6월 2일 복습&퀴즈

이태양


```java
public class HashSetTest {
    public static void main(String[] args) {
        HashSet<String> set = new HashSet<String>();

        set.add("우유");
        set.add("빵");
        set.add("베이컨");
        set.add("소시지");
        set.add("파스타");
    }
}
```

HashSet에 add하면 순서대로 되는 것이 아닌 무작위로 저장이 된다

[빵, 우유, 파스타, 베이컨, 소시지]


```
{2=Student{age=33, name='Chris'}, 3=Student{age=29, name='David'}, 7=Student{age=42, name='Bob'}, 44=Student{age=27, name='Denis'}}
{3=Student{age=29, name='David'}, 7=Student{age=42, name='Bob'}, 44=Student{age=27, name='Denis'}}
{3=Student{age=77, name='Jesica'}, 7=Student{age=42, name='Bob'}, 44=Student{age=27, name='Denis'}}
key = 3, value = Student{age=77, name='Jesica'}
key = 7, value = Student{age=42, name='Bob'}
key = 44, value = Student{age=27, name='Denis'}
key = 열쇠, value = 으아아앜!
```

```java
class Student {
    int age;
    String name;

    public Student (int age, String name) {
        this.age = age;
        this.name = name;
    }
    @Override
    public String toString() {
        return "Student{" +
                "age=" + age +
                ", name='" + name + '\'' +
                '}';
    }
}
public class HashMapTest {
    public static void main(String[] args) {
        // Map의 특성중 하나가 key와 value가 분리됨
        // Map<Key, Value>
        // 특별히 특정 데이터타입을 지켜줘야 하는 것은 없다.
        Map<Integer, Student> st = new HashMap<Integer, Student>();
        // 앞에 오는 숫자는 인덱스가 아니다.
        // 단지 사물함을 여는데 필요한 열쇠일 뿐
        st.put(7, new Student( age: 42,  name: "Bob"));
        st.put(2, new Student( age: 33,  name: "Chris"));
        st.put(44, new Student( age: 27,  name: "Denis"));
        st.put(3, new Student( age: 29,  name: "David"));
        System.out.println(st);

        st.remove( key: 2);

        System.out.println(st);

        st.put(3, new Student( age: 77,  name: "Jesica"));

        System.out.println(st);

        for (Map.Entry<Integer, Student> s : st.entrySet()) {
            Integer key = s.getKey();
            Student value = s.getValue();
            System.out.println("key = " + key + ", value = " + value);
        }

        // 나는 "열쇠"를 키로 사용하고 "으아아앜!"을 값으로 쓸거야! 하면 쓰면 된다.
        Map<String, String> strMap = new HashMap<String, String>();

        strMap.put("열쇠", "으아아앜!");

        // HashMap을 사용할때는 이 방식이 변하지 않습니다.
        // 추상화의 연장선 관점에서 아래 사항을 준수하여 코딩하면 어떤 상황에서든 key, value 값을 얻을 수 있습니다.
        // Entry<키 데이터타입, 밸류 데이터타입> 형식은 지켜주세요.
        for (Map.Entry<String, String> map : strMap.entrySet()) {
            String key = map.getKey();
            String value = map.getValue();
            System.out.println("key = " + key + ", value = " + value);
        }
    }
}
```

해시맵 사용법 같은 키값이면 덮어쓰기가 된다 put으로 저장가능


```java
public class HowToUseHashSet {
    public static void main(String[] args) {
        Set<String> s = new HashSet<String>();

        String[] sample = {"안녕", "하이", "헬로", "안녕", "안녕"};

        // 집합의 특성: 중복 허용 x
        for (String str : sample) {
            if (!s.add(str)) {
                System.out.println("중복되었습니다: " + str);
            }
        }

        // size()는 원소의 개수
        System.out.println(s.size() + " 중복을 제외한 단어: " + s);
    }
}
```

HashSet을 사용하고
중복체크까지!


```java
public class SetFeatureTestWithHashSet {
    public static void main(String[] args) {
        Set<String> s1 = new HashSet<String>();
        Set<String> s2 = new HashSet<String>();

        s1.add("Apple");
        s1.add("Tesla");
        s1.add("Microsoft");

        s2.add("Tesla");
        s2.add("Alphabet");
        s2.add("Texas Instruments");

        Set<String> union = new HashSet<String>(s1);
        union.addAll(s2);

        Set<String> intersection = new HashSet<String>(s1);
        intersection.retainAll(s2);

        System.out.println("합집합: " + union);
        System.out.println("교집합: " + intersection);
    }
}
```

Union과 intersection 사용
합집합과 교집합


```
int randNum[] = new int[4];
String comsculpture[] = new String[4];
int comrandNum[] = new int[4];
for (int i = 0; i < 4; i++) {
    sculpture[i] = pattern[(int)(Math.random() * 3)];
    comsculpture[i] = pattern[(int)(Math.random() * 3)];

    // 중복되는 숫자를 체킹하는 코드가 필요합니다:
    // 56번에서는 중복체킹하는것을 추가로 처리하여
    // 57번 문제를 풀어보도록 합시다!
   randNum[i] = (int)(Math.random() * 10);
   comrandNum[i] = (int)(Math.random() * 10);
   for(int j = 0; j<i; j++){
       if (sculpture[i] == sculpture[j] && randNum[i] == randNum[j]) {
           System.out.println((i+1)+"번째 카드에서 중복이 발생해서 다시 분배합니다");
           sculpture[i] = pattern[(int)(Math.random() * 3)];
           randNum[i] = (int)(Math.random() * 10);
       }
       if (comsculpture[i] == comsculpture[j] && comrandNum[i] == comrandNum[j]) {
           System.out.println((i+1)+"번째 카드에서 중복이 발생해서 다시 분배합니다");
           sculpture[i] = pattern[(int)(Math.random() * 3)];
           randNum[i] = (int)(Math.random() * 10);
       }

   }
    System.out.println("사용자에게 분배된 카드는 = "+ sculpture[i] +" 문양의 " + randNum[i] + " 카드입니다!");
    System.out.println("컴퓨터에게 분배된 카드는 = "+ comsculpture[i] +" 문양의 " + comrandNum[i] + " 카드입니다!");
```

중복제거한다고 어쩌다보니,, 배열로 하게되었다 아직 해시맵 사용이 익숙치않고 뭔가 개념도 덜잡혀 있는 것 같은 느낌 ,, 쉬운예제부터 차근차근 풀어보고싶습니다,,