Mảng và các thao tác trên mảng

Nội dung

1. Mảng
2. Chuỗi
3. Các lớp thao tác với Chuỗi

Mảng

1. Định nghĩa
2. Mảng một chiều
3. Mảng hai chiều

Mảng

Tại sao cần sử dụng mảng?

- cần lưu trữ nhiều giá trị trong 1 biến
- truy xuất, xử lý dễ dàng

Mảng

Định nghĩa

- kiểu dữ liệu đặc biệt
- lưu trữ nhiều giá trị có **cùng kiểu dữ liệu**
- sử dụng các ô nhớ liên tiếp

Mảng

Cấu trúc

- các phần tử trong mảng phải có cùng kiểu dữ liệu
- một biến mảng xác định bằng 1 tên
- mỗi phần tử sẽ được gắn với 1 chỉ số
- chỉ số mảng trong Java bắt đầu từ 0
- Kích thước của mảng là số lượng phần tử mảng có thể lưu trữ

Mảng

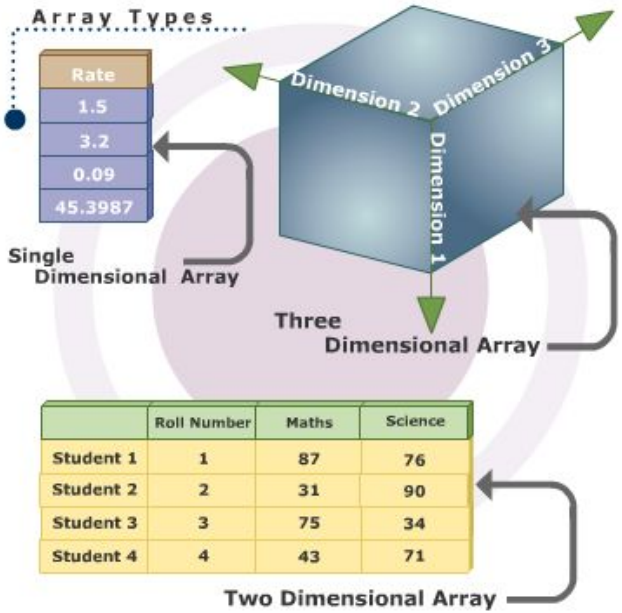
Mảng 1 chiều

- giống như bảng có nhiều hàng

  nhưng chỉ có 1 cột dữ liệu

Cú pháp

```
datatype[] arrayName = new datatype[size];
```



Mảng

Cú pháp

```
datatype[] arrayName = new datatype[size];
```

Ví dụ

```
int[] studentScores = new int[25];
```

Mảng

Lưu trữ giá trị vào mảng

- Lưu giá trị ngay từ khi khởi tạo
- Sử dụng vòng lặp để khởi tạo

Mảng

Lưu giá trị khi khởi tạo - 1

```
int[] studentScores = new int[5]
studentScores[0] = 45;
studentScores[1] = 65;
studentScores[2] = 73;
studentScores[3] = 25;
studentScores[4] = 89;
```

Mảng

Lưu giá trị khi khởi tạo - 2

```
int[] studentScores = {45, 75, 63, 48, 89};
```

Mảng

Sử dụng vòng lặp For để khởi tạo

```
float[] itemRate = new float[3];
float total = 0;
Scanner input = new Scanner(System.in);
for (int count = 0; count < 3; count++) {
    System.out.print(“Enter Item rate: ”);
    itemRate[count] = input.nextFloat();
    Total = total + itemRate[count];
}
```

Mảng

Hiển thị giá trị mảng - Duyệt mảng

- Sử dụng vòng lặp for thông thường
- Sử dụng vòng lặp for-each

Mảng

Sử dụng vòng lặp for thông thương

```
int[] studentScores = {45, 75, 63, 48, 89};
for (int i = 0; i < studentScores.length; i++) {
    System.out.println(“Student Score: ” + studentScores[i]);
}
```

Mảng

Sử dụng vòng lặp for-each

- For-each là phương pháp khác để duyệt mảng
- Được đưa vào từ Java 1.5

Cú pháp

```
for (type variable : array) {
    statement;
}
```

Mảng

Sử dụng vòng for-each

```
int[] studentScores = {45, 75, 63, 48, 89};
for (int score : studentScores) {
    System.out.print(“Student Score ” + score);
}
```

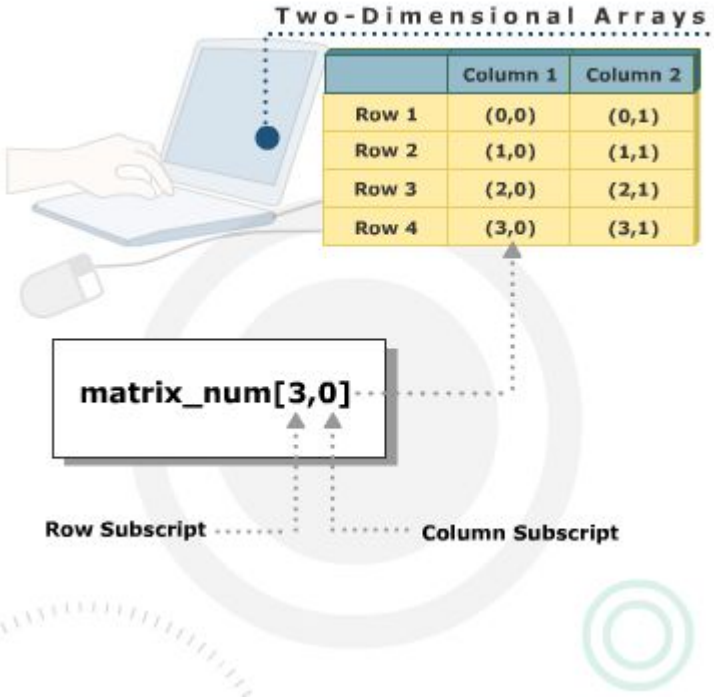
Mảng

Mảng 2 chiều (nhiều chiều)

- là mảng có nhiều hơn 1 chiều
- có thể có 3 chiều hoặc nhiều hơn
- được điều khiển bởi 2 biến chỉ số

Cú pháp

```
datatype[][] arrayName =
new datatype[rowsize][colsize];
```



Mảng

Cú pháp

```
datatype[][] arrayName = new datatype[rowsize][colsize];
```

Ví dụ

```
Int[][] matrix_num = new int[4][2];
```

Mảng

Lưu trữ giá trị vào mảng

- Lưu giá trị ngay từ khi khởi tạo
- Sử dụng vòng lặp để khởi tạo

Mảng

Lưu giá trị khi khởi tạo - 1

```
int[][] stuMarks = new int[2][3]
studentScores[0][0] = 45;
studentScores[0][1] = 65;
...
studentScores[1][0] = 73;
studentScores[1][1] = 25;
studentScores[1][2] = 89;
```

Mảng

Lưu giá trị khi khởi tạo - 2

```
int[][] studentScores = {{45, 75}, {63, 48}, {89, 90}};
```

Mảng

Sử dụng vòng lặp For để khởi tạo

```
int[] stuMarks = new int[2][3];
Scanner input = new Scanner(System.in);
for (int rowIndex = 0; rowIndex < 2; rowIndex++) {
    for (int colIndex = 0; colIndex < 3; colIndex++) {
        System.out.print(“Enter value: ”);
        stuMarks[rowIndex][colIndex] = input.nextInt();
    }
}
```

Mảng

Hiển thị giá trị mảng - Duyệt mảng

- Sử dụng vòng lặp for thông thường

Mảng

Hiển thị giá trị mảng - Duyệt mảng

- Sử dụng vòng lặp for thông thường

```
int[] stuMarks = new int[2][3];
for (int rowIndex = 0; rowIndex < 2; rowIndex++) {
    for (int colIndex = 0; colIndex < 3; colIndex++) {
        System.out.print(stuMarks[rowIndex][colIndex]);
    }
}
```

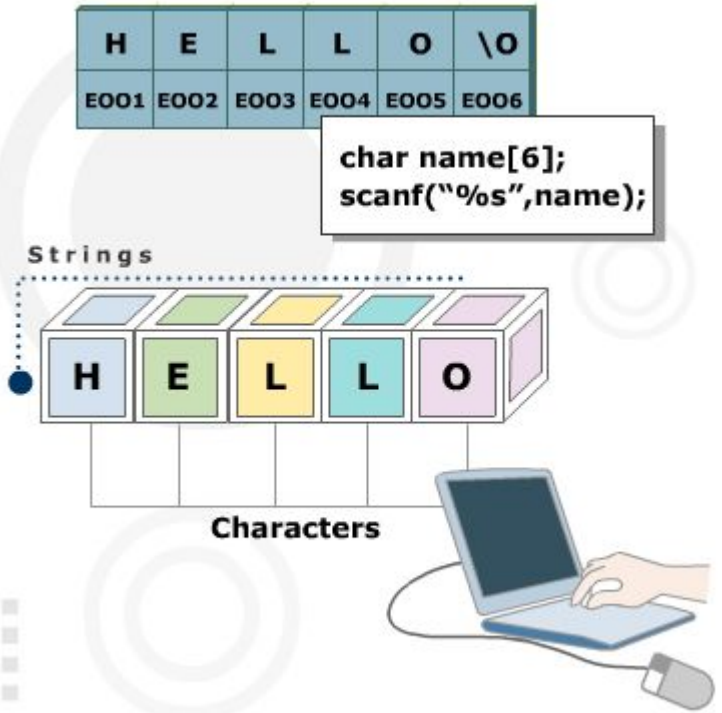
Chuỗi - Strings

1. Định nghĩa String
2. Các phương thức của String
3. Khai báo và sử dụng String

Chuỗi - Strings

Định nghĩa

- là một mảng ký tự
- lưu trữ các thông tin không phải số như: tên, địa chỉ, email,...



Chuỗi - Strings

Tạo ra 1 chuỗi - String

- Sử dụng lớp String của Java cung cấp
- được tạo bằng cách sử dụng từ khoá **new**

Ví dụ

```
String str = new String();
```

Gán gía trị cho chuỗi

```
str = “Hello World”;
```

Chuỗi - Strings

Một số phương thức (method) cả Chuỗi (String)

- lenght()
- charAt()
- concat()
- compareTo()
- indexOf()
- lastIndexOf()

Chuỗi - Strings

Một số phương thức (method) cả Chuỗi (String)

- replace()
- substring()
- toString()
- trim()

Các lớp thao tác với Chuỗi - String

1. Lớp StringBuilder
2. Lớp StringTokenize

Các lớp thao tác với Chuỗi - String

Lớp StringBuilder

- giúp tạo 1 Chuỗi có thể chỉnh sửa, mở rộng một cách linh hoạt và mềm dẻo.
- tiết kiệm tài nguyên, tối ưu hiệu năng khi sử dụng với các chuỗi có kích thước lớn.

Các lớp thao tác với Chuỗi - String

Lớp StringBuilder

- có thể được khởi tạo theo các cách

1. `StringBuilder()`
2. `StringBuilder(int capacity)`
3. `StringBuilder(String str)`

Các lớp thao tác với Chuỗi - String

Các phương thức của lớp StringBuilder

Các l

Phương thức	Miêu tả
public StringBuilder append(String s)	Được sử dụng để phụ thêm (append) chuỗi đã cho với chuỗi này. Phương thức append() được nạp chồng giống dạng append(char), append(boolean), append(int), append(float), append(double) ...
public StringBuilder insert(int offset, String s)	Được sử dụng để chèn chuỗi đã cho với chuỗi này tại vị trí đã cho. Phương thức insert() được nạp chồng giống dạng insert(int, char), insert(int, boolean), insert(int, int), insert(int, float), insert(int, double) ...
public StringBuilder replace(int startIndex, int endIndex, String str)	Được sử dụng để thay thế chuỗi từ chỉ mục ban đầu startIndex và chỉ mục kết thúc endIndex đã cho
public StringBuilder delete(int startIndex, int endIndex)	Được sử dụng để xóa chuỗi từ chỉ mục startIndex và endIndex đã cho
public StringBuilder reverse()	Được sử dụng để đảo ngược chuỗi
public int capacity()	Được sử dụng để trả về dung lượng capacity hiện tại

Các lớp thao tác với Chuỗi - String

public void ensureCapacity(int minimumCapacity)	Được sử dụng để bảo đảm rằng capacity ít nhất bằng với minimum đã cho
public char charAt(int index)	Được sử dụng để trả về ký tự tại vị trí đã cho
public int length()	Được sử dụng để trả về độ dài của chuỗi (chẳng hạn như tổng số ký tự)
public String substring(int beginIndex)	Được sử dụng để trả về chuỗi con từ chỉ mục bắt đầu beginIndex đã cho
public String substring(int beginIndex, int endIndex)	Được sử dụng để trả về chuỗi con từ beginIndex đến endIndex đã cho

Các lớp thao tác với Chuỗi - String

Ví dụ

```
class Main{
    public static void main(String args[]){
        StringBuilder sb=new StringBuilder("Hello ");
        sb.append("Java"); //bay gio chuoi ban dau bi thay doi
        System.out.println(sb); //in ra ket qua la Hello Java
    }
}
```

Các lớp thao tác với Chuỗi - String

Lớp StringTokenizer

- cung cấp nhiều tiện ích cho việc xử lý và thao tác với chuỗi
- phân tách chuỗi thành các token

Các lớp thao tác với Chuỗi - String

Lớp StringTokenizer

Constructor	Miêu tả
StringTokenizer(String str)	Tạo StringTokenizer với chuỗi string đã cho
StringTokenizer(String str, String delim)	Tạo StringTokenizer với chuỗi string và dấu phân tách delimeter đã cho
StringTokenizer(String str, String delim, boolean returnValue)	Tạo StringTokenizer với chuỗi string và dấu tách delimeter và kiểu trả về returnValue đã cho. Nếu kiểu trả về là true, các ký tự phân tách được xem như là các token. Nếu nó là false, các ký tự phân tách phục vụ như là các token riêng rẽ

Các lớp thao tác với Chuỗi - String

Các phương thức của lớp StringTokenizer

Phương thức public	Miêu tả
boolean hasMoreTokens()	Kiểm tra xem có nhiều token có sẵn không
String nextToken()	Trả về token tiếp theo từ đối tượng StringTokenizer
String nextToken(String delim)	Trả về token tiếp theo dựa trên dấu phân tách delim
boolean hasMoreElements()	Giống như phương thức hasMoreTokens()
Object nextElement()	Giống như nextToken() nhưng kiểu trả về của nó là Object
int countTokens()	Trả về tổng số token

Các lớp thao tác với Chuỗi - String

Ví dụ

```
import java.util.StringTokenizer;
public class Simple{
 public static void main(String args[]){
   StringTokenizer st = new StringTokenizer("Toi lam viec o HaNoi"," ");
     while (st.hasMoreTokens()) {
         System.out.println(st.nextToken());
     }
   }
}
```

Các lớp thao tác với Chuỗi - String

Kết quả

```
Toi
lam
viec
o
HaNoi
```