Discrete Structures

Lab 1

Youssef Hany Fathy Shamsia – 18015025

Abdallah Yasser Ibrahim Abdelhamid – 18015026

GitHub

Problem Statement:

- Write a program that takes:
  1. An input a list of strings as a Universe.
  2. Then takes another input a number of sets (that are subsets of the universe)
  3. Then ask the user about the operations they want to perform

- 3 required features to be implemented in this assignment:
  1. Union of two sets
  2. Intersection of two sets
  3. Complement of a set

Used Data Structures:

➔ Java's ArrayList<String> to store set data.

Pseudo Code:

```
Union (ArrayList a, ArrayList b){
    ArrayList res

    for each string s in a:
        res.add (s)

    for each string s in b:
        res.add (s)

    removeDuplicates(res)
    return res
}
```

```
Intersection (ArrayList a, ArrayList b){
    ArrayList res

    for each string s in a:
        if b contains s
            res.add (s)

    removeDuplicates(res)

    return res
}
```

```
removeDuplicates (ArrayList list){
    ArrayList noDuplicates

    for each string s in list:
        if noDuplicates doesn’t contain s
            noDuplicates.add (s)

    return noDuplicates
}
```

```
complement (ArrayList list, universe){
    ArrayList res
    res.add (universe)

    for each string s in list:
        res.remove (s)

    return res
}
```

Code snippets:

```
ArrayList<String> union (ArrayList<String> list1, ArrayList<String> list2) {
    ArrayList<String> res = new ArrayList<String>();
    res.addAll(list1);
    res.addAll(list2);
    res = removeDuplicates (res);

    return res;
}

ArrayList<String> intersection (ArrayList<String> list1, ArrayList<String> list2) {
    ArrayList<String> res = new ArrayList<String>();

    for (String str : list1) {
        if (list2.contains(str))
            res.add(str);
    }

    return res;
}

ArrayList<String> complement (ArrayList<String> list) {
    ArrayList<String> res = new ArrayList<String>();
    res.addAll(universe);

    for (String str : list) {
        res.remove(str);
    }

    return res;
}
```

```
static ArrayList<String> removeDuplicates (ArrayList<String> list) {
    ArrayList<String> res = new ArrayList<String>();
    for (String str : list) {
        if (!res.contains(str))
            res.add(str);
    }

    return res;

}
```

```java
//Receive Universe as input
ArrayList<String> universe = new ArrayList<String>();
int subsetNum;
Scanner reader = new Scanner(System.in);
int flag = 0;
while (flag==0) {
    System.out.println("Input string:");
    universe.add(reader.next());
    while(true) {
        System.out.println("Do you want to add more to the universe? (y/n)");
        String input = reader.next();
        if(input.compareTo("n")==0) {
            flag=1;
            break;
        }
        else if (input.compareTo("y")!=0) {
            System.out.println("Invalid input. Please try again.");
        }
        else {
            break;
        }

    }

}
universe = SetUtils.removeDuplicates(universe);
```

Sample runs:

```
Do you want to add more to the universe? (y/n)
y
Input string:
5
Do you want to add more to the universe? (y/n)
y
Input string:
5
Do you want to add more to the universe? (y/n)
n
The universe contains:
1- 1
2- 2
3- 3
4- 4
5- 5
```

```
Please enter the number of required sets:
3
Please select index from universe to place in set
Do you want to add more to the set? (y/n)
y
Input index:
1
Do you want to add more to the set? (y/n)
y
Input index:
2
Do you want to add more to the set? (y/n)
y
Input index:
3
```

Sets

```
==========================
Please select an operation:
==========================
1-Union of two sets
2-Intersection of two sets
3-Complement of a set
4-Print universe
5-Print specific set
6-Exit
5
Please select the required set:
1
The specified set contains:
1- 1
2- 2
3- 3
4- 4
5- 5
```

```
==========================
Please select an operation:
==========================
1-Union of two sets
2-Intersection of two sets
3-Complement of a set
4-Print universe
5-Print specific set
6-Exit
5
Please select the required set:
2
The specified set contains:
1- 1
2- 3
3- 5
```

```
==========================
Please select an operation:
==========================
1-Union of two sets
2-Intersection of two sets
3-Complement of a set
4-Print universe
5-Print specific set
6-Exit
5
Please select the required set:
3
The specified set contains:
1- 2
2- 4
```

Operations

```
==========================
Please select an operation:
==========================
1-Union of two sets
2-Intersection of two sets
3-Complement of a set
4-Print universe
5-Print specific set
6-Exit
3
Please select the required set:
2
The complement of the set contains:
1- 2
2- 4
```

```
==========================
Please select an operation:
==========================
1-Union of two sets
2-Intersection of two sets
3-Complement of a set
4-Print universe
5-Print specific set
6-Exit
2
Please select first required set:
1
Please select second required set:
2
The intersection of the two sets contains:
1- 1
2- 3
3- 5
```

```
==========================
Please select an operation:
==========================
1-Union of two sets
2-Intersection of two sets
3-Complement of a set
4-Print universe
5-Print specific set
6-Exit
2
Please select first required set:
2
Please select second required set:
3
The intersection of the two sets contains:

--------------------------
```

```
==========================
Please select an operation:
==========================
1-Union of two sets
2-Intersection of two sets
3-Complement of a set
4-Print universe
5-Print specific set
6-Exit
1
Please select first required set:
1
Please select second required set:
2
The union of the two sets contains:
1- 1
2- 2
3- 3
4- 4
5- 5
```

Assumptions:

- It’s permitted to use dynamic arrays (ArrayList<String>), as they don’t have built-in set functionalities.