SWIFT
Data Structure - Stack

Bill Kim(김정훈) | ibillkim@gmail.com

목차

Stack

스택(Stack)는 리스트의 한쪽 끝에서 자료의 삽입과 삭제가 이루어지는 자료구조입니다.

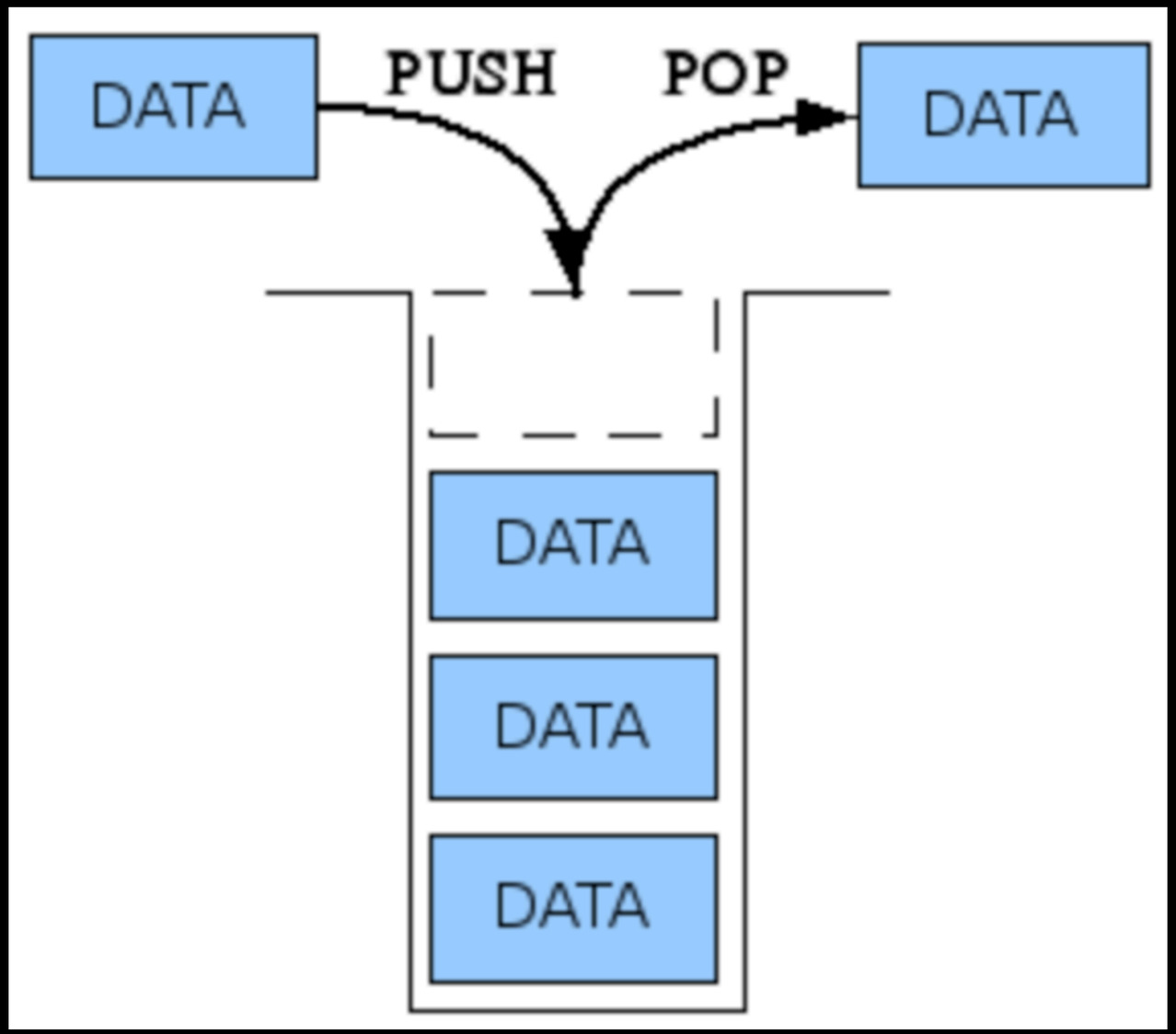
가장 최근에 들어온 자료가 가장 먼저 나가게 되는 LIFO(Last-In First-Out) 형태를 가지고 있습니다.

따라서 중간에서는 데이터를 추가 및 삭제하는 것이 불가능합니다

스택이 입출력이 이루어지는 부분을 스택 상단(Stack top) , 바닥 부분을 스택 하단(Stack bottom), 스택에 저장되는 것을 요소(Element) 라 부르며 스택에 요소가 하나도 없을 때 그러한 스택을 공백 스택(Empty stack) 이라고 합니다.

Concept

Stack의 기본 동작 흐름은 아래와 같습니다.



Features

Stack의 특징을 살펴보면 아래와 같습니다.

- 데이터를 한 곳에서 삽입, 삭제
- Last In First Out (LIFO, 후입선출) 방식
- 중간에서 데이터 삽입 및 삭제 불가
- 배열과 비슷하지만 삽입, 삭제 등이 더 제한적임

Implementation

Swift를 활용하여 Stack 을 구현해보겠습니다.
우선 필요한 메소드는 아래와 같습니다.

- init : 리스트를 초기화하는 함수
- push : 데이터 입력
- pop : 마지막 데이터 삭제
- peak : 마지막 데이터 반환
- count : 현재 리스트의 크기를 반환
- isEmpty : 현재 리스트의 크기가 비어있는지 체크

Implementation

```
class Stack<T> {
    private var elements = Array<T>()
    public init() {}

    public func push(element: T) {
        self.elements.append(element)
    }

    public func pop() -> T? {
        return self.elements.popLast()
    }

    public func peek() -> T? {
        return self.elements.last
    }

    public var isEmpty: Bool {
        return self.elements.isEmpty
    }

    public var count: Int {
        return self.elements.count
    }
}
```

Implementation

데이터 반복(Iterator) 제어를 위하여 아래의 프로토콜들을 체택합니다.

```
public struct ArrayIterator<T> : IteratorProtocol {
    var currentElement: [T]
    init(elements: [T]) {
        self.currentElement = elements
    }

    public mutating func next() -> T? {
        if !self.currentElement.isEmpty {
            return self.currentElement.popLast()
        }
        return nil
    }
}

extension Stack : Sequence {
    public func makeIterator() -> ArrayIterator<T> {
        return ArrayIterator<T>(elements: self.elements)
    }
}
```

Implementation

```swift
let stack = Stack<Int>()

stack.push(1)
stack.push(2)
stack.push(3)
stack.push(4)
stack.push(5)

// 현재 데이터 카운트 : 5
print(stack.count)

for node in stack {
    print(node)
    // 5
    // 4
    // 3
    // 2
    // 1
}
```

References

[1] [스위프트 : 자료구조] 스택 : Stack : https://the-brain-of-sic2.tistory.com/38

[2] [스위프트:자료구조] 스택: Stack: 자료구조: DataStructure: 쌓기: swift : https://the-brain-of-sic2.tistory.com/18

[3] [Swift 자료구조 ch04] Stack : https://kor45cw.tistory.com/240

[4] Swift로 작성해보는 기본 자료구조 - Stack, Queue : https://wlaxhrl.tistory.com/87

[5] Stack : https://woongsios.tistory.com/87

References

[6] 스택(Stack) : https://kka7.tistory.com/74

[7] Swift, Data Structure, Stack : https://devmjun.github.io/archive/Stack

[8] [알고리즘] 2.1. 자료구조 : 스택(Stack) 이해하기 : https://monsieursongsong.tistory.com/4

[9] [자료구조]스택과 큐. (stack and queue) : https://winplz.tistory.com/entry/자료구조스택과-큐-stack-and-queue

[10] 자료구조(1) - 스택(Stack)이란 무엇일까? : https://minosaekki.tistory.com/8

Thank you!