

“HỌC ĐỂ LÀM VIỆC”

Java for Base

Contents

1

Array, ArrayList, List and Collection

2

Strings

3

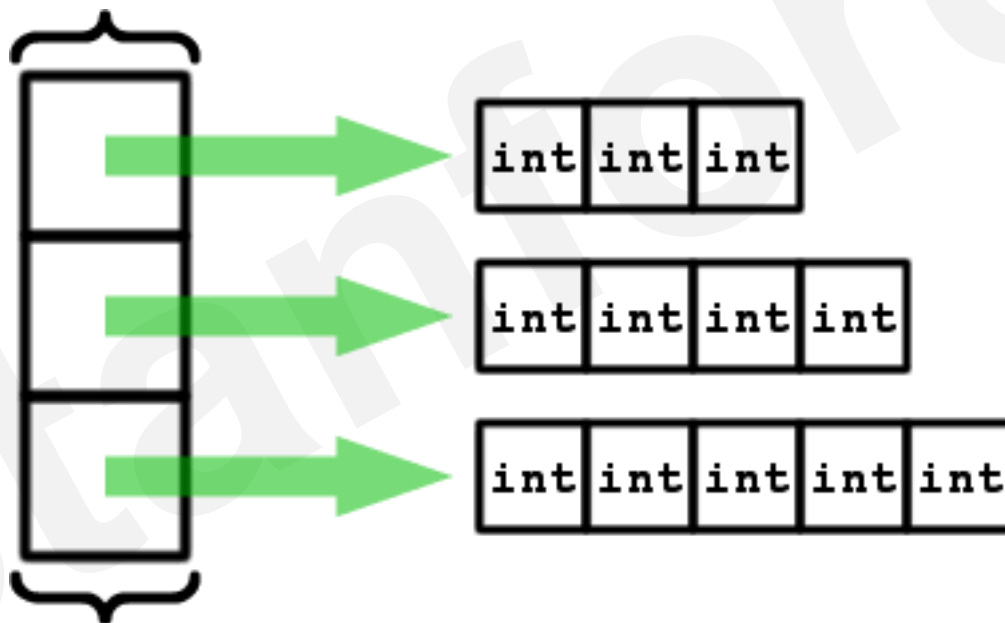
Exercises



Array, List and Collection

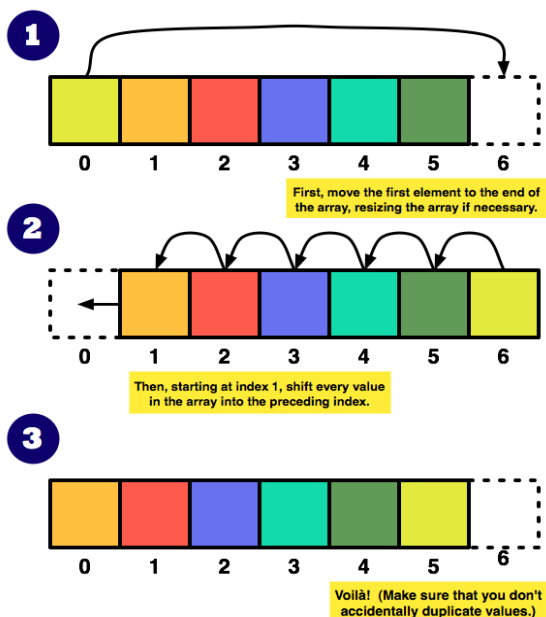
■ Array, List and Collection

- **Mảng (array):** là tập hợp các phần tử có cùng kiểu dữ liệu



■ Array, List and Collection

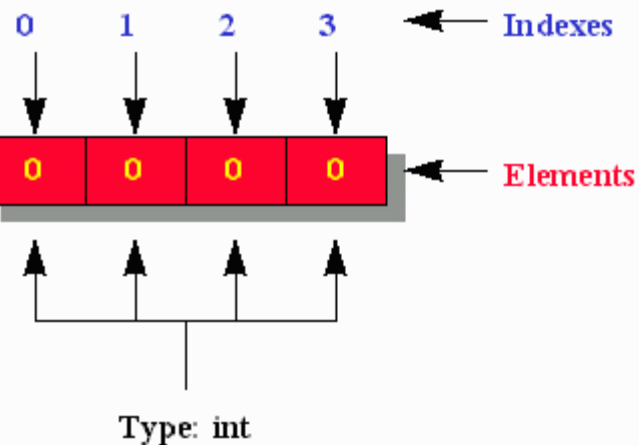
Ví dụ: Mảng kiểu số nguyên trong Java



One-dimensional
array variable

test_scores
(reference)

Type: int []



■ **Array, List and Collection**

- **Mảng 1 chiều**
- **Mảng nhiều chiều**
- **Mảng động ArrayList**
- **List**

■ Array, List and Collection

- Một số phương thức, thuộc tính trong mảng

Method	Description
length	Trả về số lượng phần tử có trong mảng
clone	Tạo và copy đối tượng hiện tại
equals	So sánh bằng với một đối tượng khác trả về true hoặc false
toString	Chuyển dữ liệu về một chuỗi

■ Array, List and Collection

■ Mảng 1 chiều:

- Mảng chứa nhiều phần tử có cùng kiểu dữ liệu
- Trong gói package của java là **java.util.Arrays**
- Cú pháp khai báo mảng:

<Kiểu dữ liệu> <Tên mảng>[] = new [KieuDuLieu][SoPhanTu];

Ví dụ: Khai báo một mảng số nguyên gồm 10 phần tử như sau:

```
//Khai báo 1 mảng gồm 10 phần tử
```

```
int arr[] = new int[]{10,40,30,20,50,60,70,90,80,100};
```

Hoặc `int arr[] = new int[10]; arr[0] = 2; arr[1] = 5;`

■ Array, List and Collection

■ Mảng 1 chiều:

- Ví dụ 1: Nhập số phần tử cho một mảng từ giao diện

```
//Khai báo đối tượng làm việc với input/output dữ liệu trong java  
Scanner input = new Scanner(System.in);
```

```
System.out.print("Nhập số phần tử:");  
int count = input.nextInt();
```

```
//Khai báo mảng  
int arrInfo[] = new int[count];
```

```
for(i=0; i<arrInfo.length; i++)  
{  
    System.out.printf("arrInfo[%s]= ", i);  
    arrInfo[i] = input.nextInt();  
}
```

```
//In thông tin ra giao diện  
for(i=0; i<arrInfo.length; i++)  
{  
    System.out.println(arrInfo[i]);  
}
```

■ Array, List and Collection

■ Mảng 1 chiều:

- Ví dụ 2: Khai báo Mảng là kiểu số nguyên

```
//Khai báo 1 mảng gồm 10 phần tử
int arr[] = new int[10]; //{10,40,30,20,50,60,70,90,80,100};

//Gán giá trị cho phần tử
arr[0] = 2;
arr[1] = 5;

//Duyệt các phần tử của mảng
System.out.println("Số phần tử của mảng là: ");
for(i=0; i<arr.length; i++)
{
    System.out.println(arr[i]);
}
```

■ Array, List and Collection

✓ Mảng nhiều chiều:

➤ Mảng 2 chiều:

■ Cú pháp:

- <Kiểu dữ liệu> <Tên Mảng> **[][]** = new <Kiểu dữ liệu>[m][n];
- m: Số dòng được tính theo hàng ngang của Mảng.
- n: Số cột được tính theo hàng dọc của Mảng.

Ví dụ: Khai báo mảng 2 chiều:

```
int iArray [ ][ ] = new int[2][3];
```

Chú ý: Nếu trường hợp n không khai báo thì mảng có m phần tử và mỗi phần tử có thể là một mảng

■ Array, List and Collection

✓ Mảng nhiều chiều:

➤ Mảng 2 chiều:

- Ví dụ 1: Khai báo Mảng 2 chiều, đưa giá trị vào mảng rồi in chúng ra ngoài màn hình.

```
String arrThongTin[][] = new String[2][2];

arrThongTin[0][0] = "Stanford";
arrThongTin[0][1] = "Day kinh nghiệm lập trình";
arrThongTin[1][0] = "Số 20 ngõ 100 Nguyễn Chí Thanh, HN";
arrThongTin[1][1] = "Website: www.stanford.com.vn";

//In thông tin ra giao diện
for(i=0; i<arrThongTin.length; i++)
{
    for(int j=0; j<arrThongTin[i].length; j++)
    {
        System.out.println(arrThongTin[i][j]);
    }
}
```

- **Array, List and Collection**

- ✓ **Mảng nhiều chiều:**

- **Mảng 2 chiều:**

- **Ví dụ 1:** Hiển thị các phần tử của mảng lên giao diện

```
//In thông tin ra giao diện
for(i=0; i<arrThongTin.length; i++)
{
    for(int j=0; j<arrThongTin.length; j++)
    {
        System.out.println(arrThongTin[i][j]);
    }
}
```

Stanford
Đay kinh nghiệm lập trình
Số 20 ngõ 100 Nguyễn Chí Thanh, HN
Website: www.stanford.com.vn

▪ Array, List and Collection

✓ Mảng nhiều chiều:

➤ Mảng n chiều:

- Tương tự như mảng 2 chiều. Cú pháp:
- `<Kiểu dữ liệu>[][]...[] <tên mảng> = new <kiểu dữ liệu>[][]...[];`

Ví dụ: Khai báo mảng nhiều chiều :

```
String[][][] strArray = new string[2,3,4];
```

■ Array, List and Collection

- **ArrayList:** Là một mảng động trong Java có thể thêm hoặc xóa phần tử trong mảng tùy theo mục đích sử dụng

Ví dụ:

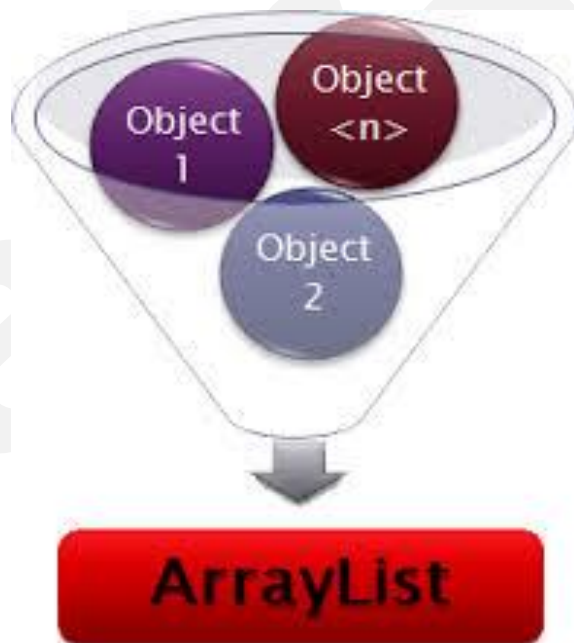
```
//Khởi tạo một mảng
ArrayList arrLst = new ArrayList();

//Gán giá trị cho mảng
arrLst.add("Nguyễn Hải Nam");
arrLst.add("Nam");
arrLst.add(30);

//Duyệt thông tin in ra màn hình
for(Object obj: arrLst)
{
    System.out.println(obj);
}
```


■ Array, List and Collection

- **List:** là một mảng động bao gồm một tập hợp các phần tử có cùng kiểu dữ liệu.



■ Array, List and Collection

■ List:

Cú pháp: `List<type> name = new ArrayList<type>();`

Sử dụng trong gói **`java.util.List`**

- **`add()`**: Thêm phần tử vào Lists
- **`remove()`**: Xóa phần tử từ Lists
- **`set()`**: Chèn phần tử vào List, Có thể chỉ định vị trí thêm vào. Có 2 tham số (index,value).
- **`clear()`**: Xóa các phần tử khỏi danh sách
- **`contains()`**: Tìm giá trị trong list, return true or false.

- **Array, List and Collection**

- **List:**

Ví dụ:

```
List<String> lstThongTin = new ArrayList<String>();

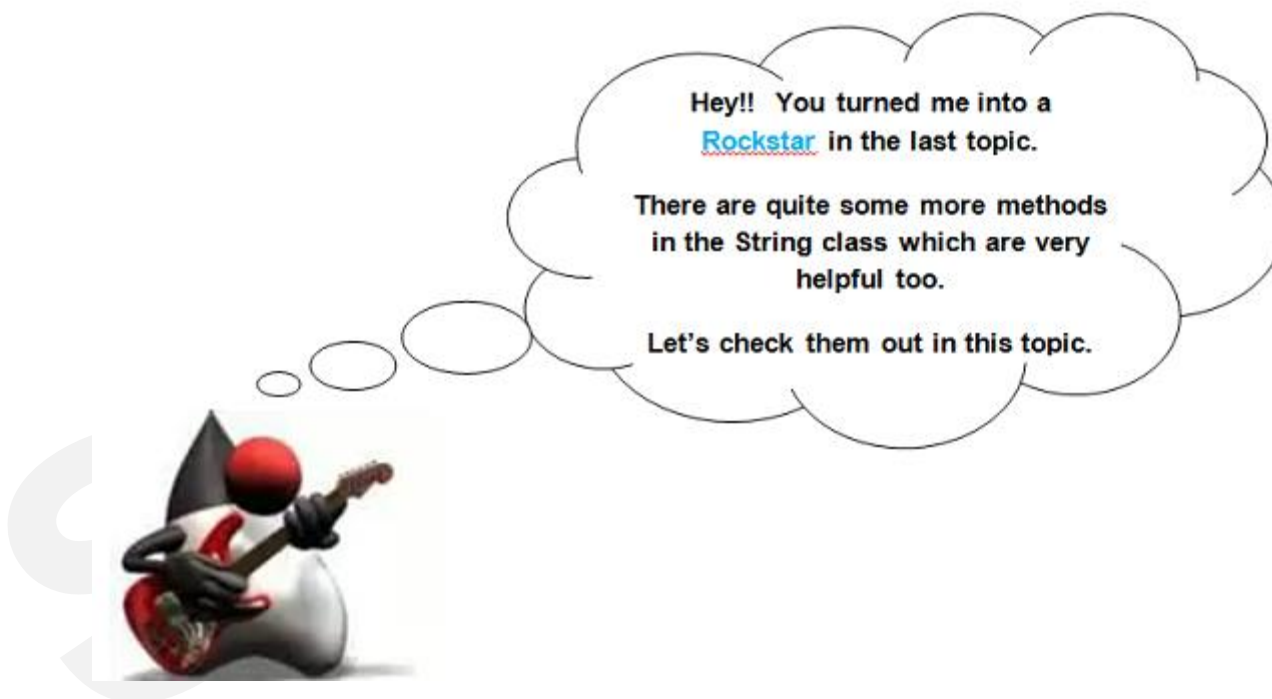
lstThongTin.add("Stanford");
lstThongTin.add("Dạy kinh nghiệm lập trình");

//Duyệt thông tin để in ra
for(Object obj: lstThongTin)
{
    System.out.println(obj);
}
```

String

■ String

- Sử dụng để làm việc với các dữ liệu dạng chuỗi, ký tự



■ String

○ Các phương thức hay sử dụng:

Phương thức	Mục đích
compareTo	So sánh với một chuỗi khác
concat	Thêm thông tin vào cuối chuỗi đã có
replace (char oldChar, char newChar)	Thay thế kí tự trong chuỗi
substring (int beginIndex, int endIndex)	Trả về một chuỗi con trong một chuỗi lớn
toLowerCase()	Chuyển chuỗi về dạng chữ thường
toUpperCase()	Chuyển chuỗi về dạng chữ hoa

■ String

○ Các phương thức hay sử dụng:

Phương thức	Mục đích
trim	Loại bỏ ký tự trắng khỏi chuỗi
indexOf	Trả về vị trí đầu tiên của một ký tự hoặc chuỗi cần tìm
lastIndexOf	Trả về vị trí cuối của một ký tự hoặc chuỗi cần tìm
contains	Kiểm tra xem chuỗi có chứa ký tự <code>String strCorp = "";</code> <code>Boolean b = strCorp.contains("S");</code>
isEmpty	Kiểm tra chuỗi ký tự có rỗng không
length	Lấy độ lớn của một chuỗi

■ String

○ Các phương thức hay sử dụng:

Phương thức	Mục đích
toString	Chuyển về một chuỗi
equals	So sánh bằng với một chuỗi
split	Trả về một mảng từ chuỗi được chia
toCharArray	Chuyển chuỗi về một mảng kí tự
startsWith	Kiểm tra xem chuỗi có bắt đầu bằng kí tự hoặc chuỗi so sánh không
endsWith	Kiểm tra xem chuỗi có kết thúc bằng kí tự hoặc chuỗi so sánh không

■ String

- **Ví dụ:** Cho một chuỗi thông tin như sau, thực hiện lấy từ Stanford trong chuỗi đã cho.

```
String strThongTin = "Stanford - Dạy kinh nghiệm lập trình\n";  
strThongTin += "Địa chỉ: Số 20 ngõ 678 Đường Láng, Đống Đa, HN\n";  
strThongTin += "Điện thoại: 04.6275 2212 - 04.6662 3355\n";  
strThongTin += "Website: www.stanford.com.vn";
```

■ String

■ StringBuilder

giúp thao tác nhanh các chuỗi với ít tổn hao bộ nhớ hơn so với String.

Ví dụ:

Cho một chuỗi “**Cong ty Stanford – Dao tạo và phát triển công nghệ**” thay thế các ký tự trong chuỗi này bằng ký tự đứng sau nó 2 vị trí.

=> Nếu dùng lệnh for và chuỗi để xử lý yêu cầu này sẽ tốn bộ nhớ vì mỗi lần thay đổi cần thêm một chuỗi mới trong khi đó StringBulder chỉ tạo ra một vùng để lưu trữ chuỗi ký tự này.

Exercises

“HỌC ĐỂ LÀM VIỆC”

Thank You !