

SWIFT

Data Structure - Array

Bill Kim(김정훈) | ibillkim@gmail.com

목차

Array List

Concept

Features

Implementation

References

Array List

자료구조에서 가장 기본이 되는 구조는 배열(리스트)입니다.
리스트는 크게 아래와 같이 2개의 리스트로 구분할 수 있습니다.

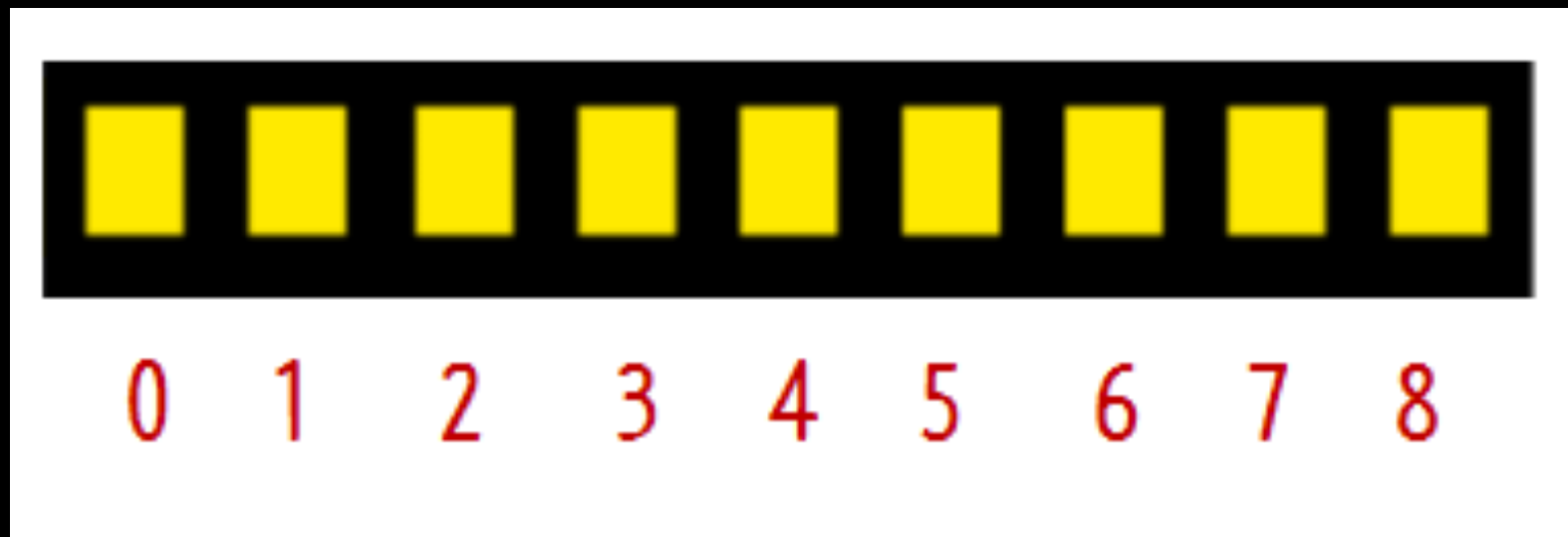
순차 리스트 :
배열을 기반으로 구현된 리스트

연결 리스트 :
메모리의 동적 할당을 기반으로 구현된 리스트

본 강의에서는 Swift 코드를 활용하여 순차 리스트에 대해서 살펴
봅니다.

Concept

순차리스트는 아래와 같이 특정 자료를 정해진 공간안에서 순차적으로 저장 및 가져오는 자료구조입니다.



Features

순차 리스트(Array)의 특징을 살펴보면 아래와 같습니다.

- 데이터를 순차적으로 저장합니다.
- 데이터 중복 저장을 허용합니다.
- 배열의 총 길이는 초기에 결정되어야 합니다.(변경 불가)
- 인덱스 값을 기준으로 데이터의 참조가 가능합니다.
- 삭제의 과정에서 데이터의 이동이 빈번하게 일어납니다.

Implementation

Swift를 활용하여 Array 를 구현해보겠습니다.
우선 필요한 메소드는 아래와 같습니다.

- `init()` : 배열의 크기를 설정하는 초기화 함수
- `insert()` : 데이터 입력
- `remove()` : 특정 인덱스의 데이터 삭제
- `removeAll()` : 모든 데이터 삭제
- `get()` : 특정 인덱스의 데이터 값을 반환
- `count()` : 현재 배열의 크기를 반환

Implementation

가장 데이터의 기본이 되는 Node 클래스입니다.

해당 **Node** 클래스는 모든 데이터 형식을 받을 수 있도록 **Generic** 형태로 구현이 되어 있습니다.

```
class Node<T> {  
    fileprivate var data:T  
  
    init(_ data: T) {  
        self.data = data  
    }  
}
```

Implementation

```
class ArrayList<T> {
    private var size:Int
    private var list:[Node<T>]

    init(size:Int) {
        self.list = [Node<T>]()
        self.size = size
    }

    func insert(item:T) {
        if self.list.count < size {
            self.list.append(Node<T>(item))
        }
    }

    func remove(at:Int) {
        if self.list.count <= 0 { return }
        if self.list.count <= at { return }

        self.list.remove(at: at)
    }

    func removeAll() {
        self.list.removeAll()
    }

    func get(index:Int) -> T {
        return self.list[index].data
    }

    func count() -> Int {
        size = list.count

        return size
    }
}
```


Implementation

아래와 같이 사용이 가능합니다.

```
let list:ArrayList<Int> = ArrayList(size: 5)

list.insert(item: 1)
list.insert(item: 2)
list.insert(item: 3)
list.insert(item: 4)
list.insert(item: 5)

// 최대 리스트 크기가 5로 설정하였으므로 추가 불가
list.insert(item: 6)

// 현재 리스트 카운트 : 5
print(list.count())

for i in 0..<list.count() {
    print(list.get(index: i))
    // 1
    // 2
    // 3
    // 4
    // 5
}
```

Implementation

```
// 4번째 요소 삭제
list.remove(at: 3)

// 현재 리스트 카운트 : 4
print(list.count())

for i in 0..
```

References

[1] [Swift 자료구조 ch01] 자료구조 살펴보기 : <https://kor45cw.tistory.com/238>

[2] Swift로 자료구조, 알고리즘 공부하기 (3) - Array List : <https://kor45cw.tistory.com/3>

[3] Swift의 Array 완전 정복 - 01. 생성과 조작 : <https://soooprmx.com/archives/7045>

[4] (Swift 기초) 자료구조(Collection) - 배열(Array) (2) : https://m.blog.naver.com/PostView.nhn?blogId=el_vin&logNo=221019918085&categoryNo=10&proxyReferer=&proxyReferer=https:%2F%2Fwww.google.com%2F

[5] [Swift4] 컬렉션 타입 :: Array / Dictionary / 메소드 / Optional / nil : <https://yereol.tistory.com/9>

References

[6] Swift Language 배열 : <https://sodocumentation.net/ko/swift/topic/284/배열>

[7] Swift Collection Types : <https://www.hohyeonmoon.com/blog/swift-collection-types/>

[8] 10장 Swift 기초 8, 묶음(collection) 자료 구조, 배열(array) : <https://www.youtube.com/watch?v=UgMHDoFpN9M>

[9] Swift의 Sequence와 Collection에 대해 알아야 하는것들 : <https://academy.realm.io/kr/posts/try-swift-sorosh-khanlou-sequence-collection/>

[10] 자료구조: Linked List 대 ArrayList : <http://www.nextree.co.kr/p6506/>

Thank you!