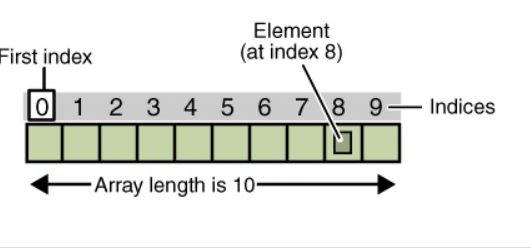
**Chuỗi và mảng**

1. **Mảng**

Thông thường, mảng (array) là một tập hợp các phần tử có cùng kiểu được lưu trữ gần nhau trong bộ nhớ.

Mảng trong java là một đối tượng chứa các phần tử có kiểu dữ liệu giống nhau. Mảng là một cấu trúc dữ liệu nơi lưu trữ các phần tử giống nhau. Với mảng trong java chúng ta chỉ có thể lưu trữ một tập các phần tử có số lượng phần tử cố định.

Mảng trong java lưu các phần tử theo chỉ số, chỉ số của phần tử đầu tiên là 0.



### **Cú pháp khai báo mảng**

Để khai báo mảng, chúng ta cần phải xác định trước **3 thông tin cần thiết** sau:

* Kiểu dữ liệu của mảng.
* Tên của mảng.
* Số lượng các phần tử (hay kích thước) của mảng.

**Câu lệnh khai báo mảng sẽ có 2 dạng như sau:**

**Dạng 1:**

|  |  |
| --- | --- |
| 1 | [Kiểu\_dữ\_liệu] tên\_mảng[]; |

**Dạng 2**:

|  |  |
| --- | --- |
| 1 | [Kiểu\_dữ\_liệu][] tên\_mảng; |

trong đó, [Kiểu\_dữ\_liệu] mô tả kiểu của mỗi phần tử thuộc mảng (như int, char, double, String,...), tên\_mảng là tên của mảng và quy tắc đặt tên phải tuân theo quy tắc đặt tên biến trong Java.

**Ví dụ:** int[] a;: khai báo mảng tên a và có kiểu dữ liệu là int.

### **Cấp phát bộ nhớ cho mảng**

Tương tự như chuỗi, bản chất của mảng là 1 đối tượng (chi tiết về đối tượng chúng ta sẽ được học trong chương Lập trình hướng đối tượng), vì vậy mảng cần phải được cấp phát bộ nhớ trước khi sử dụng. Để cấp phát bộ nhớ cho mảng thì chúng ta có 2 cách như sau:

****Cách 1:****[Kiểu\_dữ\_liệu] tên\_mảng[] = new [Kiểu\_dữ\_liệu] [Số\_phần\_tử\_của\_mảng];

**Cách 2:** [Kiểu\_dữ\_liệu][] tên\_mảng = new [Kiểu\_dữ\_liệu] [Số\_phần\_tử\_của\_mảng];

trong đó, [Số\_phần\_tử\_của\_mảng] chỉ ra số lượng phần tử tối đa mà mảng có thể lưu trữ, giá trị này phải là một số nguyên dương.

Ngoài ra, Java còn cho phép chúng ta vừa có thể khai báo mảng và vừa khởi tạo giá trị cho mảng,

**ví dụ:**

int[] a = new int[] {2, 10, 4, 8, 5};: khai báo mảng một chiều có tên là a, kiểu dữ liệu là int và mảng này chứa 5 phần tử có giá trị lần lượt là 2, 10, 4, 8, 5.

### **Truy xuất các phần tử của mảng**

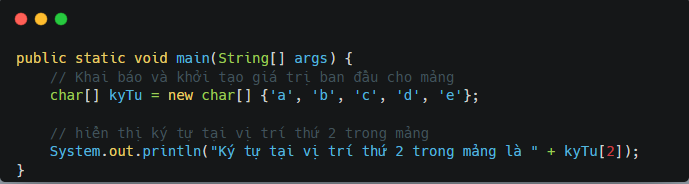
Đối với mảng thì chúng ta có thể truy xuất các phần tử của mảng thông qua các **chỉ số của phần tử đó**. Cú pháp như sau:

|  |  |
| --- | --- |
| 1 | Tên\_mảng[Chỉ\_số\_phần\_tử]; |

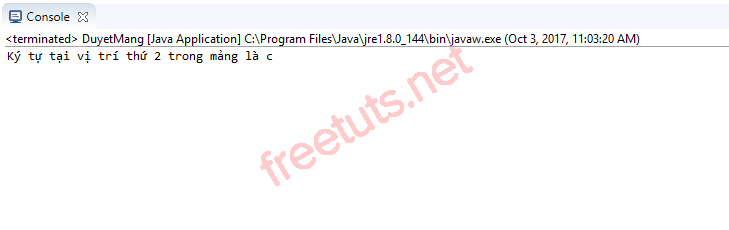
trong đó, [Chỉ\_số\_phần\_tử] là số thứ tự của các phần tử trong mảng và bắt đầu từ 0. Như vậy, mảng có n phần tử thì các phần tử của nó có chỉ số lần lượt là 0, 1, 2,..., n - 1.

****Ví dụ:****

Chúng ta có đoạn chương trình sau:



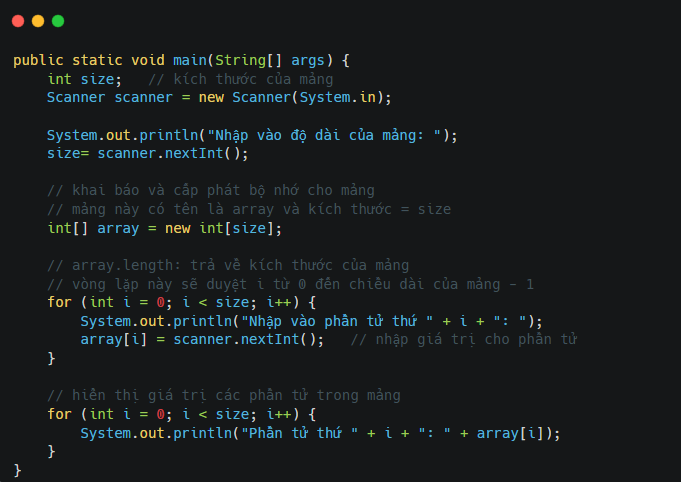
**Kết quả sau khi biên dịch chương trình:**



* 1. **Nhập xuất Mảng**

Chương trình dưới đây sẽ minh họa cách nhập các phần tử cho mảng một chiều từ bàn phím và sau đó hiển thị các phần tử đó ra màn hình.

**Ví dụ**



**Kết quả sau khi biên dịch chương trình:**



1. **Chuỗi**

Trong Java, để khai báo 1 chuỗi ký tự thì chúng ta có 2 cách như sau:

**Cách 1:**

**Cú pháp**

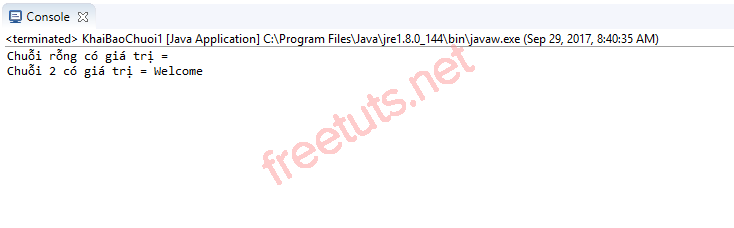
String tenChuoi = "giá\_trị\_khởi\_tạo";

trong đó, giá\_trị\_khởi\_tạo của chuỗi có thể có hoặc không và nếu có thì phải đặt trong cặp dấu " ". Nếu một chuỗi có giá\_trị\_khởi\_tạo = " " thì chuỗi đó được gọi là chuỗi rỗng.

**Ví dụ**



**Kết quả sau khi biên dịch chương trình:**



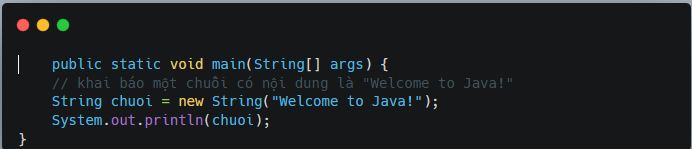
**Cách 2:** Sử dụng từ khóa new.

**Cú pháp**

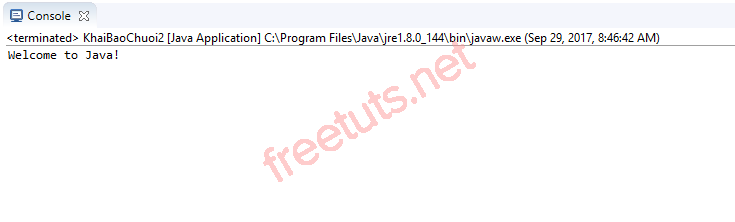
String tenChuoi = new String("giá\_trị");

trong đó giá\_trị là một chuỗi bất kỳ và phải đặt trong cặp dấu " ".

**Ví dụ**



**Kết quả sau khi biên dịch chương trình:**



**4.1. Hàm xác định độ dài chuỗi ký tự**

**Cú pháp**

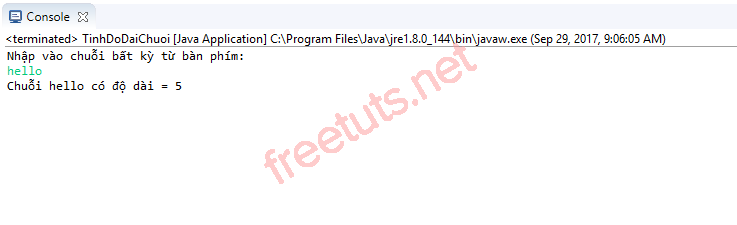
int length = tên\_chuỗi.length();

**Chức năng:** Hàm trả về độ dài chuỗi ký tự bằng cách đếm các ký tự trong chuỗi.

**Ví dụ**



**Kết quả sau khi biên dịch chương trình:**



**4.2. Hàm nối 2 chuỗi ký tự**

Trong Java, để nối 2 chuỗi ký tự lại với nhau thì chúng ta có 2 cách. Cách thứ nhất là chúng ta dùng dấu + để nối chuỗi (cách này tôi đã đề cập trong các bài trước) và cách thứ hai là dùng phương thức concat().

**Cú pháp**

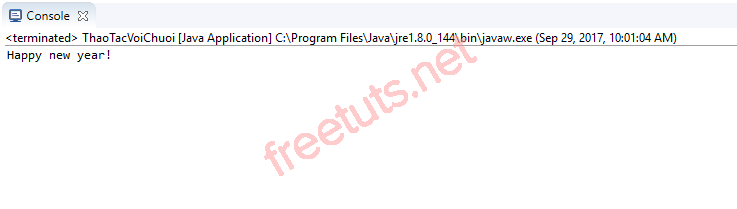
String string3 = string1.concat(String string2);

**Chức năng**: Hàm có tác dụng nối chuỗi string2 vào string1 và trả về chuỗi string3.

**Ví dụ**



**Kết quả sau khi biên dịch chương trình:**



**4.3. Hàm trả về một ký tự trong chuỗi.**

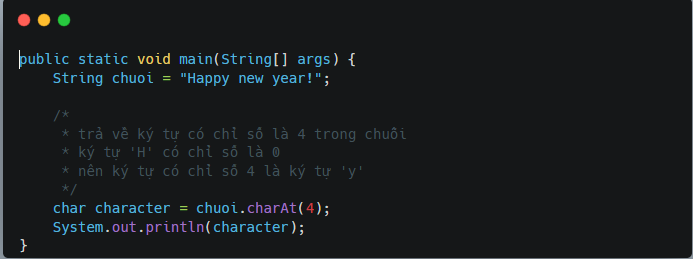
**Lưu ý:** Một chuỗi là tập hợp các ký tự và ký tự đầu tiên trong chuỗi sẽ có chỉ số (index) là 0. Ví dụ: một chuỗi có chiều dài là 10 thì chỉ số của các ký tự trong chuỗi đó sẽ được đánh số từ 0 đến 9.

**Cú pháp**

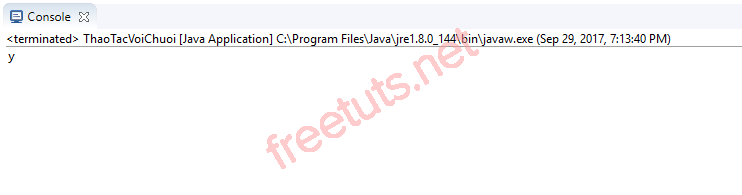
char character = chuoi.charAt(int index);

**Chức năng:** Hàm trả về ký tự character trong chuỗi có chỉ số là index.

**Ví dụ**



**Kết quả sau khi biên dịch chương trình:**



**4.4. Hàm so sánh 2 chuỗi ký tự**

**Cú pháp**

int result = string1.compareTo(String string2);

**Chức năng:** Hàm có tác dụng so sánh hai chuỗi string1, string2 và trả về kết quả:

Nếu result = 0 thì hai chuỗi đó bằng nhau.

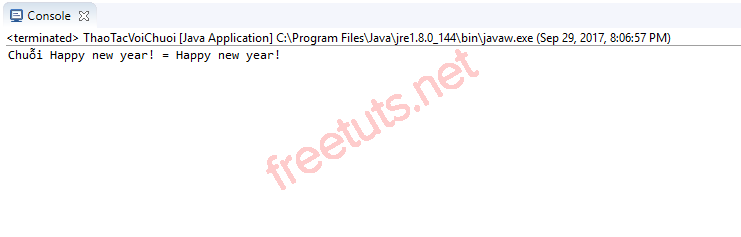
Nếu result < 0 thì chuỗi string1 < string2.

Nếu result > 0 thì chuỗi string1 > string2.

**Ví dụ**



**Kết quả sau khi biên dịch chương trình:**



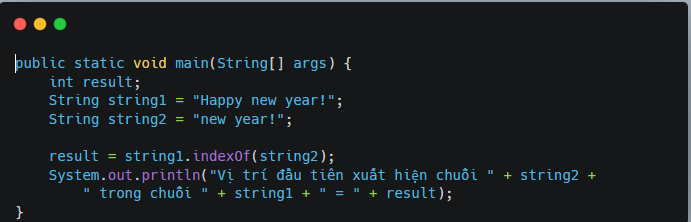
**4.5. Hàm trả về vị trí xuất hiện đầu tiên của một chuỗi trong chuỗi khác.**

**Cú pháp**

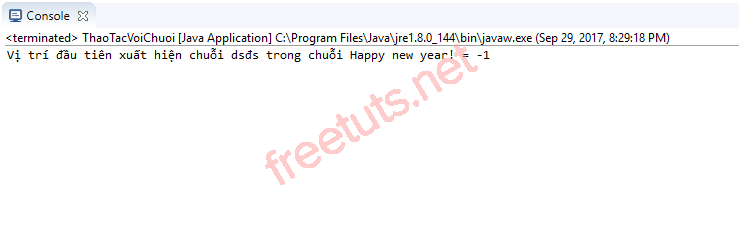
int result = string1.indexOf(String string2);

**Chức năng:** Hàm trả về vị trí xuất hiện đầu tiên của chuỗi string2 trong string1. Nếu chuỗi string2 không có trong chuỗi string1 thì kết quả trả về result = -1.

**Ví dụ**



**Kết quả sau khi biên dịch chương trình:**



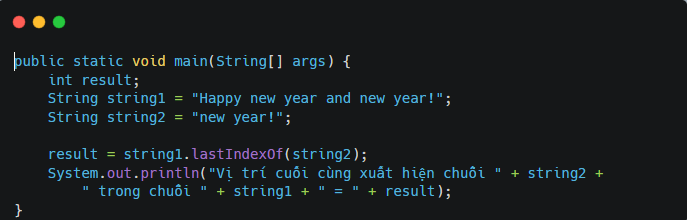
**4.6. Hàm trả về vị trí xuất hiện cuối cùng của một chuỗi trong chuỗi khác.**

**Cú pháp**

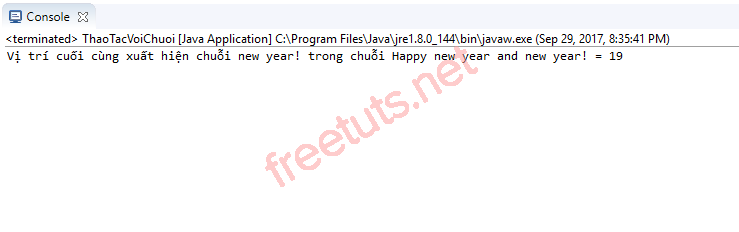
int result = string1.lastIndexOf(String string2);

**Chức năng:** Hàm trả về vị trí xuất hiện cuối cùng của chuỗi string2 trong string1. Nếu chuỗi string2 không có trong chuỗi string1 thì kết quả trả về result = -1.

**Ví dụ**



**Kết quả sau khi biên dịch chương trình:**



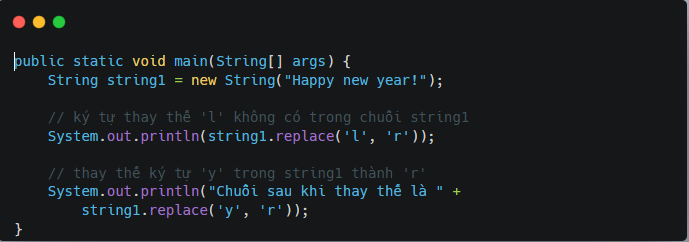
**4.7. Hàm thay thế một chuỗi con trong chuỗi ký tự bằng chuỗi khác**

**Cú pháp**

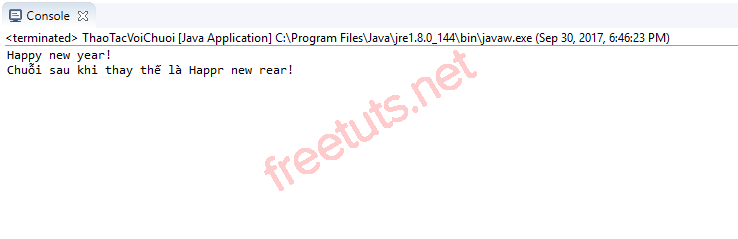
string1.replace(char oldChar, char newChar);

**Chức năng:**Hàm sẽ thay thế ký tự oldChar bằng ký tự newChar trong chuỗi string1. Nếu ký tự cần thay thế không có trong chuỗi string1 thì chương trình sẽ trả về chuỗi string1.

**Ví dụ**



**Kết quả sau khi biên dịch chương trình:**



**4.8. Hàm loại bỏ các khoảng trắng thừa ở đầu và cuối chuỗi**

**Cú pháp**

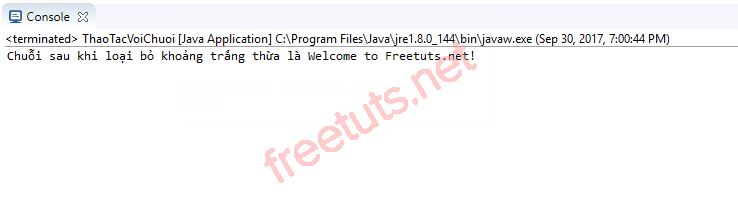
String string1 = string1.trim();

**Chức năng:** Hàm sẽ loại bỏ các khoảng trắng thừa ở đầu và cuối chuỗi string1. Nếu chuỗi đó không có khoảng trắng thừa thì chương trình sẽ trả về chuỗi gốc.

**Ví dụ**



**Kết quả sau khi biên dịch chương trình:**



**4.9. Hàm tạo một chuỗi con từ vị trí index trong chuỗi cha**

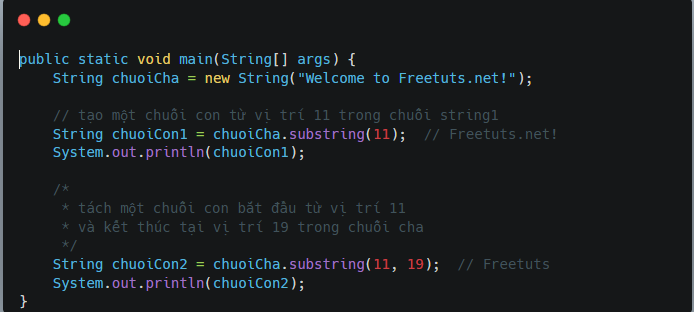
**Cú pháp**

String chuoiCon = chuoiCha.substring(int beginIndex);

String chuoiCon = chuoiCha.substring(int beginIndex, int endIndex);

**Chức năng:** Hàm sẽ tạo một chuỗi con từ vị trí có chỉ số là beginIndex trong chuỗi cha. Trong cú pháp thứ 2, thì hàm sẽ tạo một chuỗi con bắt đầu từ vị trí có chỉ số là beginIndex và kết thúc tại vị trí có chỉ số endIndex - 1 trong chuỗi cha.

**Ví dụ**



**Kết quả sau khi biên dịch chương trình:**

