PRACTICAL 3 – SET INTERFACE (ADVANCED JAVA)

AIM

To study and implement Set Interface in Java and perform traversal, insertion, deletion, and search operations.

SOFTWARE REQUIREMENTS

- Operating System: Windows / Linux
- JDK: 7 or 8
- IDE: NetBeans IDE 7.x

PART A: Set Traversal using Iterator (Forward & Reverse)

Set does not support indexing. To traverse in reverse order, elements are first converted into a List.

SOURCE CODE

```
package setdemo;

import java.util.*;

public class SetIteratorDemo {
    public static void main(String[] args) {

        Set<String> items = new HashSet<String>();
        items.add("Java");
        items.add("Python");
        items.add("C++");
        items.add("Advanced Java");

        System.out.println("Forward Traversal:");
        Iterator<String> itr = items.iterator();
        while (itr.hasNext()) {
            System.out.println(itr.next());
        }

        System.out.println("Reverse Traversal:");
        List<String> list = new ArrayList<String>(items);
        ListIterator<String> listItr = list.listIterator(list.size());

        while (listItr.hasPrevious()) {
            System.out.println(listItr.previous());
        }
    }
}
```

OUTPUT

```
Forward Traversal:
Java
Python
```

```
C++
Advanced Java

Reverse Traversal:
Advanced Java
C++
Python
Java
```

NETBEANS 7 EXECUTION STEPS

1. Open NetBeans IDE 7
2. File → New Project → Java → Java Application
3. Project Name: SetDemo
4. Uncheck 'Create Main Class'
5. Finish
6. Right-click Source Packages → New → Java Package
7. Package Name: setdemo
8. Right-click package → New → Java Class
9. Class Name: SetIteratorDemo
10. Paste code and press Shift + F6

PART B: Set Operations

This program demonstrates add, merge, remove, and search operations on Set.

SOURCE CODE

```
package setdemo;

import java.util.*;

public class SetOperationsDemo {
    public static void main(String[] args) {

        Set<String> set1 = new HashSet<String>();
        set1.add("Apple");
        set1.add("Banana");
        set1.add("Mango");

        Set<String> set2 = new HashSet<String>();
        set2.add("Orange");
        set2.add("Grapes");

        set1.addAll(set2);

        set1.remove("Banana");

        if (set1.contains("Apple")) {
            System.out.println("Apple is found in the set");
        }

        System.out.println("Final Set:");
        for (String item : set1) {
            System.out.println(item);
        }
    }
}
```

OUTPUT

```
Apple is found in the set
Final Set:
Apple
Mango
Orange
Grapes
```

RESULT

Thus, Set Interface was successfully implemented and all operations were performed.