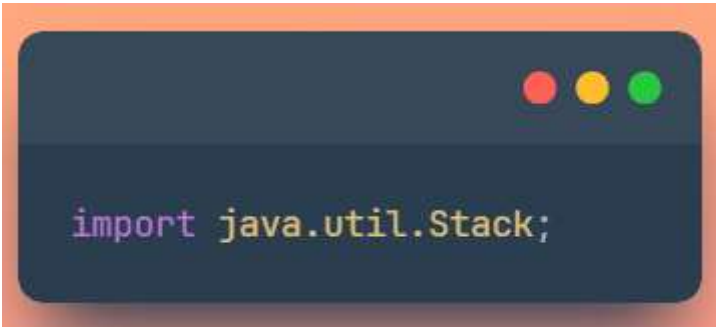


🕒 Cập nhật tháng 8 năm 2024

[Bài đọc] Các phương thức làm việc với Stack


1. Làm việc với Stack trong Java

- Trong Java, **Stack** là một lớp nằm trong gói `java.util` và kế thừa từ lớp `Vector`
- Để sử dụng:



```
import java.util.Stack;
```

- Tạo một stack:



```
import java.util.Stack;

Stack<Integer> stack = new Stack<>();
```

2. Các phương thức làm việc với Stack

- **push(E item)** – Thêm phần tử vào đỉnh stack
 - Thêm một phần tử mới lên đầu Stack (đỉnh)

```
import java.util.Stack;

Stack<Integer> stack = new Stack<>();

stack.push(10);
```

- **pop()** – Loại bỏ và trả về phần tử trên cùng
 - Lấy phần tử trên đỉnh và **xóa** nó khỏi stack. Nếu stack rỗng, ném `EmptyStackException`

```
import java.util.Stack;

Stack<Integer> stack = new Stack<>();

int top = stack.pop();
```

- **peek()** – Lấy phần tử trên cùng mà không xóa
 - Trả về phần tử ở đỉnh nhưng **không xóa** nó khỏi stack. Nếu stack rỗng, cũng ném `EmptyStackException`

```
import java.util.Stack;

Stack<Integer> stack = new Stack<>();

int top = stack.peek();
```

- **empty()** – Kiểm tra stack có rỗng không
 - Trả về `true` nếu stack không chứa phần tử nào, ngược lại `false`

```
import java.util.Stack;

Stack<Integer> stack = new Stack<>();

if (stack.empty()) {
    System.out.println("Stack rỗng");
}
```

- **search(Object o)** - Tìm vị trí tương đối của phần tử
 - Trả về vị trí (**1-based**) của phần tử tính từ **đỉnh stack**. Trả về -1 nếu không tìm thấy

```
import java.util.Stack;

Stack<Integer> stack = new Stack<>();

int position = stack.search(10);
```

3. Ví dụ tổng hợp:

```
import java.util.Stack;

public class StackMethodsExample {
    public static void main(String[] args) {
        Stack<String> stack = new Stack<>();

        stack.push("A");
        stack.push("B");
        stack.push("C");

        System.out.println("Stack hiện tại: " + stack);    // [A, B, C]
        System.out.println("Phần tử trên cùng: " + stack.peek()); // C

        stack.pop();
        System.out.println("Sau khi pop: " + stack);      // [A, B]

        System.out.println("Có rỗng không? " + stack.empty()); // false
        System.out.println("Vị trí của A: " + stack.search("A")); // 2
    }
}
```

4. Một vài lưu ý:

- Stack trong Java kế thừa từ Vector, vì vậy nó **đồng bộ (synchronized)**
- Nếu bạn cần hiệu năng cao hơn và không cần đồng bộ, bạn có thể dùng Deque (như ArrayDeque) thay thế

Tài nguyên đọc thêm: <https://www.geeksforgeeks.org/java/stack-class-in-java>

Danh sách các bài học

