Stack Data Structure and Swift Sample Code

ddas.tech/stack-data-structure-and-swift-sample-code/

January 2, 2023

In this post we will explore the Stack Data Structure and provide a simple implementation in Swift.

Introduction to Stack Data Structure

Stack is an abstract data type and is one of the simplest data structures. It has LIFO – Last In first out order
Stack is like an array but with limited functionality.

Stack Data Structure Functionalities

Some of the functionalities available in Stack data structures are as follows

- Push to add a new object to the top
- Pop to remove the last object
- Top to look at the top most/last element without removing it

- In a protected Singly Linked List
  - ***Push*** -> Insert at head
  - ***Pop*** -> Remove from head
  - ***Peek*** -> Look at the head Item
- **There are multiple applications of a Stack Data Structure** such as
  - Bracket Matching
  - Post Fix Calculator

```
Logic for Implementing a Stack Data Structure is as follows
POP
if empty , do nothing
temp = head
head = head.next
delete temp
Time complexity is O(1)

PUSH
Vertex vtx = new Vertex(v)
vtx.next = head
head = vtx

PEEK
if empty return not found
return head.item
```

Stack Data Structure – Sample Implementation in Swift

```
import Cocoa
//Here the stack is created as a wrapper around a swift array that allows
//push an object to the stack,
//pop the last object and
//look at the top element of the stack
public struct Stack<T> {
    fileprivate var array = [T]()

    public var isEmpty: Bool{
        return array.isEmpty
    }
    public var count: Int{
        return array.count
    }
    public mutating func push(_ element:T){
        array.append(element)
    }
    public mutating func pop() -> T?{
        return array.popLast()
    }
    public var top: T? {
        return array.last
    }
}

//Testing the stack implementation
var stack = Stack<Int>()
stack.push(10)
print(stack)
//Stack<Int>(array: [10])

stack.push(20)
print(stack)
//Stack<Int>(array: [10, 20])

stack.push(30)
print(stack) //Stack<Int>(array: [10, 20, 30])
if let val = stack.pop(){
    print("The popped item = \(val)")
    //The popped item = 30
}
print(stack) //Stack<Int>(array: [10, 20])

if let topElement = stack.top{
    print("The Top Element is = \(topElement)")
}
//The Top Element is = 20

print(stack)
//Stack<Int>(array: [10, 20])
```

```
//The pop removes the last element however
//the top only looks at the last element without removing it.
```