

Student 1 Name: Muhammad Ibrahim

Elkotb Abdel-Aziz Kotb

Student 1 ID : 19016258

Student 2 Name: Amr Magdy Mahmoud

Hanafy

Student 2 ID: 19016116

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):***
 - Union of two sets***
 - Intersection of two sets***
 - Complement of a set***

Used Data Structures:

<i>Data Structure</i>	
<i>Arrays of Strings String[]</i>	<i>Used to Parse Input Strings and Preform operations on them.</i>
<i>List of Strings List<String></i>	<i>Used in Adding and Removing Strings using List Interface.</i>
<i>ArrayList of Strings ArrayList<String[]></i>	<i>Used to Store Input Sets into a List .</i>

The Program is Implemented in Java 16 using IntelliJ IDEA as an IDE.

Structure of the Program:

Problem

- + universe: String[]
- + sets: ArrayList<String[]>
- + size: int

- + Problem(): void
- + unique(): String[]
- + isSubset(): boolean
- + addSets(): void
- + editSet(): void
- + printSetMenu(): void
- + printSet(): void
- + list2StringList(): String[]
- + removeSet(): void
- + readUniverse(): void
- + Union(): String[]
- + Intersection(): String[]
- + Complement(): String[]
- + main(): void

Fundamental Methods:

1. Union(String[] set1, String[] set2)

Method returns a union of two given sets by looping over the two sets and adding different elements into the output.

```
/**
 * Method to return the Union of Two Sets
 * By Looping Over The Two Sets
 * @param set1
 * @param set2
 * @return String[] Array of Strings Contains The Union of set1 and set2
 */
public String[] Union(String[] set1, String[] set2)
{
    List<String> union_list = new ArrayList<>();
    for(int i = 0; i < set1.length; i++){
        union_list.add(set1[i]);
    }
    for(int i = 0; i < set2.length; i++){
        boolean found = false;
        for(int j = 0; j < union_list.size(); j++){
            if(union_list.get(j).equals(set2[i])){
                found = true;
                break;
            }
        }
        if(!found){
            union_list.add(set2[i]);
        }
    }
    return list2StringList(union_list);
}
```

2. *Intersection(String[] set1, String[] set2)*

Method returns an intersection of two given sets by looping over the two sets and adding the element that is contained in both sets

```
/**
 * Method to return the Intersection of Two Sets
 * By Looping Over The Two Sets
 * @param set1
 * @param set2
 * @return String[] Array of Strings Contains The Intersection of set1 and set2
 */
public String[] Intersection(String[] set1, String[] set2)
{
    List<String> intersection_list = new ArrayList<>();

    for(int i = 0; i < set1.length; i++){
        for(int j = 0; j < set2.length; j++){
            if(set1[i].equals(set2[j])){
                intersection_list.add(set1[i]);
            }
        }
    }
    return list2StringList(intersection_list);
}
```

3. *Complement(String[] set, String[] Universe)*

Method returns the complement of Set by looping over the set and the universe and adding to output the element that is not contained in set and contained in Universe

```
/**
 * Method to return The Complement of A Set
 * By Looping Over The Set and Universe
 * @param set1
 * @return String[] Array of Strings Contains Complement set1
 */
public String[] Complement(String[] set1, String[] Universe)
{
    List<String> complement_list = new ArrayList<>();

    for(int i = 0; i < Universe.length; i++){
        boolean found = false;
        for(int j = 0; j < set1.length; j++){
            if(Universe[i].equals(set1[j])){
                found = true;
                break;
            }
        }
        if(!found){
            complement_list.add(Universe[i]);
        }
    }
    return list2StringList(complement_list);
}
```

Helping Methods:

1. Unique(String[] set)

Method that removes the repeated elements from a Set by Looping over the Set in nested loop flag repeated indexes of elements.

```
/**
 * Method to Remove Repeated Element from a Set
 * @param set The Set to be Inspected
 * @return String[] The Set after removing Repeated Elements
 */

public String[] unique(String[] set){
    List<String> unique_list = new ArrayList<>();
    unique_list.add(set[0]);
    for(int i = 0; i < set.length; i++){
        boolean found = false;
        for(int j = 0; j < unique_list.size(); j++){
            if(unique_list.get(j).equals(set[i])){
                found = true;
            }
        }
        if(!found){
            unique_list.add(set[i]);
        }
    }
    return list2StringList(unique_list);
}
```


2. *isSubset(String[] Universe, String[] set)*

Method that returns true if the Given Set is a subset of the Universe by making a flag if an element that is in set and not in Universe.

```
/**
 * Method to Ensure that a Specified Set is a Subset of the Given Universe
 * @param Universe
 * @param set
 * @return true if the set is a subset of Universe
 * @return false if the set is not a subset of Universe
 */

public boolean isSubset(String[] Universe, String[] set)
{
    for(int i = 0; i < set.length; i++){
        boolean notSubset = true;
        for(int j = 0; j < Universe.length; j++){
            if(Universe[j].equals(set[i])){
                notSubset = false;
            }
        }
        if(notSubset) {
            return false;
        }
    }
    return true;
}
```

3. *list2StringList(List<String> string_list)*

Method that converts given List of Strings into an Array of Strings

```
/**
 * Method to Convert from List<String> Object to Array of Strings
 * @param string_list List of Strings
 * @return String[] Array of Strings
 */
public static String[] list2StringList(List<String> string_list){
    String[] string = new String[string_list.size()];
    for(int i = 0; i < string_list.size(); i++){
        string[i] = string_list.get(i);
    }
    return string;
}
```

4. *printSet(String[] set)*

Method that Prints a given Array of Strings to console

```
/**
 * Method to Print Array of Strings to the Console
 * Uses for Loop
 * @param set Array of Strings to Printed
 */
public void printSet(String[] set)
{
    System.out.print("{ ");
    for(int i = 0; i < set.length; i++){
        if(i == set.length - 1)
            System.out.print(set[i]);
        else
            System.out.print(set[i] + ", ");
    }
    System.out.println(" }");
}
```

5. *addSets()*

Method:

- *Asks the User for the Number of Sets to be Added*
- *The User Enters elements each separated by space*
- *then the Method calls `isSubset(String[] set)` to Ensure that the given set is a subset*
- *then if it is subset then it calls `unique(String[] set)` and add it to the List of Sets of the Object*
- *if it is not a subset it asks again for a set that is a subset*

```
/**
 * Method to Add New Sets into the Object to Preform Operations on them
 * Before Adding a Certain Set It-->>
 * 1) Removes Repeated elements
 * 2) Makes Sure it is a Subset of the Universe
 */

public void addSets() {
    System.out.println("Enter The Number Of Sets To Be Added");
    Scanner in = new Scanner(System.in);
    int n = Integer.parseInt(in.nextLine());
    System.out.println("Enter Elements of Each Set Seperated by Space");
    for(int i = 0; i < n; i++) {
        System.out.println("Enter The #" + Integer.sum(i, 1) + " Set:");
        String input = in.nextLine();
        if(!isSubset(this.universe, input.split( regex: " "))) {
            System.out.println("Please Enter a Subset of the Universe ");
            i -= 1;
            continue;
        }
        this.sets.add(unique(input.split( regex: " ")));
        this.size++;
    }
    System.out.println(n + " Sets have been added successfully!");
}
```

6. *readUniverse()*

Method that reads the Universe from the Console. The User Should input Each element separated by space.

```
/**
 * Method to Read the Universe from User on Console
 * Reads the Input as A String then Parses to Array of Strings
 */

public void readUniverse()
{
    System.out.println("Enter The Universe (Enter Each Element Separated by Space and Press Enter):");
    Scanner in = new Scanner(System.in);
    String input = in.nextLine();
    this.universe = unique(input.split(regex: " "));
}
```

Additional Methods:

- 1. editSet ()***
- 2. removeSet()***
- 3. printSetMenu()***

Pseudocode for main() method:

1. Create an **Instant of Class Problem** and call it **problem**
2. Call **problem.readUniverse()**
3. Call **problem.addSets()**
4. Wait for User input **(1 -> 8)**
5. If **input = 1** then asks input of index1 and index2
6. Print **Intersection** of sets(index1) and sets(index2)
7. If index1 or index2 **out of bounds go to line 4**
8. If **input = 2** then asks input of index1 and index2
9. Print **Union** of sets(index1) and sets(index2)
10. If index1 or index2 **out of bounds go to line 4**
11. If **input = 3** then asks input of index
12. Print **Complement** of sets(index)
13. If index **out of bounds go to line 4**
14. If **input = 4**
15. Go to line 3
16. If **input = 5**
17. Call **problem.editSet()**
18. If **input = 6**

19. Call **problem.printSetMenu()**
20. If **input = 7**
21. Call **problem.removeSet()**
22. If **input = 8** then **Exit Program**

```
public static void main(String[] args) {  
  
    try {  
        Problem problem = new Problem();  
        problem.readUniverse();  
        problem.addSets();  
        Scanner in = new Scanner(System.in);  
        while (true) {  
            System.out.println("=====");  
            System.out.print("Universe ==> ");  
            problem.printSet(problem.universe);  
            for(int i = 0; i < problem.size; i++){  
                System.out.print(i + 1 + " ");  
                problem.printSet(problem.sets.get(i));  
            }  
            System.out.println("=====");  
            System.out.println( "[1] Find Intersection Of 2 Sets\n" +  
                "[2] Find Union Of 2 Sets\n" +  
                "[3] Find The Complement Of A Set\n" +  
                "[4] Add Sets\n" +  
                "[5] Edit A Set\n" +  
                "[6] Print One Set\n" +  
                "[7] Remove A Set\n" +  
                "[8] Exit");  
            System.out.print("Choose A Value :::--> ");  
            int current = Integer.parseInt(in.nextLine());  
            int a, b;  
            switch(current) {  
                case 4:  
                    problem.addSets();  
                    break;  
                case 5:  
                    problem.editSet();  
                    break;  
                case 1:  
                    System.out.print("Enter The Index Of The First Set: ");  
                    a = Integer.parseInt(in.nextLine())-1;  
                    System.out.print("Enter The Index Of The Second Set: ");  
                    b = Integer.parseInt(in.nextLine())-1;  
                    if(a <= problem.size && b <= problem.size){  
                        System.out.println("=====");  
                        System.out.print("====>> Their Intersection Is : ");  
                        problem.printSet(problem.Intersection(problem.sets.get(a), problem.sets.get(b)));  
                        System.out.println("Please Press Enter to Procced ");  
                        in.nextLine();  
                    }  
                    else{  
                        System.out.println("=====");  
                        System.out.println("Given Index Is Out Of Bounds");  
                        System.out.println("=====");  
                        System.out.println("Please Press Enter to Procced ");  
                        in.nextLine();  
                        continue;  
                    }  
                }  
            }  
            break;  
        }  
    }  
}
```

```

        case 2:
            System.out.print("Enter The Index Of The First Set: ");
            a = Integer.parseInt(in.nextLine())-1;
            System.out.print("Enter The Index Of The Second Set: ");
            b = Integer.parseInt(in.nextLine())-1;
            if(a <= problem.size && b <= problem.size){
                System.out.println("=====");
                System.out.print("====>> Their Union Is : ");
                problem.printSet(problem.Union(problem.sets.get(a), problem.sets.get(b)));
                System.out.println("Please Press Enter to Procced ");
                in.nextLine();
            }
            else{
                System.out.println("=====");
                System.out.println("Given Index Is Out Of Bounds");
                System.out.println("=====");
                System.out.println("Please Press Enter to Procced ");
                in.nextLine();
                continue;
            }
            break;
        case 3:
            System.out.print("Enter The Index Of The Set: ");
            a=Integer.parseInt(in.nextLine())-1;
            if(a <= problem.size){
                System.out.println("=====");
                System.out.print("====>> Their Complement Is : ");
                problem.printSet(problem.Complement(problem.sets.get(a), problem.universe));
                System.out.println("Please Press Enter to Procced ");
                in.nextLine();
            }
            else{
                System.out.println("=====");
                System.out.println("Given Index(s) Is Out Of Bounds");
                System.out.println("=====");
                System.out.println("Please Press Enter to Procced ");
                in.nextLine();
                continue;
            }
            break;
        case 6:
            problem.printSetMenu();
            System.out.println("Please Press Enter to Procced ");
            in.nextLine();
            break;
        case 7:
            problem.removeSet();
            System.out.println("Please Press Enter to Procced ");
            in.nextLine();
            break;
        case 8:
            System.exit( status: 0);
            break;
        default:
            break;
    }
}

}

}

catch (Exception e){
    System.out.println("Error " + e);
}

}

```

How to Operate the Program:

- 1. Enter the Elements of the Universe Each element separated by Space.***
- 2. Enter Number of Sets to be Added.***
- 3. Enter Each Element of each set separated by Space.***

```
Enter The Universe (Enter Each Element Seperated by Space and Press Enter):
A B F KKTR E R C V QWE TG A
Enter The Number Of Sets To Be Added
4
Enter Elements of Each Set Seperated by Space
Enter The #1 Set:
KKTR A F
Enter The #2 Set:
E R C V TG
Enter The #3 Set:
B F R C E
Enter The #4 Set:
A B F
4 Sets have been added successfully!
```

- 4. The Universe and the Entered Sets will be Printed on Console for Reference.***
- 5. The User Should Choose a Value to Procceed and Preform Operations.***

```
=====
Universe ==>> { A, B, F, KKTR, E, R, C, V, QWE, TG }
1) { KKTR, A, F }
2) { E, R, C, V, TG }
3) { B, F, R, C, E }
4) { A, B, F }
```

```
=====
[1] Find Intersection Of 2 Sets
[2] Find Union Of 2 Sets
[3] Find The Complement Of A Set
[4] Add Sets
[5] Edit A Set
[6] Print One Set
[7] Remove A Set
[8] Exit
Choose A Value :::--)
```


6. If user chose [1], The Program asks for the indexes of the Sets to calculate their Intersection, then press Enter to Proceed.

```
=====
Universe ==>> { A, B, F, KKTR, E, R, C, V, QWE, TG }
1) { KKTR, A, F }
2) { E, R, C, V, TG }
3) { B, F, R, C, E }
4) { A, B, F }
=====
[1] Find Intersection Of 2 Sets
[2] Find Union Of 2 Sets
[3] Find The Complement Of A Set
[4] Add Sets
[5] Edit A Set
[6] Print One Set
[7] Remove A Set
[8] Exit
Choose A Value :::--> 1
Enter The Index Of The First Set: 1
Enter The Index Of The Second Set: 4
=====
==>>> Their Intersection Is : { A, F }
Please Press Enter to Proceed
```

7. If User chose [2] then the same process will happen with Union.

8. If User chose [3] then the Program asks for the Index of the Set to calculate its complement, then press enter to proceed.

```
=====
Universe ==>> { A, B, F, KKTR, E, R, C, V, QWE, TG }
1) { KKTR, A, F }
2) { E, R, C, V, TG }
3) { B, F, R, C, E }
4) { A, B, F }
=====
[1] Find Intersection Of 2 Sets
[2] Find Union Of 2 Sets
[3] Find The Complement Of A Set
[4] Add Sets
[5] Edit A Set
[6] Print One Set
[7] Remove A Set
[8] Exit
Choose A Value :::--> 3
Enter The Index Of The Set: 2
=====
==>>> Their Complement Is : { A, B, F, KKTR, QWE }
Please Press Enter to Proceed
```

9. If the user chose [4] then the program will ask the number of sets to be added, then the user should enter them each element of each set separated by space

```
=====
[1] Find Intersection Of 2 Sets
[2] Find Union Of 2 Sets
[3] Find The Complement Of A Set
[4] Add Sets
[5] Edit A Set
[6] Print One Set
[7] Remove A Set
[8] Exit
Choose A Value :::--> 4
Enter The Number Of Sets To Be Added
2
Enter Elements of Each Set Seperated by Space
Enter The #1 Set:
KKRE R C V
Enter The #2 Set:
TG A B
2 Sets have been added successfully!
=====
Universe ==>> { A, B, F, KKRE, E, R, C, V, QWE, TG }
1) { A, B, KKRE }
2) { E, R, C, V }
3) { QWE, TG }
4) { A, B, F }
5) { KKRE, R, C, V }
6) { TG, A, B }
=====
```

10. If the user wants to remove a set choose [7] then Enter the index of set to be removed.

11. If the user wants to edit a set choose[5] then Enter the index of set to be edited.

12. If the user wants to print a set choose [6] then Enter the index of set to be printed.

Sample Runs:

1.)

```
Enter The Universe (Enter Each Element Seperated by Space and Press Enter):
ahmed omar amr mohammed omar ali
Enter The Number Of Sets To Be Added
a3
Please make sure to enter a valid number !
3
Enter Elements of Each Set Seperated by Space
Enter The #1 Set:
ahmed ali omar
Enter The #2 Set:
amr mohammed ali
Enter The #3 Set:
ahmed amr omar amr
3 Sets have been added successfully!
=====
Universe ==>> { ahmed, omar, amr, mohammed, ali }
1) { ahmed, ali, omar }
2) { amr, mohammed, ali }
3) { ahmed, amr, omar }
=====
[1] Find Intersection Of 2 Sets
[2] Find Union Of 2 Sets
[3] Find The Complement Of A Set
[4] Add Sets
[5] Edit A Set
[6] Print One Set
[7] Remove A Set
[8] Exit[
Choose A Value :::
```

```
Universe ==>> { ahmed, omar, amr, mohammed, ali }
1) { ahmed, ali, omar }
2) { amr, mohammed, ali }
3) { ahmed, amr, omar }
=====
[1] Find Intersection Of 2 Sets
[2] Find Union Of 2 Sets
[3] Find The Complement Of A Set
[4] Add Sets
[5] Edit A Set
[6] Print One Set
[7] Remove A Set
[8] Exit
Choose A Value ::: 1
Enter The Index Of The First Set: 1
Enter The Index Of The Second Set: 2
=====
[==>>> Their Intersection Is : { ali }
```

```
Universe ==>> { ahmed, omar, amr, mohammed, ali }
```

```
1) { ahmed, ali, omar }
```

```
2) { amr, mohammed, ali }
```

```
3) { ahmed, amr, omar }
```

```
=====
```

```
[1] Find Intersection Of 2 Sets
```

```
[2] Find Union Of 2 Sets
```

```
[3] Find The Complement Of A Set
```

```
[4] Add Sets
```

```
[5] Edit A Set
```

```
[6] Print One Set
```

```
[7] Remove A Set
```

```
[8] Exit
```

```
Choose A Value ::: 1
```

```
Enter The Index Of The First Set: 1
```

```
Enter The Index Of The Second Set: 3
```

```
=====
```

```
==>>> Their Intersection Is : { ahmed, omar }
```

```
Universe ==>> { ahmed, omar, amr, mohammed, ali }
```

```
1) { ahmed, ali, omar }
```

```
2) { amr, mohammed, ali }
```

```
3) { ahmed, amr, omar }
```

```
=====
```

```
[1] Find Intersection Of 2 Sets
```

```
[2] Find Union Of 2 Sets
```

```
[3] Find The Complement Of A Set
```

```
[4] Add Sets
```

```
[5] Edit A Set
```

```
[6] Print One Set
```

```
[7] Remove A Set
```

```
[8] Exit
```

```
Choose A Value ::: 2
```

```
Enter The Index Of The First Set: 2
```

```
Enter The Index Of The Second Set: 3
```

```
=====
```

```
==>>> Their Union Is : { amr, mohammed, ali, ahmed, omar }
```

```
Please Press Enter to Procced
```

```

Universe ==>> { ahmed, omar, amr, mohammed, ali }
1) { ahmed, ali, omar }
2) { amr, mohammed, ali }
3) { ahmed, amr, omar }
=====
[1] Find Intersection Of 2 Sets
[2] Find Union Of 2 Sets
[3] Find The Complement Of A Set
[4] Add Sets
[5] Edit A Set
[6] Print One Set
[7] Remove A Set
[8] Exit
Choose A Value ::: 3
Enter The Index Of The Set: 1
{
=====
==>>> Their Complement Is : { amr, mohammed }
Please Press Enter to Procced

```

2.)

```

Enter The Universe (Enter Each Element Seperated by Space and Press Enter):
A R G C V I G Y A W V G
Enter The Number Of Sets To Be Added
4
Enter Elements of Each Set Seperated by Space
Enter The #1 Set:
A R C G T
Please Enter a Subset of the Universe
Enter The #1 Set:
A R G C I
Enter The #2 Set:
G W V A
Enter The #3 Set:
R G Y
Enter The #4 Set:
A R W V G
4 Sets have been added successfully!
=====

```

```

Universe ==>> { A, R, G, C, V, I, Y, W }
1) { A, R, G, C, I }
2) { G, W, V, A }
3) { R, G, Y }
4) { A, R, W, V, G }
=====
[1] Find Intersection Of 2 Sets
[2] Find Union Of 2 Sets
[3] Find The Complement Of A Set
[4] Add Sets
[5] Edit A Set
[6] Print One Set
[7] Remove A Set
[8] Exit
Choose A Value ::: 1
Enter The Index Of The First Set: 1
Enter The Index Of The Second Set: 4
=====
==>>> Their Intersection Is : { A, R, G }
Please Press Enter to Procced

```

```

Universe ==>> { A, R, G, C, V, I, Y, W }
1) { A, R, G, C, I }
2) { G, W, V, A }
3) { R, G, Y }
4) { A, R, W, V, G }
=====
[1] Find Intersection Of 2 Sets
[2] Find Union Of 2 Sets
[3] Find The Complement Of A Set
[4] Add Sets
[5] Edit A Set
[6] Print One Set
[7] Remove A Set
[8] Exit
Choose A Value ::: 7
Enter The Index Of Sets To Be Removed
2
{ G, W, V, A }
Has Been Removed
Please Press Enter to Procced

```



```
Universe ==>> { A, R, G, C, V, I, Y, W }
```

```
1) { A, R, G, C, I }
```

```
2) { R, G, Y }
```

```
3) { A, R, W, V, G }
```

```
=====
```

```
[1] Find Intersection Of 2 Sets
```

```
[2] Find Union Of 2 Sets
```

```
[3] Find The Complement Of A Set
```

```
[4] Add Sets
```

```
[5] Edit A Set
```

```
[6] Print One Set
```

```
[7] Remove A Set
```

```
[8] Exit
```

```
Choose A Value ::: 5
```

```
Enter The Index Of Set To Be Edited
```

```
1
```

```
Enter the New Set
```

```
A R G V Y W
```

```
The #1 Set Has Been Updated Successfully!
```

```
=====
```

```
=====
```

```
Universe ==>> { A, R, G, C, V, I, Y, W }
```

```
1) { A, R, G, V, Y, W }
```

```
2) { R, G, Y }
```

```
3) { A, R, W, V, G }
```

```
=====
```

```
[1] Find Intersection Of 2 Sets
```

```
[2] Find Union Of 2 Sets
```

```
[3] Find The Complement Of A Set
```

```
[4] Add Sets
```

```
[5] Edit A Set
```

```
[6] Print One Set
```

```
[7] Remove A Set
```

```
[8] Exit
```

```
Choose A Value ::: 3
```

```
Enter The Index Of The Set: 2
```

```
=====
```

```
==>>> Their Complement Is : { A, C, V, I, W }
```

```
Please Press Enter to Procced
```

```
=====
```

```
=====
```

```
Universe ==>> { A, R, G, C, V, I, Y, W }
```

```
1) { A, R, G, V, Y, W }
```

```
2) { R, G, Y }
```

```
3) { A, R, W, V, G }
```

```
=====
```

```
[1] Find Intersection Of 2 Sets
```

```
[2] Find Union Of 2 Sets
```

```
[3] Find The Complement Of A Set
```

```
[4] Add Sets
```

```
[5] Edit A Set
```

```
[6] Print One Set
```

```
[7] Remove A Set
```

```
[8] Exit
```

```
Choose A Value ::: 2
```

```
Enter The Index Of The First Set: 1
```

```
Enter The Index Of The Second Set: 3
```

```
=====
```

```
==>>> Their Union Is : { A, R, G, V, Y, W }
```

```
Please Press Enter to Procced
```

Assumptions taken:

- ***Each Element Entered in any Set can only be String and Cannot be Spaces.***
- ***The Sets are 1-Indexed.***
- ***The order of Elements doesn't matter.***
- ***Repeated Elements are removed.***