Tutorial/Lab6 - Sets and Maps

Aim

This tutorial/lab aims to further the understanding of the topics covered in Week 7’s lecture. Meanwhile, it aims to get students familiar with the question types that might be appeared in their future test.

Exercise6.1 HashSet Methods

Suppose set s1 is [1, 2, 5] and set s2 is [2, 3, 6].

After s1.addAll(s2), s1 is __________

After s1.addAll(s2), s2 is __________

After s1.removeAll(s2), s1 is __________

After s1.retainAll(s2), s1 is __________

Exercise6 .2 Non-duplicated Set

Recall the question on Lecture Note page 12. What if we have the following Person class:

```
import java.util.Objects;

public class Person {
    private String name;
    private int age;
    public Person(String name, int age) {
        this.name = name;
        this.age = age;
    }
    @Override
    public String toString() {
        return "Person{" +
                "name='" + name + '\'' + ", age=" + age + '}';
    }
}
```

We are now conducting the following operation:

```
        Set<Person> set1 = new HashSet<>();
        set1.add(new Person("John",19));
        set1.add(new Person("Mary",20));
        set1.add(new Person("John",19));
        System.out.println(set1);
```

Question: What would be the printed result? How should we make set1 a non-duplicated set?

Exercise6.3 Methods in TreeSet

In the following code, what method should be filled in the blank if the expected printed results are 1.) “Blue” and “Red” respectively?

```
import java.util.*;

public class Test {
  public static void main(String[] args) throws Exception {
    TreeSet<String> set = new TreeSet<>();

    set.add("Red");
    set.add("Yellow");
    set.add("Green");
    set.add("Blue");

    System.out.println(set.______("Purple").first());
  }
}
```

Rewrite the MyStack class in Lecture 2.2 (page 43) to perform a deep copy of the list field.

Exercise6.4 Maps

Write a program that reads an unspecified number of integers and finds the one that has the most occurrences. The input ends when the input is **0**. For example, if you entered **2 3 40 3 5 4 –3 3 3 2 0**, the number **3** occurred most often. If not one but several numbers have the most occurrences, all of them should be reported. For example, since **9** and **3** appear twice in the list **9 30 3 9 3 2 4**, both occurrences should be reported.

Hints:

- Use a Map to Track Occurrences: Consider using a map data structure where each key is a unique integer from the input and each value is the count of occurrences of that integer.

- Update Counts in the Map: When reading each integer, update its corresponding count in the map. If the integer is not yet in the map, add it with a count of 1. If it is already in the map, increment its count.

- Find the Maximum Count: After all integers have been read, find the maximum count value in the map. This represents the most occurrences of any number.

- Identify Numbers with Maximum Occurrences: Iterate through the map to find all numbers that have the maximum count and print them.