[16]collection

1. Cellection 이란?

컬렉션이란, 우리말로 쉽게 말해서 자료구조입니다. 쉽게 말해서 다수의 데이터, 즉 데이터그룹을 의미합니다.
데이터 군(群)을 저장하는 하기 위해서 우리는 배열을 사용해왔는데요, 배열과 비슷한 구조에다가 다수의 데이터를 쉽게 처리할 수 있는 방법(method)을 제공하는 API입니다.
우리가 배웠던 배열이 아주 훌륭한 자료구조입니다. 하지만 더욱 훌륭한 자료구조 형을 JAVA에서는 많이 준비했습니다. 그리고 우리는 어려운 자료구조 형의 내부를 공부할 필요 없이 API document를 보면서 그냥 사용만 하면 됩니다.

무수히 많은 데이터를 어떤 형식으로 잘 정리하느냐에 따라 추후 데이터를 찾아서 사용 할 때 업무의 효율성이 높아 질 수 있습니다. 그래서 JAVA에서는 다양한 자료구조형를 제공 하고 있습니다. **다양한 자료구조 형이 제공되는 이유는 데이터의 성질에 따라서 데이터를 관리(정리)해야 하는 방식이 다르기 때문**입니다. 자료구조형 안에서는 객체의 레퍼런스 만을 관리합니다.

- ArrayList ; 배열과 매우 비슷. 인덱스가 존재하며 데이터는 중복을 허용. 인덱스가 가장 중요
    - add(객체)
    - add(index, 객체)
    - set(index, 객체)
    - get(index)
    - size()
    - remove(index)
    - remove(객체)
    - contains(객체) 주어진 객체가 저장되어 있는지 여부를 boolean 값으로 반환
    - isEmpty()
    - clear()

- LinkedList ; ArrayList와 거의 비슷. ArrayList는 접근시간(읽어오는데 걸리는) 시간은 빠르나 데이터를 추가하거나 삭제하는데 많은 데이터를 옮겨야 해서 시간이 많이 걸린다는 단점이 있다. 이점을 보완한 LinkedList. ArrayList와 달리 불연속적으로 존재하는 데이터를 연결. 순차적으로 데이터를 추가/삭제할 경우 ArrayList가 빠르고 비순차적으로 데이터를 추가/삭제하는 경우 LinkedList가 빠르다. 접근시간도 ArrayList가 빠르다

Map계열의 자료구조는 인덱스 대신 키 값으로 데이터를 액세스합니다.

List계열과 달리 인덱스가 없고, 키와 값만 있습니다. 그리고 키는 유니크 해야 합니다. 우리가 값

을 관리하고자 한다면 키를 이용해서 값을 관리할 수 있습니다.

- HaspMap

(3) Set 계열 Collection 클래스 살펴보기

Set계열 자료구조에서는 데이터의 순서는 없습니다(인덱스 없다). 하지만 중복된 데이터는 허락하지 않습니다.

중복된 데이터의 의미는 hashcode()값이 같거나 equal()메소드의 결과값에 의해 해석

<예제풀어보기>

```
1  package com.lec.ex05_quiz2;
2
3  public class Member {
4
5      private String name;
6      private String tel;
7      private String address;
8      public Member(String name, String tel, String address) {
9          super();
10         this.name = name;
11         this.tel = tel;
12         this.address = address;
13     }
14     @Override
15     public String toString() {
16         return name + " " + tel + " " + address;
17     }
18
19
```

```java
public static void main(String[] args) {
    ArrayList<Member> member = new ArrayList<Member>();
    Scanner scanner = new Scanner(System.in);
    String answer, name,  address, tel;

    do {
        System.out.print("회원가입을 하시겠습니까(Y/N) ");
        answer = scanner.next();
    if(answer.equalsIgnoreCase("n")) {
        break;
    }else if(answer.equalsIgnoreCase("y")) {
        System.out.print("성함을 입력하세요 : ");
        name = scanner.next();
        System.out.print("전화번호를 입력하세요 : ");
        tel = scanner.next();
        scanner.nextLine();
        System.out.print("주소를 입력하세요 : ");
        address = scanner.next();
        member.add(new Member(name, tel, address));
    }
    }while(true);
    if(member.isEmpty()) {
        System.out.println("****등록된 회원이 없습니다****");
    }else {
        for(Member temp : member) {
            System.out.println(temp);
        }
    }
}
```

```java
    public static void main(String[] args) {
        HashMap<String, Member> member = new HashMap<String, Member>();
        Scanner scanner = new Scanner(System.in);
        String answer, name, tel, address;
        do {
            System.out.print("회원 가입을 원하십니까 ? ");
            answer = scanner.next();
            if(answer.equalsIgnoreCase("n")) {
                break;
            }else if(answer.equalsIgnoreCase("y")) {
                System.out.print("성함을 입력해 주세요 : ");
                name = scanner.next();
                System.out.print("전화번호를 입력해 주세요 : ");
                tel = scanner.next();
                if(member.get(tel) != null) {
                    System.out.println("기존에 등록된 전화번호가 존재합니다");
                    continue;
                }
                System.out.print("주소를 입력해 주세요 : ");
                scanner.nextLine();
                address = scanner.nextLine();
                member.put(tel, new Member(name, tel, address));
            }
        }while(true);
        if(member.isEmpty()) {
            System.out.println("****등록된 회원이 없습니다****");
        }else {
            System.out.println("****등록된 회원 목록 ****");
            Iterator<String> iterator = member.keySet().iterator();
            while(iterator.hasNext()) {
                String key = iterator.next();
                System.out.println(member.get(key));
            }
        }
    }
}
```