해시맵(HashMap)

- description : HashMap
- author : 오션
- email : shlim@repia.com
- lastupdate : 2022-08-18 Thu

HashMap

Array: 생성 시 크기가 고정, 사용 중 크기 변경 불가
ArrayList :

```
List 인터페이스의 구현 클래스, ArrayList에 객체를 추가하면, 객체를 인덱스로 관리
저장 용량을 초과한 객체들이 들어오면 자동적으로 저장용량이 늘어남
저장할 객체 타입을 타입 파라미터로 표기, 기본 생성자를 호출
인덱스 번호(int type)로 아이템에 접근
```

HashMap :

```
Map 인터페이스를 구현한 대표적인 Map 컬렉션
아이템을 "key/value" 쌍으로 저장
String 타입같은 인덱스로 아이템에 접근
하나의 객체는 다른 value객체에 대한 키(인텍스)로 사용됨
```

메소드

- 객체 추가 : put() 메소드 사용
- 객체 접근 : get() 메소드 사용
- 객체 삭제 : remove() 메소드 사용
- 객체 모두 삭제 : clear() 메소드 사용
- 객체 크기 확인 : size() 메소드 사용

```
package sec01.hashMap;

import java.util.HashMap;

public class HashMapExample1 {

```

```
    public static void main(String[] args) {

        HashMap<String, String> capitalCities = new HashMap<String,
String>();

        // Add items
        capitalCities.put("England", "London");
        capitalCities.put("Germany", "Berlin");
        capitalCities.put("Norway", "Oslo");
        capitalCities.put("USA", "Washington DC");

        // To find out how many items there are, use the size() method
        System.out.println("There are " + capitalCities.size() + " items");
        // There are 4 items

        System.out.println("capitalCities : " + capitalCities);
        // capitalCities : {USA=Washington DC, Norway=Oslo, England=London,
Germany=Berlin}

        // Access an item
        // To access a value in the HashMap, use the get() method and refer
to its key
        System.out.println("England : " + capitalCities.get("England"));
        // England : London

        // to remove an item, use the remove() method and refer to the key
        capitalCities.remove("England");
        System.out.println("after removing England, capitalCities : " +
capitalCities);
        //after removing England, capitalCities : {USA=Washington DC,
Norway=Oslo, Germany=Berlin}

        // To remove all items, use the clear() method
        capitalCities.clear();
        System.out.println("after clearing, capitalCites : " +
capitalCities);
        // after clearing, capitalCites : {}
    }
}
```

keySet()

key 만을 원할 때 사용

```
package sec01.hashMap;

import java.util.HashMap;
```

```
public class HashMapExample_keySet {

    public static void main(String[] args) {

        HashMap<String, String> capitalCities = new HashMap<String,
String>();

        capitalCities.put("England", "London");
        capitalCities.put("Germany", "Berlin");
        capitalCities.put("Norway", "Oslo");
        capitalCities.put("USA", "Washington DC");

        System.out.println("capitalCities : " + capitalCities);

        // Loop through the items of a HashMap with a for-each loop
        // Use the keySet() method to get only the keys

        for (String i : capitalCities.keySet()) {
            System.out.println("capitalCities key : " + i);
            //capitalCities key : USA
            //capitalCities key : Norway
            //capitalCities key : England
            //capitalCities key : Germany
        }
    }
}
```

values()

값(value)만들 원할 때

```
package sec01.hashMap;

import java.util.HashMap;
import java.util.Map;

public class HashMapExample_Values {

    public static void main(String[] args) {

        HashMap<String, String> capitalCities = new HashMap<String,
String>();
        capitalCities.put("England", "London");
        capitalCities.put("Germany", "Berlin");
        capitalCities.put("Norway", "Oslo");
        capitalCities.put("USA", "Washington DC");
```

```
        // Use the values() method if you only want the values
        for (String i : capitalCities.values()) {
            System.out.println("capitalCities Values ::: " + i);
            //capitalCities Values ::: Washington DC
            //capitalCities Values ::: Oslo
            //capitalCities Values ::: London
            //capitalCities Values ::: Berlin
        }
    }
}
```

key와 value를 원할 때

```
package sec01.hashMap;

import java.util.HashMap;
import java.util.Map;

public class HashMapExample_keySet_and_values {

    public static void main(String[] args) {

        Map<String, String> capitalCities = new HashMap<String, String>();

        capitalCities.put("England", "London");
        capitalCities.put("Germany", "Berlin");
        capitalCities.put("Norway", "Oslo");
        capitalCities.put("USA", "Washingto DC");

        for (String keyAndVal : capitalCities.keySet()) {
            System.out.println("key : " + keyAndVal + "\t" + "value : " +
capitalCities.get(keyAndVal));
            //key : USA        value : Washingto DC
            //key : Norway        value : Oslo
            //key : England        value : London
            //key : Germany        value : Berlin
        }
    }
}
```

키와 값의 타입

기본 타입(primitive type: byte, short, int, float, double, boolean, char)을 사용할 수 없다.
클래스 및 인터페이스 타입만 사용할 수 있다. wrapper class를 사용해야 한다.

wrapper class: 기본 데이터 타입을 객체로 사용할 수 있는 방법을 제공

```
package sec01.hashMap;

import java.util.HashMap;

public class HashMap_other_types {

    public static void main(String[] args) {

        // people이라는 HashMap 객체를 생성
        HashMap<String, Integer> people = new HashMap<String, Integer>();

        // key와 value를 추가
        people.put("John", 32);
        people.put("Steve", 30);
        people.put("Angie", 33);

        for(String key : people.keySet()) {
            System.out.println("Name : " + key + "\t" + " Age : " +
people.get(key));
            //Name : Angie    Age : 33
            //Name : Steve    Age : 30
            //Name : John    Age : 32
        }
    }
}
```

Ref LInk

Java HashMap

오션,, HashMap,, 해시맵