**Aim: Write a Java program using Set interface containing list of items and perform the following operations:**

1. **Add items in the set.**
2. **Insert items of one set in to other set.**
3. **Remove items from the set.**
4. **Search the specified item in the set.**
5. **Get total number of elements in a Set.**
6. **Check if a Set is empty.**
7. **Subset operation.**
8. **Union operation.**
9. **Intersection operation.**

**Code-**

**package** Interface\_program;

**import** java.util.\*;

**public** **class** set\_interface {

**public** **static** **void** main(String args[])

{

Set<String> names = **new** LinkedHashSet<>();

names.add("Tom");

names.add("Mary");

names.add("Neha");

System.***out***.println("Element of names set : "+ names);

System.***out***.println();

List<String> listString = Arrays.*asList*("abc","pqr","xyz","Tom","Mary","Neha");

System.***out***.println("List elements : " +listString);

Set<String> uniqueString = **new** HashSet<>(listString);

System.***out***.println("Elements of uniqueString set : "+uniqueString);

System.***out***.println();

//Removing an element from a Set

**if** (names.remove("Mary")) {

System.***out***.println("Marry is removed");

}

**else**

{

System.***out***.println("Element not found");

}

System.***out***.println("Element of names set after remove : "+ names);

//Check if a Set is empty:

System.***out***.println();

**if** (names.isEmpty())

{

System.***out***.println("The set is empty");

}

**else**

{

System.***out***.println("The set is not empty");

}

//Get total number of elements in a Set:

System.***out***.println();

System.***out***.println("The uniqueString set has %d elements"+ uniqueString.size());

//Searching for an element in a Set

System.***out***.println();

**if** (names.contains("Tom"))

{

System.***out***.println("Found Tom");

}

**else**

{ System.***out***.println("Not Found Tom");

}

// Subset operation:s1.containsAll(s2) returns true if s2is a subset of s1

//(s2 is a subset of s1 if s1 contains all of the elements in s2).

**if** (uniqueString.containsAll(names)) {

System.***out***.println("names is a subset of uniqueString");

}

**else** {

System.***out***.println("names is not a subset of uniqueString");

}

//Union operation:

//s1.addAll(s2) — transforms s1 into the union of s1 and s2. (The union of two sets is the set //containing all of the elements contained in either set.)

System.***out***.println("names before union: " + names);

names.addAll(uniqueString);

System.***out***.println("names after union: " + names);

//Intersection operation:

//s1.retainAll(s2) — transforms s1 into the intersection of s1 and s2. (The //intersection of two //sets is the set containing only the elements common to both sets.)

System.***out***.println("names before intersection: " + names);

names.retainAll(uniqueString);

System.***out***.println("names after intersection: " + names);

}

}