

Java Vector class - Lớp Vector trong Java

Một trong các vấn đề với một mảng là chúng ta phải biết nó lớn như thế nào khi chúng ta tạo nó. Nó thì không thể xác định kích thước của mảng trước khi tạo nó.

Lớp Vector của Java giải quyết vấn đề này. Nó cung cấp một dạng mảng với kích thước ban đầu, mảng này có thể tăng thêm khi nhiều phần tử được thêm vào. Một lớp Vector lưu trữ các item của kiểu Object, nó có thể dùng để lưu trữ các thể hiện của bất kỳ lớp Java nào. Một lớp Vector đơn lẻ có thể lưu trữ các phần tử khác nhau, các phần tử khác nhau này là thể hiện của các lớp khác nhau.

Package: *java.util*

Tại bất kỳ thời điểm, một lớp Vector có dung lượng để lưu trữ một số nào đó của các phần tử. Khi một lớp Vector biết về dung lượng của nó, thì dung lượng của nó được gia tăng bởi một số lượng riêng cho Vector đó. Lớp Vector cung cấp ba phương thức xây dựng khác nhau mà có thể chúng ta chỉ định dung lượng khởi tạo, và tăng số lượng của một Vector, khi nó được tạo ra.

Các phương thức khởi tạo của lớp Vector được tóm tắt trong bảng sau:

STT	Constructor	Desc
1	Vector(int)	Tạo ra một lớp Vector mới với dung lượng ban đầu được chỉ định.
2	Vector(int, int)	Tạo ra một lớp Vector mới với dung lượng ban đầu được chỉ định, và tăng số lượng.
3	Vector()	Tạo ra một lớp Vector mới với dung lượng khởi tạo mặc định, và tăng số lượng.

Một mục (item) được thêm vào một lớp Vector bằng cách sử dụng hàm `addElement()`. Tương tự, một phần tử có thể được thay thế bằng cách sử dụng hàm `setElementAt()`. Một lớp Vector có thể tìm kiếm bằng cách sử dụng phương thức `contains()`, phương thức này trông có vẻ dễ dàng cho một lần xuất hiện của một đối tượng (Object). Phương thức `elements()` thì hữu dụng bởi vì nó trả về một bảng liệt kê của các đối tượng được lưu trữ trong lớp Vector.

Các phương thức này và các phương thức thành viên khác của lớp Vector được tóm tắt trong bảng dưới đây:

STT	Method	Desc
1	addElement(Object)	Chèn các phần tử được chỉ định vào lớp Vector.
2	capacity()	Trả về số phần tử mà sẽ vừa đủ cho phần được cấp phát hiện thời của lớp Vector.
3	clone()	Tạo bản sao của vector, nhưng không phải là các phần tử của nó.
4	contains(Object)	Trả về True nếu lớp Vector chứa đối tượng được chỉ định.
5	copyInto(Object [])	Sao chép các phần tử của lớp Vector vào mảng được chỉ định.
6	elementAt(int)	Truy lục phần tử được cấp phát tại chỉ mục được chỉ định.
7	elements()	Trả về một bảng liệt kê của các phần tử trong lớp Vector.
8	ensureCapacity(int)	Chắc chắn rằng lớp Vector có thể lưu trữ ít nhất dung lượng tối thiểu được chỉ định.
9	firstElement()	Trả về phần tử đầu tiên trong lớp Vector.
10	indexOf(Object)	Tìm kiếm lớp Vector, và trả về chỉ mục zero cơ bản cho khớp với đối tượng đầu tiên.
11	indexOf(Object, int)	Tìm kiếm lớp Vector đang bắt đầu tại số chỉ mục được chỉ định, và trả về chỉ mục zero cơ bản cho khớp với đối tượng kế tiếp.
12	insertElementAt(Object, int)	Thêm các đối tượng được chỉ định tại chỉ mục được chỉ định.
13	isEmpty()	Trả về True nếu lớp Vector không có phần tử.
14	lastElement()	Trả về phần tử cuối cùng trong lớp Vector.
15	lastIndexOf(Object)	Tìm kiếm lớp Vector, và trả về chỉ mục zero cơ bản cho khớp với đối tượng cuối cùng.
16	lastIndexOf(Object, int)	Tìm kiếm lớp Vector đang bắt đầu tại số chỉ mục được chỉ định, và trả về chỉ mục zero cơ bản cho khớp với đối tượng trước.
17	removeAllElements()	Xoá tất cả các phần tử từ lớp Vector.

18	<code>removeElement(Object)</code>	Xoá đối tượng được chỉ định từ lớp Vector.
19	<code>removeElementAt(int)</code>	Xoá đối tượng tại chỉ mục được chỉ định.
20	<code>setElementAt(Object, int)</code>	Thay thế đối tượng tại chỉ mục được chỉ định với đối tượng được chỉ định.
21	<code>setSize(int)</code>	Thiết lập kích thước của lớp Vector thành kích thước mới được chỉ định.
22	<code>setSize(int)</code>	Thiết lập kích thước của lớp Vector thành kích thước mới được chỉ định.
23	<code>size()</code>	Trả về số của các phần tử hiện thời trong lớp Vector.
24	<code>toString()</code>	Trả về một đại diện chuỗi được định dạng nội dung của lớp Vector.
25	<code>trimToSize()</code>	Định lại kích thước của lớp Vector để di chuyển dung lượng thừa trong nó.

Chương trình sau tạo ra một lớp Vector "vect". Nó chứa 6 phần tử: "Numbers In Words", "One", "Two", "Three", "Four", "Five". Phương thức `removeElement()` được sử dụng để xoá các phần tử từ "vect".

```
import java.util. * ;
public class VectorImplementation {
    public static void main(String args[]) {
        Vector vect = new Vector();
        vect.addElement("One");
        vect.addElement("Two");
        vect.addElement("Three");
        vect.addElement("Four");
        vect.addElement("Five");
        vect.insertElementAt("Numbers In Words", 0);
        vect.insertElementAt("Four", 4);
        System.out.println("Size: " + vect.size());
        System.out.println("Vector");
        for (int i = 0; i < vect.size(); i++) {
            System.out.println(vect.elementAt(i) + ", ");
        }
        vect.removeElement("Five");
        System.out.println("");
        System.out.println("Size: " + vect.size());
        System.out.println("Vector");
        for (int i = 0; i < vect.size(); i++) {
```

```
        System.out.print(vect.elementAt(i) + ", ");  
    }  
}  
}
```

Revision #3

Created 26 December 2020 15:51:06 by Laptrinh.vn

Updated 9 January 2021 17:48:57 by Laptrinh.vn