

Lab#09

Implementation of Stack

Objective

- Stack
- Example Programs
- Exercise

Theory

Array: <https://medium.com/@rafia.shaikh61/java-stack-ecb9541dcd16>

Linked list: <https://medium.com/@rafia.shaikh61/java-stack-ii-975afaba1af7>

Code Example

Array: <https://medium.com/@rafia.shaikh61/java-stack-ecb9541dcd16>

Linked list: <https://medium.com/@rafia.shaikh61/java-stack-ii-975afaba1af7>

Constructor Summary

Constructor Stack() or Stack(int length)
Description Creates an empty Stack.

Method

Summary

push()

Method Signature:

```
public T push(T item)
```

Pushes an item onto the top of this stack. This has exactly the same effect as:

```
addElement(item)
```

Parameters:

item - the item to be pushed onto this stack.

Returns:

the item argument.

pop():

Method Signature:

```
public T pop()
```

Removes the object at the top of this stack and returns that object as the value of this function.

Returns:

The object at the top of this stack (the last item of the Vector object).

Throws:

EmptyStackException - if this stack is empty.

peek()

Method Signature:

```
public T peek()
```

Looks at the object at the top of this stack without removing it from the stack.

Returns:

the object at the top of this stack (the last item of the Vector object).

Throws:

EmptyStackException - if this stack is empty.

isEmpty()

Method Signature:

```
public boolean  
isEmpty()
```

Tests if this stack is empty.

Returns:

true if and only if this stack contains no items; false otherwise.

search()

Method Signature:

```
public int search(T item)
```

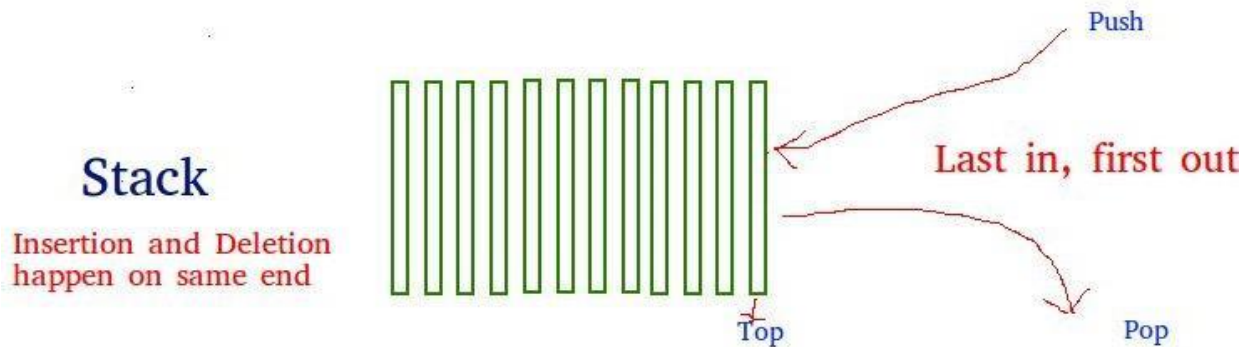
Returns the 1-based position where an item is on this stack. If the item occurs in this stack, this method returns the distance from the top of the stack of the occurrence nearest the top of the stack; the topmost item on the stack is considered to be at distance 1.

Parameters: item to be searched

Returns:

Time Complexities of operations on stack:

push(), pop(), isEmpty() and peek() all take $O(1)$ time. We do not run any loop in any of these operations.

**Applications of stack:**

- Balancing of symbols
- Infix to Postfix /Prefix conversion
- Redo-undo features at many places like editors, photoshop.
- Forward and backward feature in web browsers
- Used in many algorithms like Tower of Hanoi, tree traversals, stock span problem, - histogram problem.

Implementation:

There are two ways to implement a stack:

- Using array
- Using linked list

Stack Implementation using Linked list:

We'll be using the same classes of Linked List that we have developed. Few methods like `add(int Location)`, `remove(int Location)` will be removed now. As stack does not

support addition or removal at any given position. Instead, it only adds and removes any item from last. The stack add function is known as `push()` while deletion is known as `pop()`.

We'll add another function `peek()`, for that we can change the `get(int Location)` method to `peek()`.

Tasks

Write a program to create a stack of 10 student's names.

- a) Implement the `isEmpty()` method in the Stack
- b) Implement the `getSize()` method in the Stack
- c) Implement the `search()` method in the Stack to check whether the element exists in the stack or not. (search student name).
- d) Implement the `push()` method in the Stack and then Push 10 student's names.
- e) Implement the `pop()` method in the Stack and print the stack using `pop`.
- f) Display the all stack data using `display()` method..

Exercise

Question#01: To develop stack using Array (**using generic class**)

- a) Implement the `isEmpty()` method in the Stack
- b) Implement the `getSize()` method in the Stack
- c) Implement the `search()` method in the Stack to check whether the element exists in the stack or not.
- d) Implement the `push()` method in the Stack and then Push 10 values.
- e) Implement the `pop()` method in the Stack and print the stack using `pop`.
- f) Display the all stack data using `display()` method..

Question#02: To develop stack using Linked List implemented in the previous lab (**using generic class**)

- a) Implement the `isEmpty()` method in the Stack

- b) Implement the `getSize()` method in the Stack
- c) Implement the `search()` method in the Stack to check whether the element exists in the stack or not.
- d) Implement the `push()` method in the Stack and then Push 10 values.
- e) Implement the `pop()` method in the Stack and print the stack using `pop`.
- f) Display the all stack data using `display()` method..

Question#03: Implement the **forward** and **backward** feature in web browsers using stack.

(Use either array or linked list it is your choice)

