

# Bài 66: Sử dụng stack và queue

---

- ✓ Sử dụng stack
- ✓ Sử dụng queue
- ✓ Ví dụ minh họa

# Stack

- Khái niệm: stack hay còn gọi ngăn xếp, là một cấu trúc dữ liệu theo tiêu chí FILO, tức vào sau ra trước
- Có thể hình dung một stack như cái thùng có 1 đáy. Phần tử vào đầu tiên ở dưới cùng, phần tử vào sau ở trên phần tử vào trước
- Các hành động chính của stack gồm:
  - push(): thêm phần tử vào đầu stack
  - pop(): lấy và xóa phần tử ở đầu stack
  - peek(): lấy phần tử ở đầu và không xóa
  - empty(): kiểm tra stack có rỗng không
  - search(): tìm kiếm phần tử trong stack

# Ứng dụng của stack

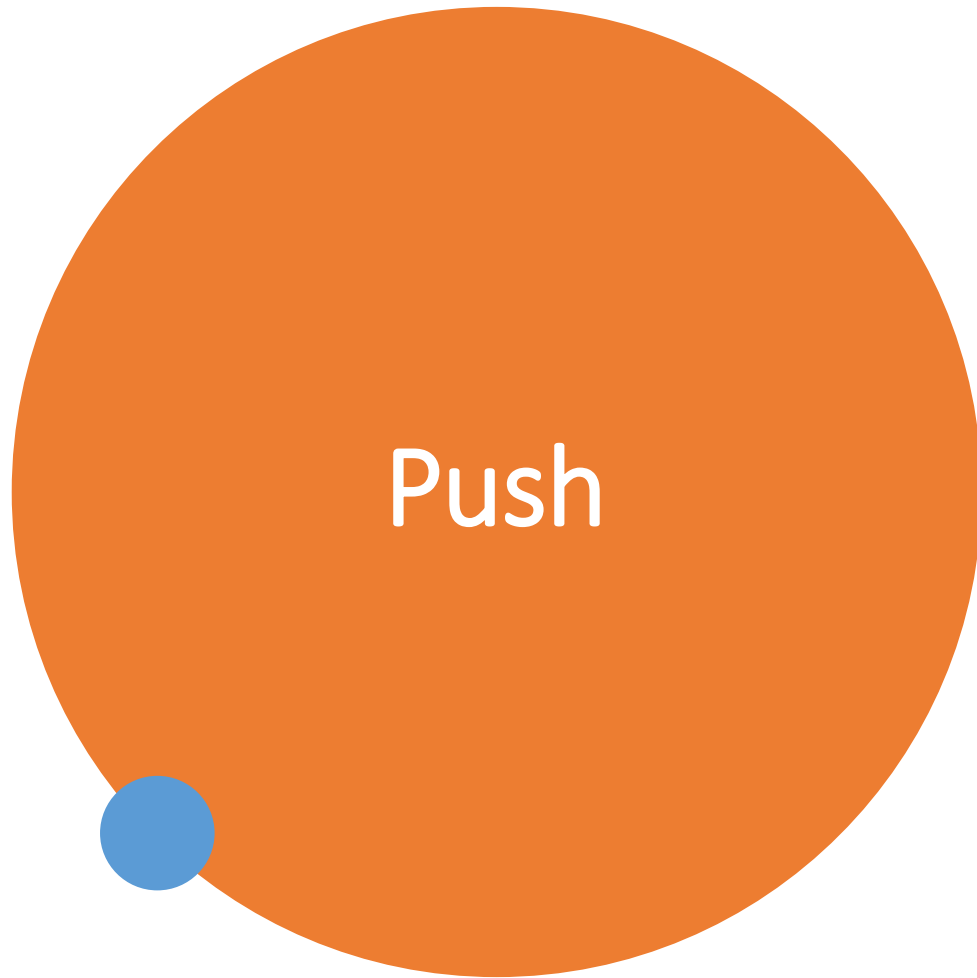
- Lưu các lời gọi của chương trình, thực hiện việc tính toán các biểu thức...

```
public class StackTest {  
    public static void main(String[] args) {  
        method1();  
    }  
  
    private static void method1() {  
        method2();  
    }  
  
    private static void method2() {  
        String str = null;  
        System.out.println(str.length());  
    }  
}
```

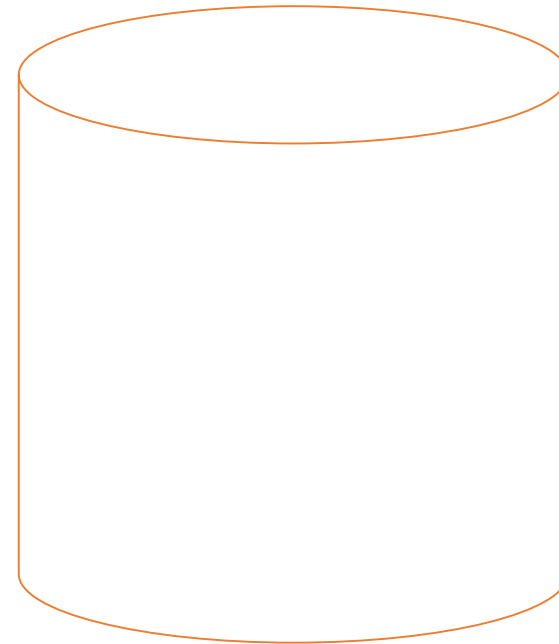
StackTest x

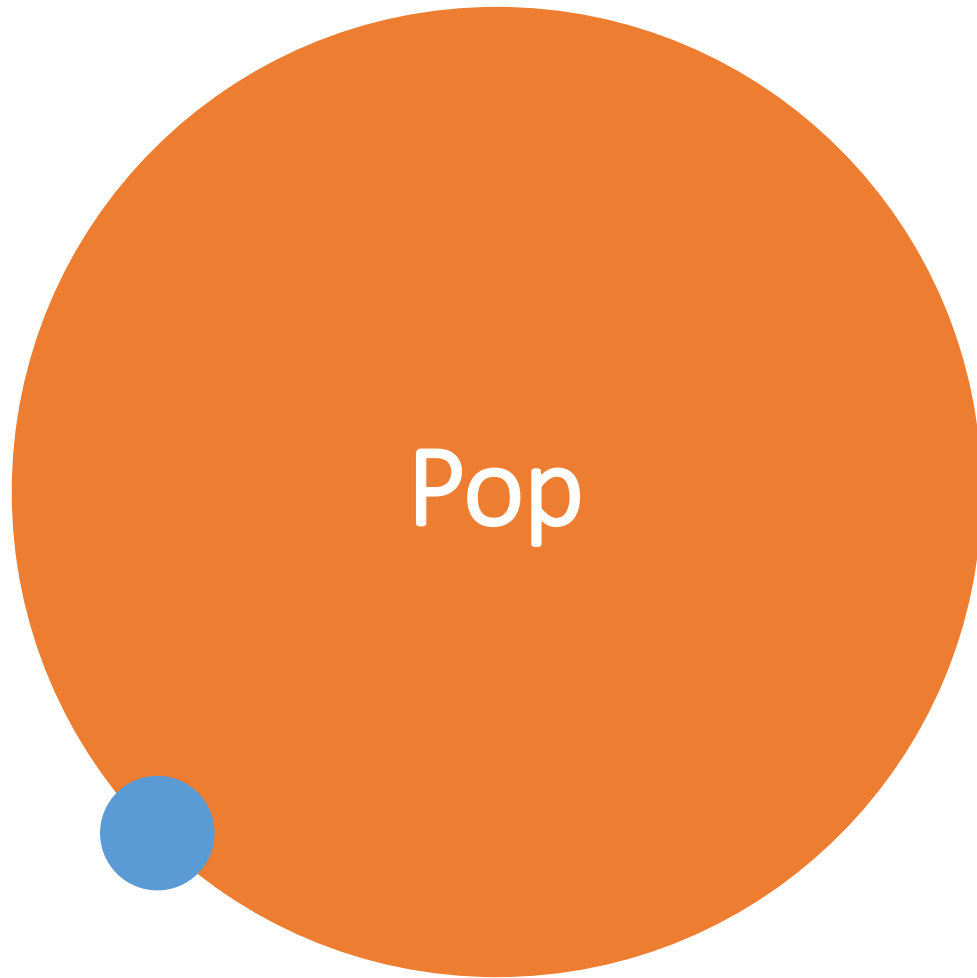
"C:\Program Files\Java\jdk-14.0.1\bin\java.exe" "-javaagent:C:\Program F:  
Exception in thread "main" java.lang.NullPointerException: Create breakpoint  
at lesson65.StackTest.method2(StackTest.java:14)  
at lesson65.StackTest.method1(StackTest.java:9)  
at lesson65.StackTest.main(StackTest.java:5)

The diagram illustrates the call stack flow with three numbered arrows: Arrow 1 points from the `main` method to `method1`; Arrow 2 points from `method1` to `method2`; Arrow 3 points from `method2` back to `main`, indicating the return path after the exception is thrown.

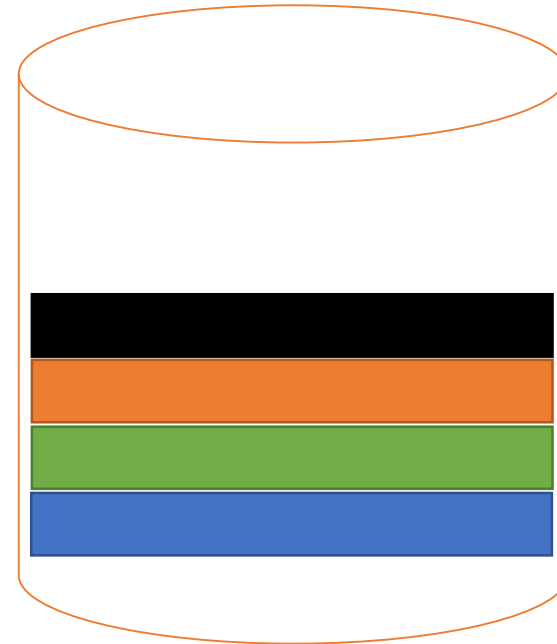


➤ Ví dụ push vào stack:





➤ Ví dụ pop khỏi stack:



# Queue

- Queue là cấu trúc dữ liệu hàng đợi theo tiêu chí vào trước ra trước(FIFO)
- Có thể hình dung queue như một cái ống rỗng hai đầu, ta thả một hòn đá từ đầu này thì nó sẽ ra ở đầu bên kia vậy
- Ứng dụng của queue:
  - Phục vụ yêu cầu trong hệ thống dùng chung tài nguyên, ví dụ máy in, lập lịch CPU
  - Trong thực tế, tổng đài sử dụng queue để xếp thứ tự cuộc gọi đến và phục vụ lần lượt
  - Xử lý lỗi trong hệ thống, lỗi nào xảy ra trước xử lý trước

# Queue

- Một số phương thức chính của Queue:
  - add(): Thêm phần tử vào cuối queue
  - remove(): xóa bỏ phần tử ở đầu queue
  - peek(): lấy phần tử đầu queue nhưng không xóa
- Một số phương thức của ArrayDeque:
  - addFirst(): thêm vào đầu
  - addLast(): thêm vào cuối
  - peekFirst(): lấy ở đầu
  - peekLast(): lấy ở cuối
  - removeFirst(): lấy và xóa phần tử ở đầu
  - removeLast(): lấy và xóa phần tử ở cuối



# Minh họa

➤ Thực hiện trong công cụ lập trình



# Nội dung tiếp theo

Sử dụng `LinkedList`, `HashSet`,  
`TreeSet`