList



STT	java.util.ArrayList <t></t>	Ý nghĩa
1	ArrayList <t>()</t>	Hàm dựng tạo một mảng array list rỗng chứa các phần tử kiểu "T"
2	ArrayList <t>(int d)</t>	Hàm dựng tạo một mảng array list với khả năng lưu trữ bộ nhớ ban đầu cho "d" phần tử thuộc kiểu "T"
3	boolean add (T obj)	Thêm một phần tử "obj" kiểu "T" vào cuối mảng array list
4	Void add (int index, T obj)	Chèn thêm phần tử "obj" vào vị trí "index" của mảng array list
5	<pre>int size ()</pre>	Trả về số phần tử đang có trong mảng array list
6	<pre>void set (int index, T obj)</pre>	Gán giá trị "obj" cho phần tử thứ "index" của mảng array list
7	T get (int index)	Trả về phần tử ở vị trí "index" của mảng
8	T remove (int index)	Xóa phần tử thứ "index" và dời tất cả các phần tử còn lại xuống 1 vị trí



3. Wrapper



- Đặt vấn đề: Làm sao tạo ra một đối tượng array list để lưu các phần tử số nguyên (interger)
 - → ArrayList<int> arr=new ArrayList<>; //error
 - → ArrayList<Integer> arr=new ArrayList<>(); //ok
- Các kiểu nguyên thủy và lớp Wrapper tương ứng

Primitive Type	Wrapper class
boolean	Boolean
char	Character
byte	Byte
short	Short
int	Integer
long	Long
float	Float
double	Double



Autoboxing và Unboxed



- Autoboxing: cơ chế chuyển đổi tự động kiểu nguyên thủy sang kiểu Wrapper tương ứng
 - Ví dụ:

```
int a=20;
```

Integer i=Integer.valueOf(a);// chuyển tường minh

Integer j=a;//autoboxing

- Unboxed: cơ chế chuyển đổi tự động kiểu Wrapper sang kiểu nguyên thủy tương ứng
 - Ví dụ:

```
Integer a=new Integer(3);
```

int i=a.intValue();//chuyển tường minh

int j=a;//unboxing



Bài thực hành cuối chương 4



ĐÈ BÀI

- Anh/Chị hãy viết chương trình quản lý sinh viên gồm các chức năng sau:
- Cho phép nhập vào danh sách sinh viên
- Hiển thị danh sách sinh viên
- Sắp xếp sinh viên theo điểm trung bình tăng dần
- Biết rằng lớp "Student" được kế thừa từ lớp cha "Person":
 - + Lớp "Person" gồm các thuộc tính: name, age
 - + Lớp "Student" kế thừa lớp Person gồm các thuộc tính bổ sung:
 - ID (Mã sinh viên), average (điểm trung bình)



THANK YOU!