

The screenshot displays an IDE with two main panels. The left panel shows the source code for a Java class named `Array_List`. The code implements an ArrayList using the non-generic approach. It includes imports for `java.util.ArrayList` and `System.out`. The `main` method creates an `ArrayList` object, prints its initial state (empty, size 0), and then adds several elements: "VIJAY", "ROLL NO", 56, and 5.6. It also demonstrates the `add` method with an index, `isEmpty`, `set` (replacing "HI" with "IIPS"), `size`, `contains` (checking for "AJAY" and "VIJAY"), and `get` (retrieving the element at index 3).

```
1 //IMPLEMENTING ARRAYLIST PROGRAM (NON GENERIC APPROACH)
2 import java.util.ArrayList;
3 public class Array_List
4 {
5     public static void main(String[] args)
6     {
7         ArrayList arr_obj = new ArrayList();//CREATING OBJECT OF ARRAY LIST
8         System.out.println("IMPLEMENTING ARRAYLIST PROGRAM (NON GENERIC APPROACH)");
9         System.out.println(arr_obj.isEmpty());//THIS METHOD CHECKS IS IT EMPTY AND RETURNS BOOLEAN VALUE
10        System.out.println(arr_obj.size());//RETURNS INTEGER COUNT OF NUMBER OF ELEMENTS IN ARRAYLIST
11
12        arr_obj.add("VIJAY");// ADD() METHOD ADD;s ELEMENT IN THE ARRAY LIST
13        arr_obj.add("ROLL NO");
14        arr_obj.add(56);
15        arr_obj.add(5.6);
16
17        arr_obj.add(0, "HI");
18        //ADD ( INDEX,VALUE) --ADD's ELEMENT AT PARTICULAR INDEX AND SHIFT REST ELEMENTS OF ARRAYLIST
19
20        System.out.println(arr_obj.isEmpty());//THIS METHOD CHECKS IS IT EMPTY AND RETURNS BOOLEAN VALUE
21        System.out.println(arr_obj);
22
23        arr_obj.set(0, "IIPS");
24        //Replaces the element at the specified position in this list with the specified element.
25        System.out.println(arr_obj);
26
27        System.out.println(arr_obj.size());
28        //RETURNS INT VALUE WHICH TELLS COUNT OF ELEMENT PRESENT IN ARRAYLIST
29
30        System.out.println(arr_obj.contains("AJAY"));
31        //RETURNS BOOLEAN VALUE THAT TELLS ELEMENT EXIST IN ARRAY LIST OR NOT ?
32        System.out.println(arr_obj.contains("VIJAY"));
33
34        System.out.println(arr_obj.get(3));//RETURNS ELEMENT AT PARTICULAR INDEX
35    }
36 }
37
38 }
39
```

The right panel shows the console output of the program. It starts with a header indicating the application path. The output matches the code's print statements: it confirms the list is empty and has a size of 0, lists the added elements, confirms the successful replacement of "HI" with "IIPS", shows the updated size, and confirms the presence of "VIJAY" in the list.

```
<terminated> Array_List [Java Application] C:\Program Files\Java\jdk-15\bin\
IMPLEMENTING ARRAYLIST PROGRAM (NON GENERIC APPROACH)
true
0
false
[HI, VIJAY, ROLL NO, 56, 5.6]
[IIPS, VIJAY, ROLL NO, 56, 5.6]
5
false
true
56
```

ArrayList2.java NAME - VIJAY BORATE ROLL NO - IT2K18-56

```
1 // TRAVERSING ARRAYLIST (NON GENERIC APPROACH) USING FOR LOOP
2 import java.util.ArrayList;
3 public class ArrayList2
4 {
5     public static void main(String[] args)
6     {
7         ArrayList arr_obj = new ArrayList(); // CREATING OBJECT OF ARRAY LIST
8         System.out.println(" TRAVERSING ARRAYLIST (NON GENERIC APPROACH) USING FOR LOOP");
9
10         arr_obj.add("HELLOW"); // ADD() METHOD ADDS ELEMENT IN THE ARRAY LIST
11         arr_obj.add("JAVA");
12         arr_obj.add("PROGRAMMER");
13         arr_obj.add(56);
14
15         arr_obj.add(0, "HI");
16         // ADD ( INDEX, VALUE ) -- ADD'S ELEMENT AT PARTICULAR INDEX AND SHIFT REST ELEMENTS OF ARRAYLIST
17
18         arr_obj.set(0, "IIPS");
19         // Replaces the element at the specified position in this list with the specified element.
20
21         // ***** USING FOR LOOP TO TRAVEL IN ARRAYLIST *****
22
23         for (int i = 0; i < arr_obj.size(); i++)
24         {
25             System.out.println(arr_obj.get(i));
26         }
27     }
28 }
29
30
31
32
33
34
35
```

Console

```
<terminated> ArrayList2 [Java Application] C:\Program Files\Java\jdk-15\bin\javaw.exe
 TRAVERSING ARRAYLIST (NON GENERIC APPROACH) USING FOR LOOP
IIPS
HELLOW
JAVA
PROGRAMMER
56
```

Array_List3.java NAME - VIJAY BORATE ROLL NO - IT2K18-56

```
1 // TRAVERSING ARRAYLIST (NON GENERIC APPROACH) USING 2) FOR EACH LOOP
2 import java.util.ArrayList;
3 public class Array_list3
4 {
5     public static void main(String[] args)
6     {
7         ArrayList arr_obj = new ArrayList(); // CREATING OBJECT OF ARRAY LIST
8         System.out.println(" TRAVERSING ARRAYLIST (NON GENERIC APPROACH) USING FOR EACH LOOP");
9         arr_obj.add("HELLOW"); // ADD() METHOD ADDS ELEMENT IN THE ARRAY LIST
10        arr_obj.add("JAVA");
11        arr_obj.add("PROGRAMMER");
12        arr_obj.add(56);
13
14        arr_obj.add(0, "HI");
15        // ADD ( INDEX, VALUE) - ADDS ELEMENT AT PARTICULAR INDEX AND SHIFT REST ELEMENTS OF ARRAYLIST
16
17        arr_obj.set(0, "IIPS");
18        // Replaces the element at the specified position in this list with the specified element.
19
20        // ***** USING FOR EACH LOOP TO TRAVEL IN ARRAYLIST *****
21
22        for (Object i : arr_obj)
23        {
24            System.out.println(i);
25        }
26
27
28
29
30    }
31
32 }
33
```

Console

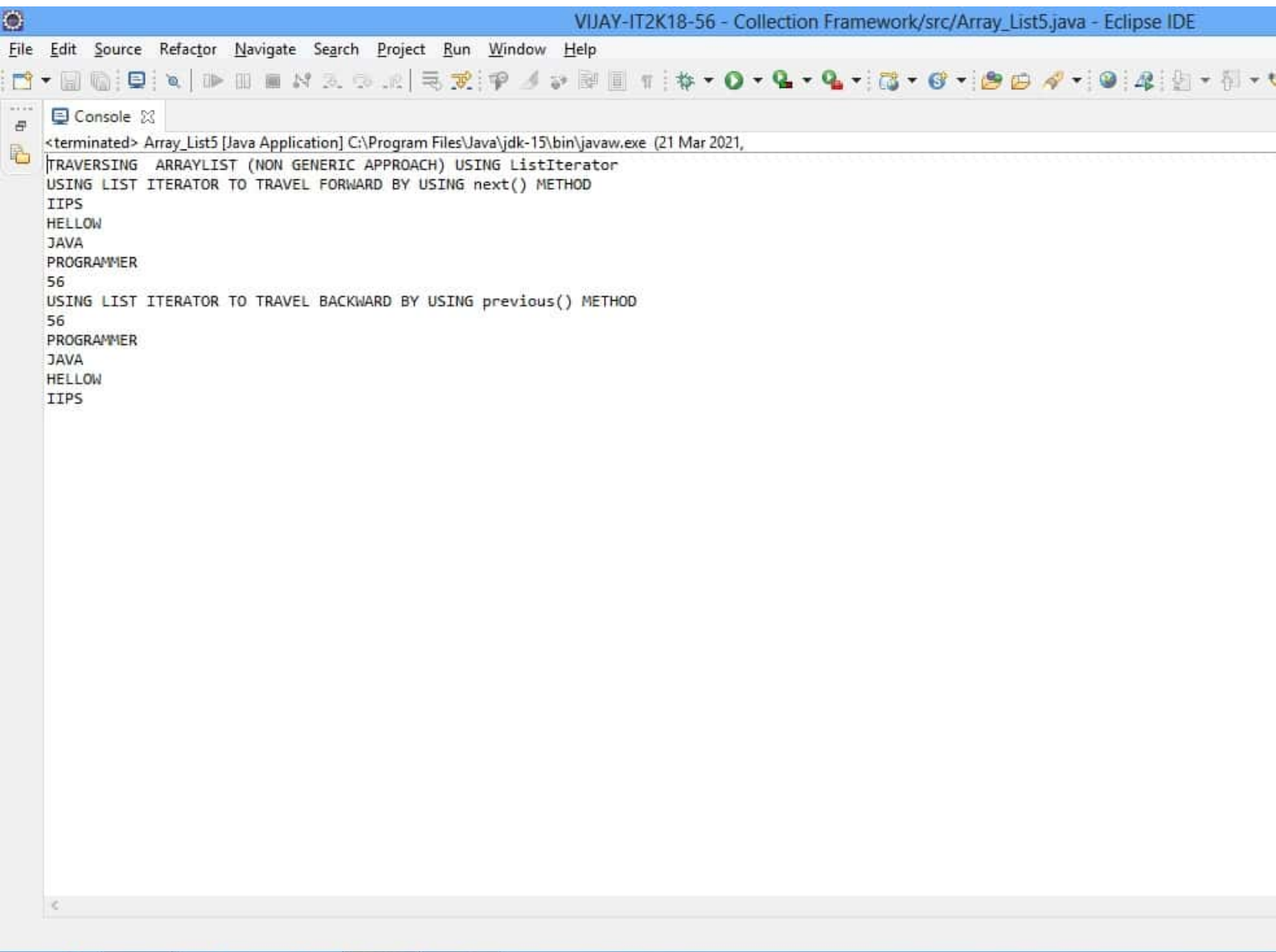
```
<terminated> Array_List3 [Java Application] C:\Program Files\Java\jdk-15\bin\javaw.exe (2)
TRAVERSING ARRAYLIST (NON GENERIC APPROACH) USING FOR EACH LOOP
IIPS
HELLOW
JAVA
PROGRAMMER
56
```

```
Array_List4.java NAME - VIJAY BORATE ROLL NO - IT2K18-56
1 // TRAVERSING ARRAYLIST (NON GENERIC APPROACH) USING 3) Iterator
2 import java.util.ArrayList;
3
4
5
6 public class Array_List4
7 {
8     public static void main(String[] args)
9     {
10         ArrayList arr_obj = new ArrayList(); // CREATING OBJECT OF ARRAY LIST
11         System.out.println(" TRAVERSING ARRAYLIST (NON GENERIC APPROACH) USING 3) Iterator");
12         arr_obj.add("HELLOW"); // ADD() METHOD ADDS ELEMENT IN THE ARRAY LIST
13         arr_obj.add("JAVA");
14         arr_obj.add("PROGRAMMER");
15         arr_obj.add(56);
16
17         arr_obj.add(0, "HI");
18         // ADD ( INDEX, VALUE ) -- ADDS ELEMENT AT PARTICULAR INDEX AND SHIFT REST ELEMENTS OF ARRAYLIST
19
20         arr_obj.set(0, "IIPS");
21         // Replaces the element at the specified position in this list with the specified element.
22
23         // ***** USING ITERATOR TO TRAVEL IN ARRAYLIST *****
24
25         Iterator obj = arr_obj.iterator();
26         while (obj.hasNext()) {
27             Object o1 = (Object) obj.next();
28             System.out.println(o1);
29         }
30     }
31 }
32
33
34 }
35
36 }
37
```

```
Console
<terminated> Array_List4 [Java Application] C:\Program Files\Java\jdk-15\bin\javaw.exe
TRAVERSING ARRAYLIST (NON GENERIC APPROACH) USING 3) Iterator
IIPS
HELLOW
JAVA
PROGRAMMER
56
```

```
VIJAY-IT2K18-56 - Collection Framework/src/Array_List5.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help

*Array_List5.java
1 // TRAVERSING ARRAYLIST (NON GENERIC APPROACH) USING 3) ListIterator
2 import java.util.ArrayList;
3 public class Array_List5
4 {
5     public static void main(String[] args)
6     {
7         ArrayList arr_obj = new ArrayList(); // CREATING OBJECT OF ARRAY LIST
8         System.out.println(" TRAVERSING ARRAYLIST (NON GENERIC APPROACH) USING ListIterator");
9
10        arr_obj.add("HELLO"); // ADD() METHOD ADDS ELEMENT IN THE ARRAY LIST
11        arr_obj.add("JAVA");
12        arr_obj.add("PROGRAMMER");
13        arr_obj.add(56);
14
15        arr_obj.add(0, "HI"); // ADD ( INDEX, VALUE) -- ADDS ELEMENT AT PARTICULAR INDEX AND SHIFT REST ELEMENTS OF ARRAYLIST
16
17        arr_obj.set(0, "IIPS"); // Replaces the element at the specified position in this list with the specified element.
18        // ***** USING LIST ITERATOR TO TRAVEL FORWARD[next()] & BACKWARD[previous()] IN ARRAYLIST *****
19
20        ListIterator obj = arr_obj.listIterator();
21
22        System.out.println(" USING LIST ITERATOR TO TRAVEL FORWARD BY USING next() METHOD");
23        while (obj.hasNext())
24        {
25            Object O1 = (Object) obj.next(); // ***** next Allows you to Travel Forward Direction
26
27            System.out.println(O1);
28        }
29
30        System.out.println(" USING LIST ITERATOR TO TRAVEL BACKWARD BY USING previous() METHOD");
31        while (obj.hasPrevious())
32        {
33            Object O1 = (Object) obj.previous(); // ***** next Allows you to Travel Backward Direction
34
35            System.out.println(O1);
36        }
37    }
38 }
39
40
41
```



VIJAY-IT2K18-56 - Collection Framework/src/Array_List6.java - Eclipse IDE

```
1 //PASSING ARRAY LIST INSIDE ANOTHER ARRAY LIST (NON GENERIC APPROACH)
2 import java.util.ArrayList;
3 public class Array_List6
4 {
5     public static void main(String[] args)
6     {
7         System.out.println("PASSING ARRAY LIST INSIDE ANOTHER ARRAY LIST (NON GENERIC APPROACH)");
8         ArrayList arr_obj1 = new ArrayList();//CREATING OBJECT OF ARRAY LIST 1
9         arr_obj1.add("HI");// ADD() METHOD ADDS ELEMENT IN THE ARRAY LIST 1
10        arr_obj1.add("WELCOME");
11        arr_obj1.add("GOODBYE");
12        System.out.println("ELEMENTS PRESENT IN ARRAY LIST 1 ARE ");
13        System.out.println(arr_obj1);
14
15        ArrayList arr_obj2 = new ArrayList(arr_obj1);//Creating new ARRAYLIST2 & PASSING ARRAY LIST1 IN IT
16        arr_obj2.add(10);
17        arr_obj2.add(20);
18        arr_obj2.add(30);
19        System.out.println("ELEMENTS PRESENT IN ARRAY LIST 2 ARE ");
20        System.out.println(arr_obj2);
21    }
22 }
23
```

Console

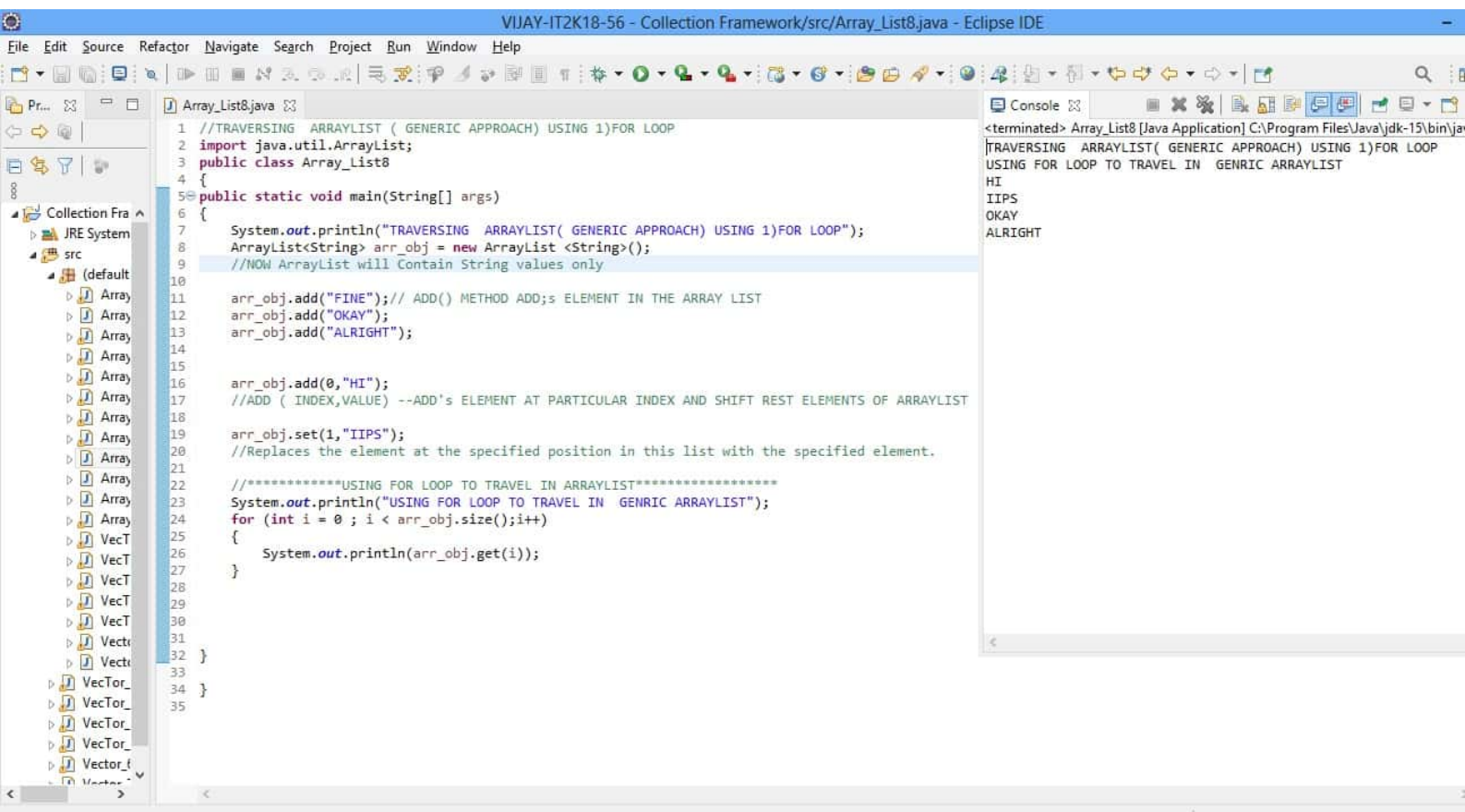
```
<terminated> Array_List6 [Java Application] C:\Program Files\Java\jdk-15\bin\javaw.exe (2
PASSING ARRAY LIST INSIDE ANOTHER ARRAY LIST (NON GENERIC APPROACH)
ELEMENTS PRESENT IN ARRAY LIST 1 ARE
[HI, WELCOME, GOODBYE]
ELEMENTS PRESENT IN ARRAY LIST 2 ARE
[HI, WELCOME, GOODBYE, 10, 20, 30]
```

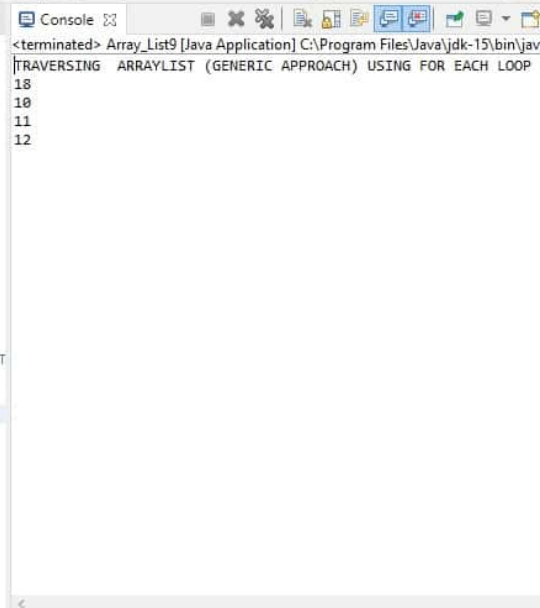

VIJAY-IT2K18-56 - Collection Framework/src/Array_List7.java - Eclipse IDE

```
1 //IMPLEMENTING ARRAYLIST PROGRAM (GENERIC APPROACH)
2 import java.util.ArrayList;
3 public class Array_List7
4 {
5     public static void main(String[] args)
6     {
7         System.out.println("IMPLEMENTING ARRAYLIST PROGRAM (GENERIC APPROACH)");
8         ArrayList<String> arr_obj = new ArrayList<String>();
9         //NOW ArrayList will Contain String values only
10        System.out.println(arr_obj.isEmpty());
11        //THIS METHOD CHECKS IS IT EMPTY AND RETURNS BOOLEAN VALUE
12        System.out.println(arr_obj.size());
13        //RETURNS INTEGER COUNT OF NUMBER OF ELEMENTS IN ARRAYLIST
14
15        arr_obj.add("SUN");// ADD() METHOD ADD;s ELEMENT IN THE ARRAY LIST
16        arr_obj.add("MOON");
17        arr_obj.add("EARTH");
18        arr_obj.add("JUPITOR");
19
20        arr_obj.add(0,"VENUS");
21        //ADD ( INDEX,VALUE) --ADD's ELEMENT AT PARTICULAR INDEX AND SHIFT REST ELEMENTS OF ARRAYLIST
22
23        System.out.println(arr_obj.isEmpty());
24        //THIS METHOD CHECKS IS IT EMPTY AND RETURNS BOOLEAN VALUE
25        System.out.println(arr_obj);
26
27        arr_obj.set(0, "MARS");
28        //Replaces the element at the specified position in this list with the specified element.
29
30        System.out.println(arr_obj);
31
32        System.out.println(arr_obj.size());//RETURNS INT VALUE WHICH TELLS COUNT OF ELEMENT PRESENT IN ARRAYLIST
33        System.out.println(arr_obj.contains("NEPTUNE"));
34        //RETURNS BOOLEAN VALUE THAT TELLS ELEMENT EXIST IN ARRAY LIST OR NOT ?
35        System.out.println(arr_obj.contains("EARTH"));
36
37        System.out.println(arr_obj.get(3));//RETURNS ELEMENT AT PARTICULAR INDEX
38    }}
39
```

Console

```
<terminated> Array_List7 [Java Application] C:\Program Files\Java\jdk-15\bin\javaw.exe
IMPLEMENTING ARRAYLIST PROGRAM (GENERIC APPROACH)
true
0
false
[VENUS, SUN, MOON, EARTH, JUPITOR]
[MARS, SUN, MOON, EARTH, JUPITOR]
5
false
true
EARTH
```



VIJAY-IT2K18-56 - Collection Framework/src/Array_List10.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

Array_List10.java

```
1 // TRAVERSING ARRAYLIST (GENERIC APPROACH) USING Iterator
2 import java.util.ArrayList;
3
4
5
6 public class Array_List10
7 {
8     public static void main(String[] args)
9     {
10         System.out.println("TRAVERSING ARRAYLIST (GENERIC APPROACH) USING Iterator");
11         ArrayList<Double> arr_obj = new ArrayList<Double>();
12         // NOW ArrayList will Contain Double values only
13         arr_obj.add(3.14); // ADD() METHOD ADDS ELEMENT IN THE ARRAY LIST
14         arr_obj.add(9.11);
15         arr_obj.add(16.5);
16         arr_obj.add(56.0);
17
18         arr_obj.add(0, 9.9);
19         // ADD ( INDEX, VALUE) -- ADDS ELEMENT AT PARTICULAR INDEX AND SHIFT REST ELEMENTS OF ARRAYLIST
20
21         arr_obj.set(0, 36.9);
22         // Replaces the element at the specified position in this list with the specified element.
23
24         // ***** USING ITERATOR TO TRAVEL IN ARRAYLIST *****
25
26         Iterator<Double> obj = arr_obj.iterator();
27         while (obj.hasNext()) {
28             Double O1 = (Double) obj.next();
29             System.out.println(O1);
30         }
31     }
32 }
33
34
35
36
37
38
```

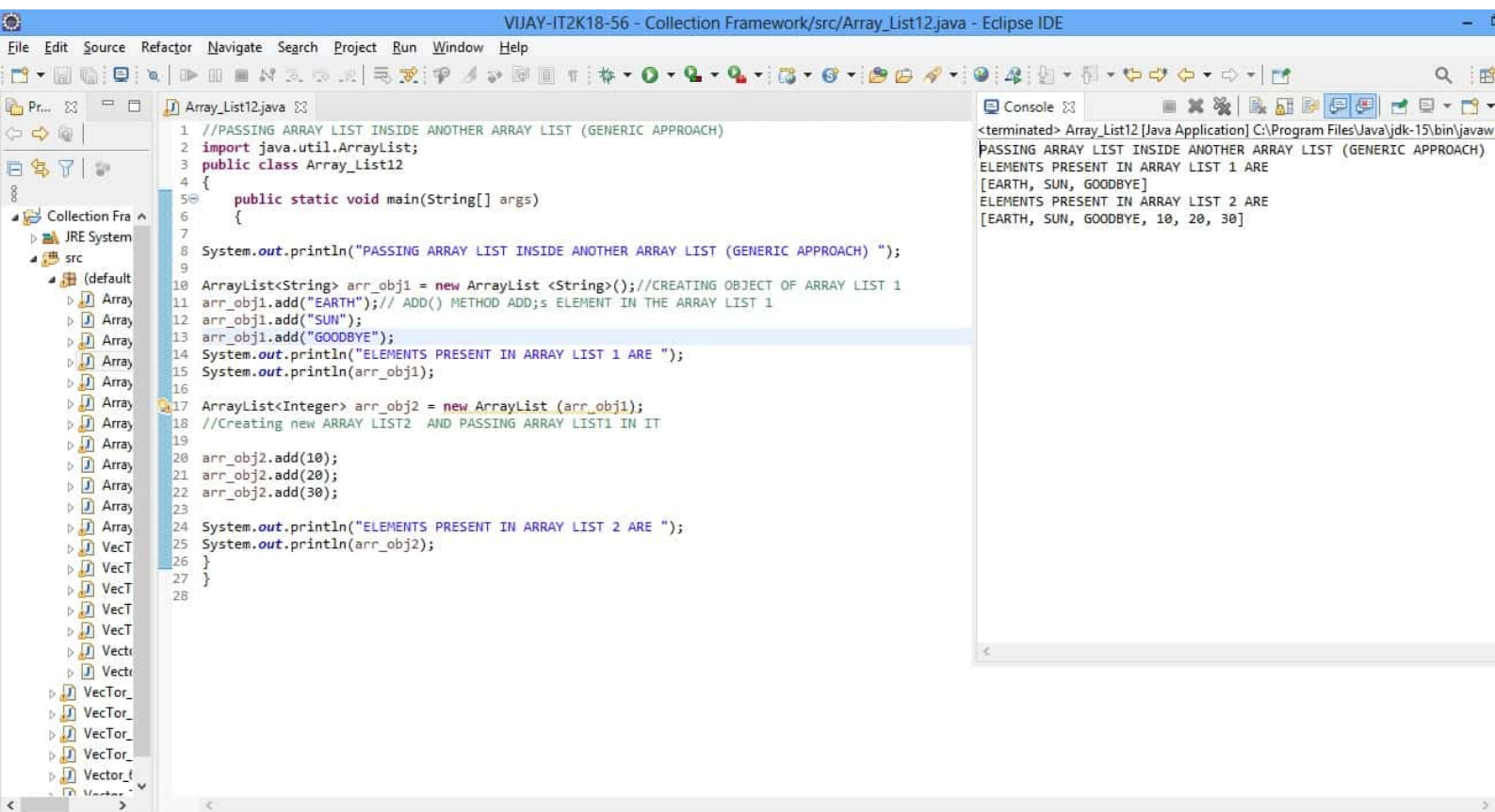
Console

```
<terminated> Array_List10 [Java Application] C:\Program Files\Java\jdk-15\bin
TRAVERSING ARRAYLIST (GENERIC APPROACH) USING Iterator
36.9
3.14
9.11
16.5
56.0
```

```
VIJAY-IT2K18-56 - Collection Framework/src/Array_List11.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help

Array_List11.java
1 //TRAVERSING ARRAYLIST ( GENERIC APPROACH) USING 3)ListIterator
2 import java.util.ArrayList;
3 public class Array_List11
4 {
5     public static void main(String[] args)
6     {
7         System.out.println("IMPLEMENTING ARRAYLIST PROGRAM (GENERIC APPROACH)");
8         ArrayList<Character> arr_obj = new ArrayList<Character>();
9         //NOW ArrayList will Contain Character values only
10        arr_obj.add('A');// ADD() METHOD ADD's ELEMENT IN THE ARRAY LIST
11        arr_obj.add('I');
12        arr_obj.add('J');
13        arr_obj.add('A');
14        arr_obj.add('V');
15        arr_obj.add('I');
16        //ADD ( INDEX,VALUE) --ADD's ELEMENT AT PARTICULAR INDEX AND SHIFT REST ELEMENTS OF ARRAYLIST
17        arr_obj.add(1,'M');
18        arr_obj.set(2,'V');
19        //Set Replaces the element at the specified position in this list with the specified element.
20        //*****USING LIST ITERATOR TO TRAVEL FORWARD[next()] & BACKWARD[previous()] IN ARRAYLIST*****
21
22        ListIterator obj = arr_obj.listIterator();
23        System.out.println("USING LIST ITERATOR TO TRAVEL FORWARD BY USING next() METHOD");
24        while (obj.hasNext())
25        {
26            Character O1 = (Character) obj.next(); //****next Allows you to Travel Forward Direction
27            System.out.println(O1);
28        }
29
30        System.out.println("USING LIST ITERATOR TO TRAVEL BACKWARD BY USING previous() METHOD");
31        while (obj.hasPrevious())
32        {
33            Character O1 = (Character) obj.previous(); //****next Allows you to Travel Backward Direction
34            System.out.println(O1);
35        }
36    }
37 }
38
39
40
41
```

```
<terminated> Array_List11 [Java Application] C:\Program Files\Java\jdk-15\bin\javaw.exe
IMPLEMENTING ARRAYLIST PROGRAM (GENERIC APPROACH)
USING LIST ITERATOR TO TRAVEL FORWARD BY USING next() METHOD
A
I
J
A
V
I
M
V
I
J
A
Y
USING LIST ITERATOR TO TRAVEL BACKWARD BY USING previous() METHOD
Y
A
J
I
V
M
I
```



VIJAY-IT2K18-56 - Collection Framework/src/VecTor_1.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

VecTor_1.java

```
1 //IMPLEMENTING VECTOR PROGRAM (NON GENERIC APPROACH)
2 import java.util.Vector;
3 public class VecTor_1
4 {
5     public static void main(String[] args)
6     {
7         System.out.println("IMPLEMENTING VECTOR PROGRAM (NON GENERIC APPROACH)");
8         Vector arr_obj = new Vector();//CREATING OBJECT OF VECTOR and CAPACITY 10
9         Vector arr_obj2 = new Vector(5);//CREATING VECTOR OF CAPACITY 5
10
11
12         System.out.println(arr_obj.capacity());//RETURNS INITIAL CAPACITY OF VECTOR
13         System.out.println(arr_obj2.capacity());
14         System.out.println(arr_obj.isEmpty());//THIS METHOD CHECKS IS IT EMPTY AND RETURNS BOOLEAN VALUE
15         System.out.println(arr_obj.size());//RETURNS INTEGER COUNT OF NUMBER OF ELEMENTS IN VECTOR
16
17         arr_obj.add("Vijay");// ADD() METHOD ADD'S ELEMENT IN THE VECTOR
18
19         arr_obj.addElement("BORATE");//THIS METHOD ALSO ADD'S ELEMENT IN THE VECTOR
20         arr_obj.add("ROLL_NO");
21         arr_obj.add(56);
22         arr_obj.add(5.6);
23
24         arr_obj.add(0, "HI");//ADD ( INDEX,VALUE) --ADD'S ELEMENT AT PARTICULAR INDEX AND SHIFT REST ELEMENTS OF VECTOR
25
26         System.out.println(arr_obj.isEmpty());//THIS METHOD CHECKS IS IT EMPTY AND RETURNS BOOLEAN VALUE
27         System.out.println(arr_obj);
28
29         arr_obj.set(0, "IIPS");//Replaces the element at the specified position in Vector with the specified element.
30         System.out.println(arr_obj);
31
32         System.out.println(arr_obj.size());//RETURNS INT VALUE WHICH TELLS COUNT OF ELEMENT PRESENT IN VECTOR
33
34         System.out.println(arr_obj.contains("AJAY"));//RETURNS BOOLEAN VALUE THAT TELLS ELEMENT EXIST IN VECTOR OR NOT ?
35         System.out.println(arr_obj.contains("VIJAY"));
36
37         System.out.println(arr_obj.get(3));//RETURNS ELEMENT AT PARTICULAR INDEX
38     }
39 }
```

Console

```
<terminated> VecTor_1 [Java Application] C:\Program Files\Java\jdk-15\bin
IMPLEMENTING VECTOR PROGRAM (NON GENERIC APPROACH)
10
5
true
0
false
[HI, Vijay, BORATE, ROLL NO, 56, 5.6]
[IIPS, Vijay, BORATE, ROLL NO, 56, 5.6]
6
false
false
false
ROLL NO
```

VIJAY-IT2K18-56 - Collection Framework/src/VecTor_2.java - Eclipse IDE

14]

```
1 // TRAVERSING VECTOR (NON GENERIC APPROACH) USING FOR LOOP
2
3
4 import java.util.Vector;
5 public class VecTor_2
6 {
7     public static void main(String[] args)
8     {
9
10        Vector arr_obj = new Vector(5,1);
11        //WE CAN SPECIFY INCREMENTAL FACTOR WHILE CREATING VECTOR
12
13        System.out.println("TRAVERSING VECTOR (NON GENERIC APPROACH) USING FOR LOOP");
14
15        arr_obj.add("HELLO");// ADD() METHOD ADDS ELEMENT IN THE VECTOR
16
17        arr_obj.add("JAVA");
18
19        arr_obj.add("PROGRAMMER");
20
21        arr_obj.addElement(56);//ADDs ELEMENT IN THE VECTOR
22
23
24        arr_obj.add(0,"HI");//ADD ( INDEX,VALUE) --ADDs ELEMENT AT PARTICULAR INDEX AND SHIFT REST ELEMENTS OF VECTOR
25
26
27        arr_obj.set(0,"IIPS");//Replaces the element at the specified position in VECTOR with the specified element.
28
29        //*****USING FOR LOOP TO TRAVEL IN VECTOR*****
30
31        for (int i = 0 ; i < arr_obj.size();i++)
32        {
33            System.out.println(arr_obj.get(i));
34        }
35
36    }
37
38 }
39
```

Console

```
<terminated> VecTor_2 [Java Application] C:\Program Files\Java\jdk-15\bin\javaw.exe
TRAVERSING VECTOR (NON GENERIC APPROACH) USING FOR LOOP
IIPS
HELLO
JAVA
PROGRAMMER
56
```


VIJAY-IT2K18-56 - Collection Framework/src/VecTor_3.java - Eclipse IDE

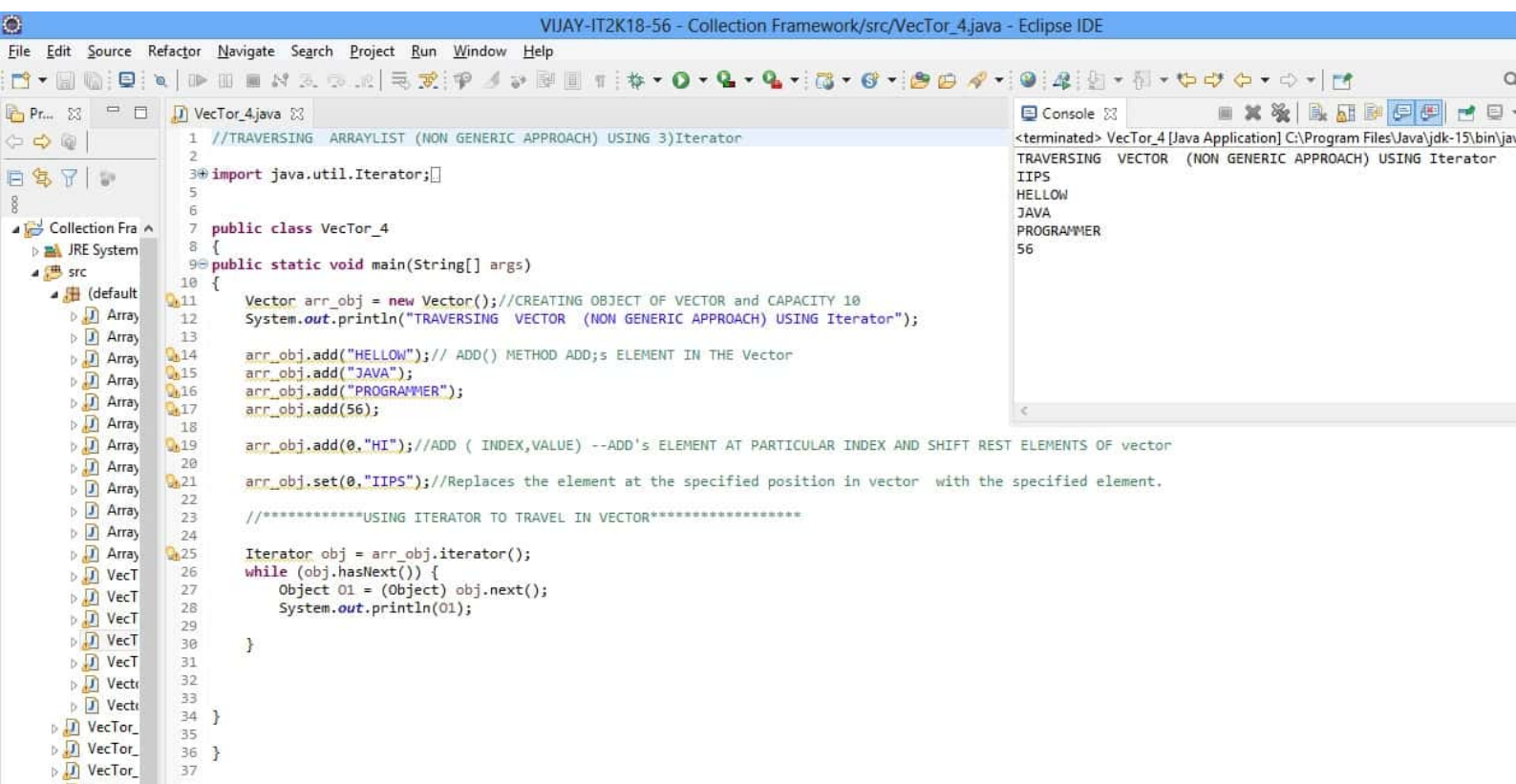
File Edit Source Refactor Navigate Search Project Run Window Help

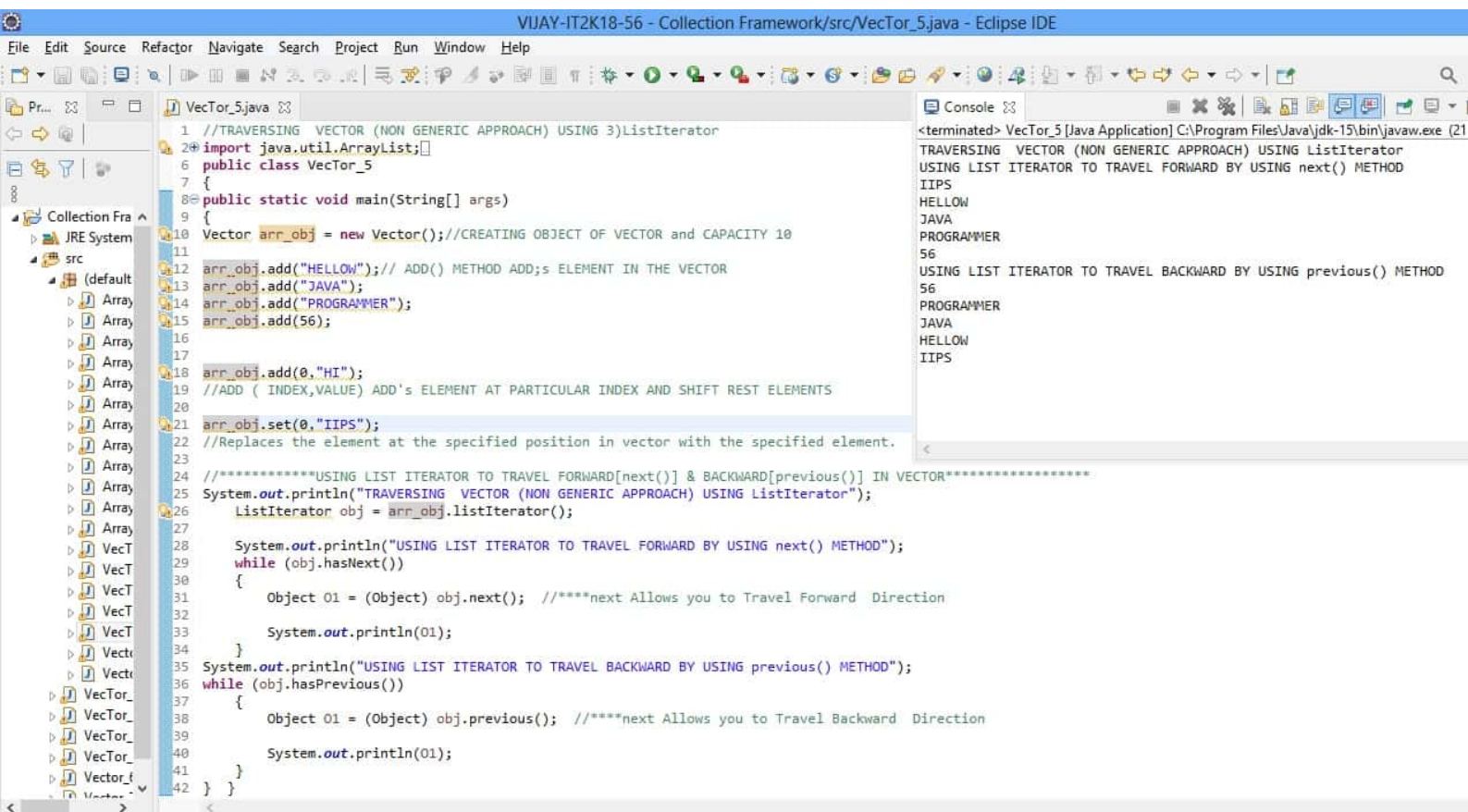
VecTor_3.java

```
1 // TRAVERSING VECTOR (NON GENERIC APPROACH) USING 2) FOR EACH LOOP
2 import java.util.ArrayList;
3
4 public class VecTor_3
5 {
6     public static void main(String[] args)
7     {
8         Vector arr_obj = new Vector();//CREATING OBJECT OF VECTOR and CAPACITY 10
9         System.out.println(" TRAVERSING VECTOR (NON GENERIC APPROACH) USING FOR EACH LOOP");
10
11         arr_obj.add("HELLOW");// ADD() METHOD ADDS ELEMENT IN THE VECTOR
12         arr_obj.add("JAVA");
13         arr_obj.add("PROGRAMMER");
14         arr_obj.add(56);
15
16         arr_obj.add(0,"HI");
17         //ADD ( INDEX,VALUE) --ADD'S ELEMENT AT PARTICULAR INDEX AND SHIFT REST ELEMENTS OF VECTOR
18
19         arr_obj.set(0,"IIPS");
20         //Replaces the element at the specified position IN VECTOR with the specified element.
21
22         //*****USING FOR EACH LOOP TO TRAVEL IN VECTOR*****
23
24         for (Object i : arr_obj)
25         {
26             System.out.println(i);
27         }
28
29
30
31
32 }
33
34
35
```

Console

```
<terminated> VecTor_3 [Java Application] C:\Program Files\Java\jdk-15\bin\javaw.exe
TRAVERSING VECTOR (NON GENERIC APPROACH) USING FOR EACH LOOP
IIPS
HELLOW
JAVA
PROGRAMMER
56
```





VJAY-IT2K18-56 - Collection Framework/src/Vector_7.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

Vector_7.java

```
1 //IMPLEMENTING VECTOR (GENERIC APPROACH)
2 import java.util.Vector;
3 public class Vector_7
4 {
5     public static void main(String[] args)
6     {
7
8         System.out.println("IMPLEMENTING VECTOR (GENERIC APPROACH)");
9         Vector<String> arr_obj = new Vector<String>();//NOW Vector will Contain String values only
10
11         System.out.println(arr_obj.isEmpty());//THIS METHOD CHECKS IS IT EMPTY AND RETURNS BOOLEAN VALUE
12         System.out.println(arr_obj.size());//RETURNS INTEGER COUNT OF NUMBER OF ELEMENTS IN VECTOR
13
14         arr_obj.add("SUN");// ADD() METHOD ADD;s ELEMENT IN THE VECTOR
15         arr_obj.add("MOON");
16         arr_obj.add("EARTH");
17         arr_obj.add("JUPITOR");
18
19         arr_obj.add(0,"VENUS");//ADD ( INDEX,VALUE) --ADD's ELEMENT AT PARTICULAR INDEX AND SHIFT REST ELEMENTS OF VECTOR
20
21         System.out.println(arr_obj.isEmpty());//THIS METHOD CHECKS IS IT EMPTY AND RETURNS BOOLEAN VALUE
22         System.out.println(arr_obj);
23
24         arr_obj.set(0, "MARS");//Replaces the element at the specified position in vector with the specified element.
25         System.out.println(arr_obj);
26
27         System.out.println(arr_obj.size());//RETURNS INT VALUE WHICH TELLS COUNT OF ELEMENT PRESENT IN VECTOR
28
29         System.out.println(arr_obj.contains("NEPTUNE"));//RETURNS BOOLEAN VALUE THAT TELLS ELEMENT EXIST IN VECTOR OR NOT ?
30         System.out.println(arr_obj.contains("EARTH"));
31
32         System.out.println(arr_obj.get(3));//RETURNS ELEMENT AT PARTICULAR INDEX
33
34     }
35 }
36
```

Console

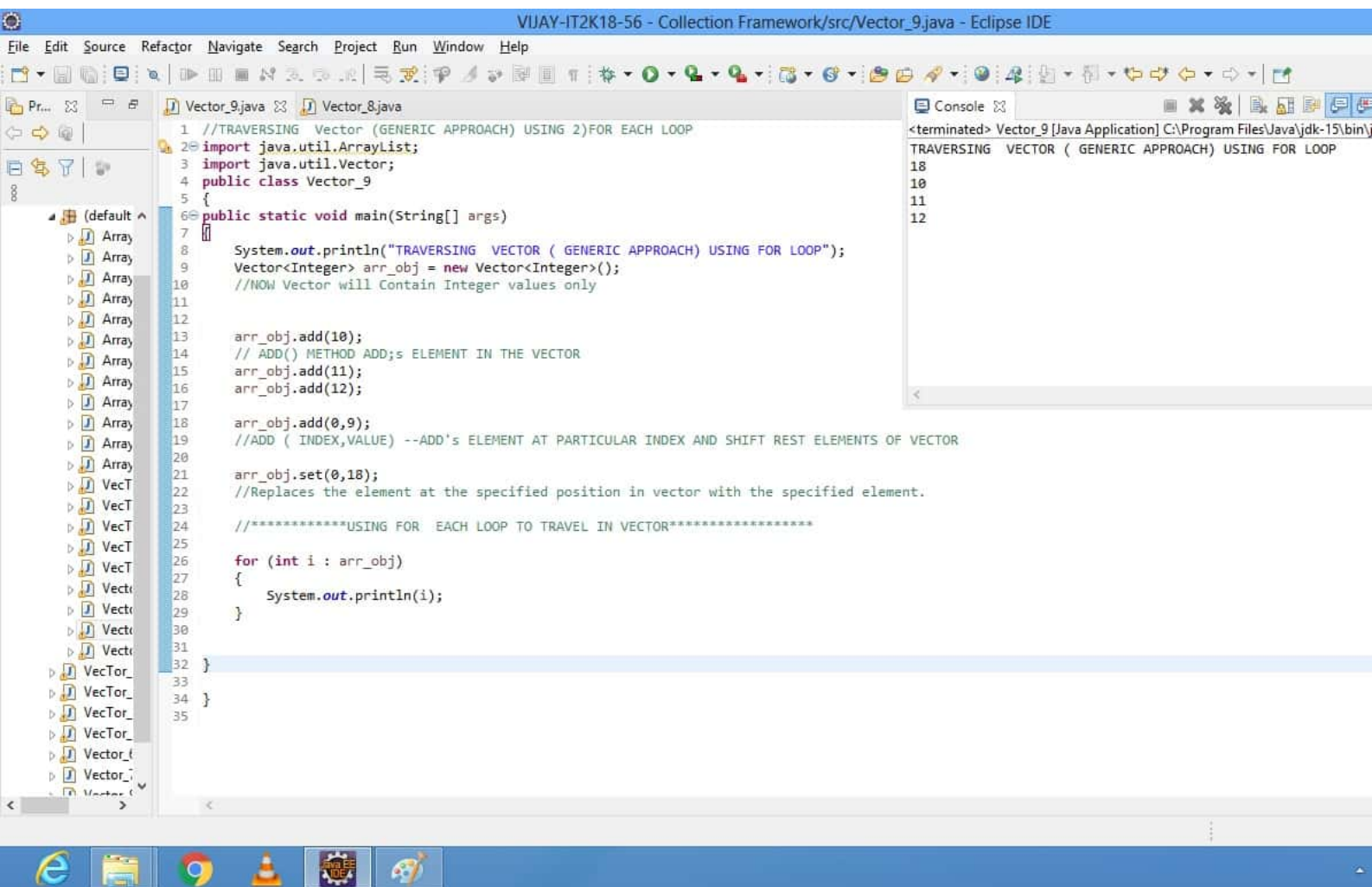
<terminated> Vector_7 [Java Application] C:\Program Files\Java\jre6\bin\java.exe
IMPLEMENTING VECTOR (GENERIC APPROACH)
true
0
false
[VENUS, SUN, MOON, EARTH, JUPITOR]
[MARS, SUN, MOON, EARTH, JUPITOR]
5
false
true
EARTH

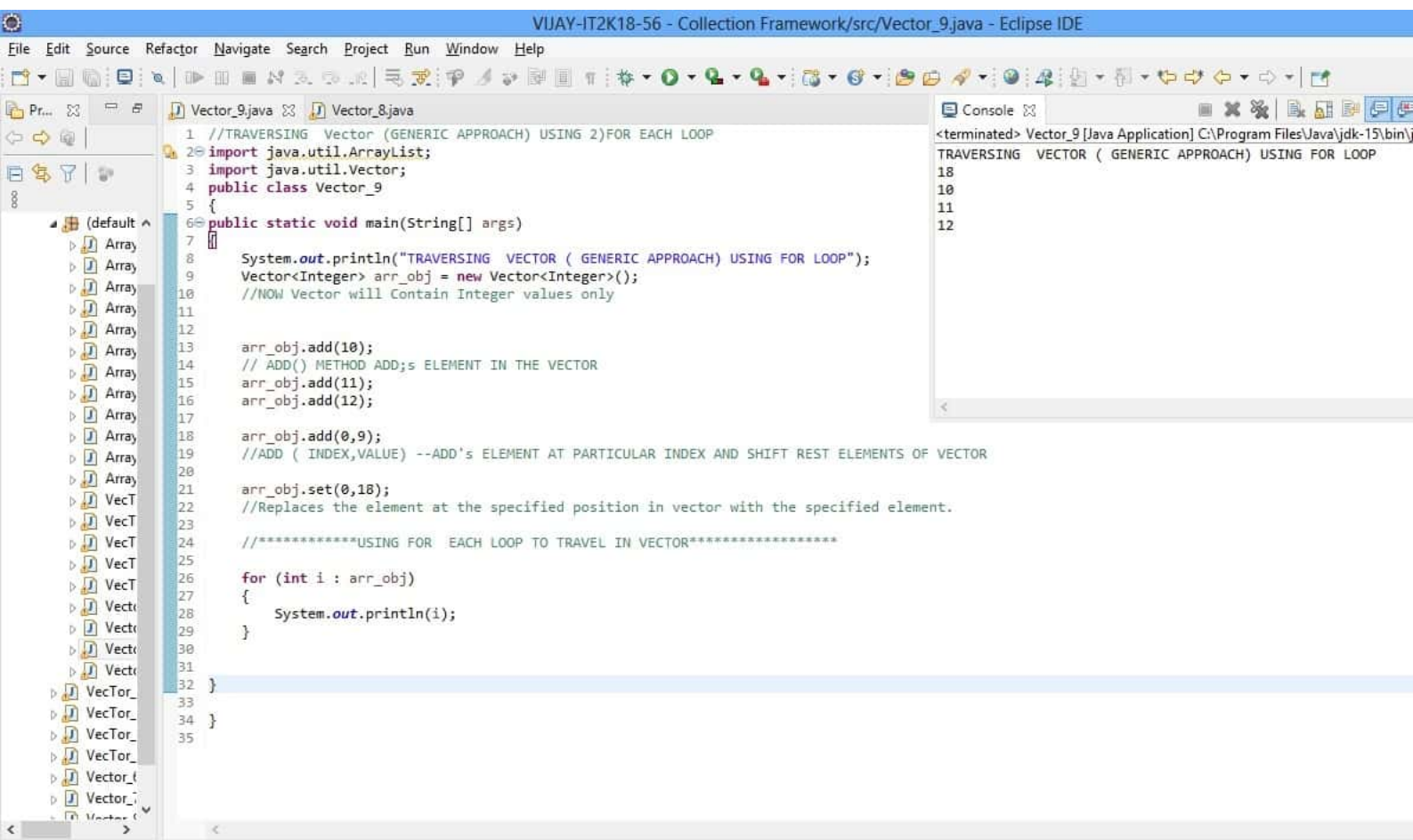
Vector_8.java

```
1 // TRAVERSING VECTOR ( GENERIC APPROACH) USING FOR LOOP
2 import java.util.ArrayList;
3 import java.util.Vector;
4 public class Vector_8
5 {
6     public static void main(String[] args)
7     {
8         System.out.println(" TRAVERSING VECTOR ( GENERIC APPROACH) USING FOR LOOP");
9         Vector<String> arr_obj = new Vector<String>(); // NOW Vector will Contain String values only
10
11         // NOW VECTOR will Contain String values only
12
13         arr_obj.add("FINE"); // ADD() METHOD ADDS ELEMENT IN THE VECTOR
14         arr_obj.add("OKAY");
15         arr_obj.add("ALRIGHT");
16
17
18         arr_obj.add(0, "HI");
19         // ADD ( INDEX, VALUE) -- ADDS ELEMENT AT PARTICULAR INDEX AND SHIFT REST ELEMENTS OF VECTOR
20
21         arr_obj.set(1, "IIPS");
22         // Replaces the element at the specified position in vector with the specified element.
23
24         // ***** USING FOR LOOP TO TRAVEL IN VECTOR *****
25         System.out.println(" USING FOR LOOP TO TRAVEL IN GENERIC VECTOR ");
26         for (int i = 0; i < arr_obj.size(); i++)
27         {
28             System.out.println(arr_obj.get(i));
29         }
30
31
32
33
34     }
35 }
36
37
```

Console

```
<terminated> Vector_8 [Java Application] C:\Program Files\Java\jdk-15\bin\java.exe
TRAVERSING VECTOR ( GENERIC APPROACH) USING FOR LOOP
USING FOR LOOP TO TRAVEL IN GENERIC VECTOR
HI
IIPS
OKAY
ALRIGHT
```





VIJAY-IT2K18-56 - Collection Framework/src/Vector_11.java - Eclipse IDE

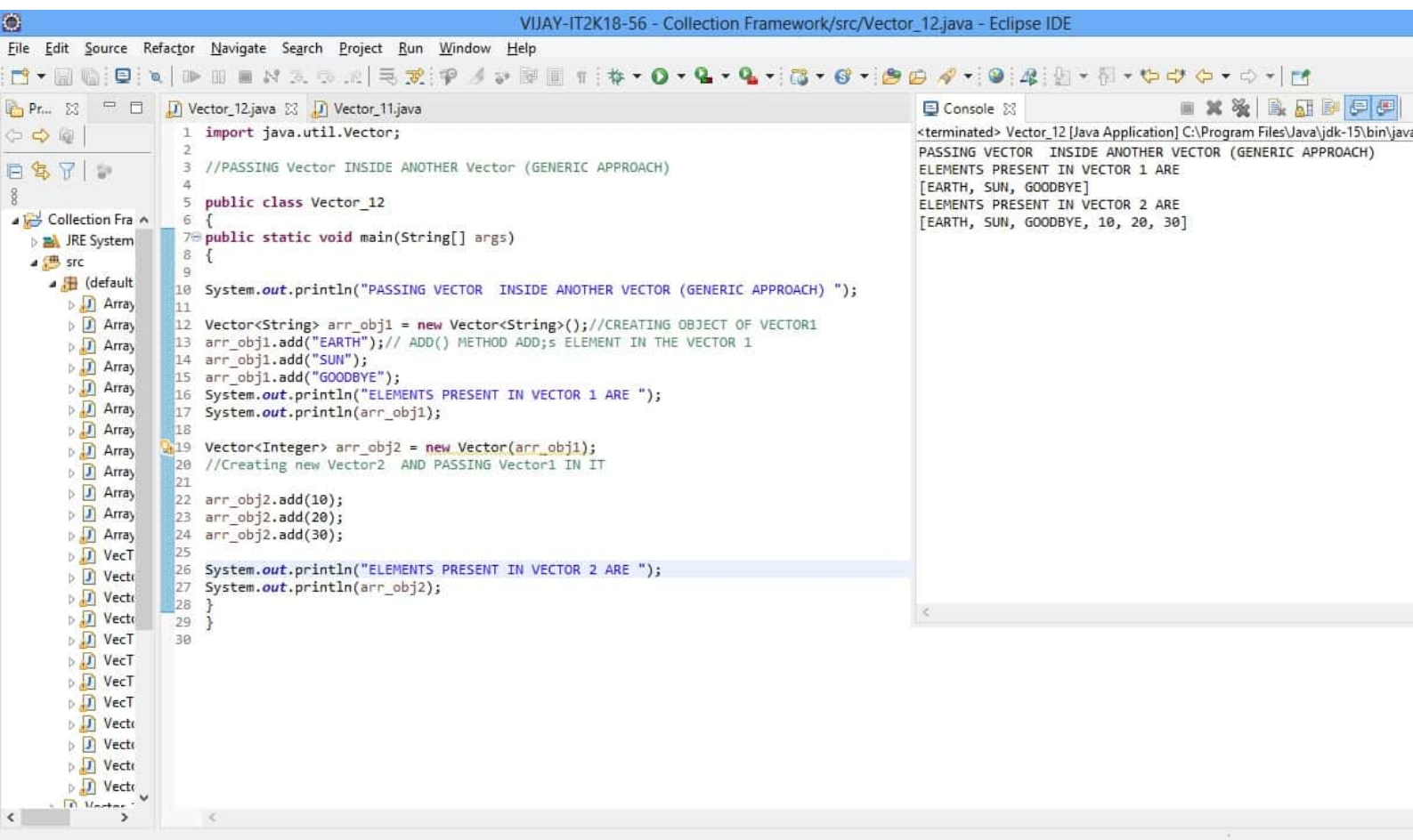
File Edit Source Refactor Navigate Search Project Run Window Help

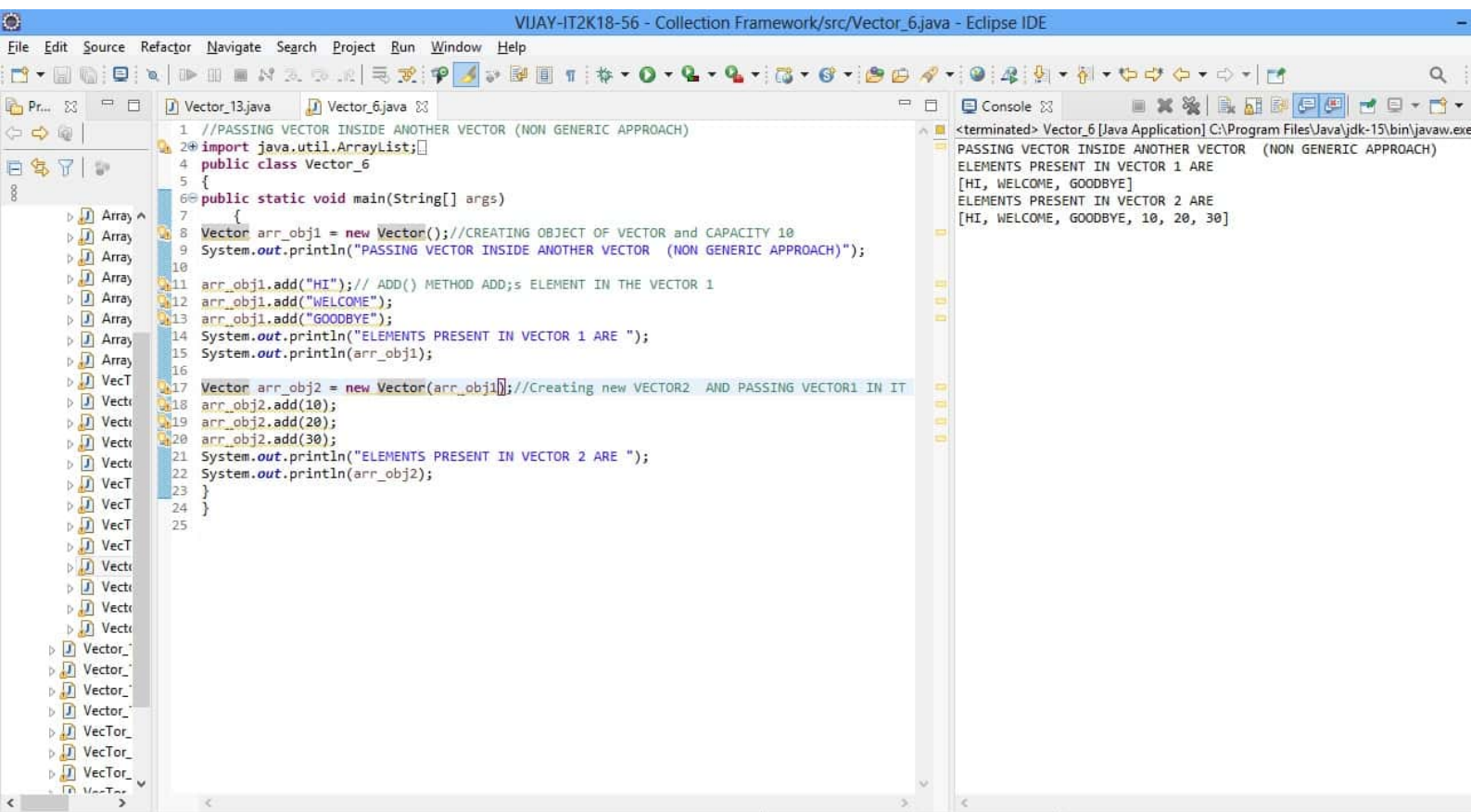
Vector_11.java Vector_10.java

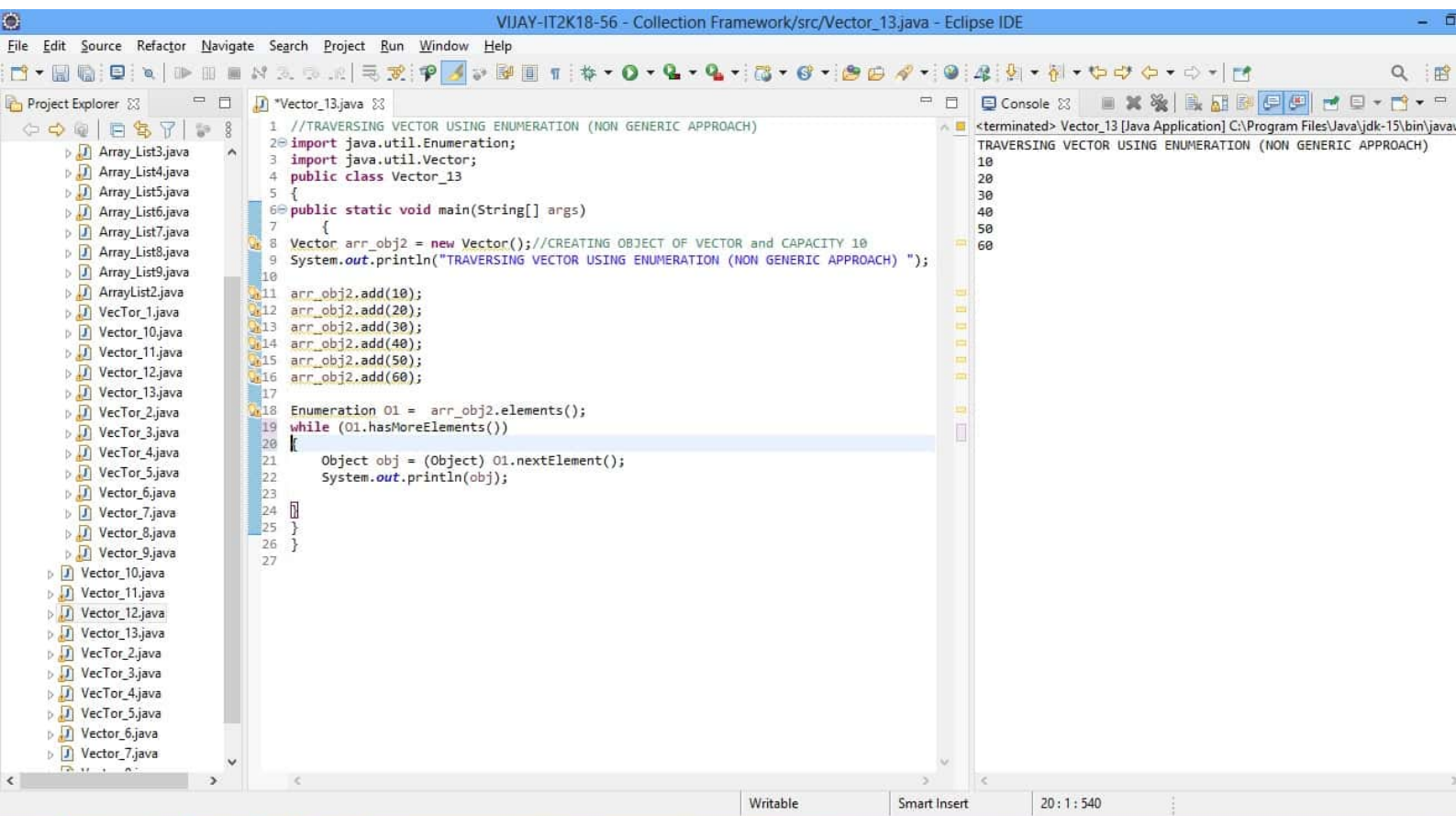
```
1 //TRAVERSING VECTOR ( GENERIC APPROACH) USING 3)ListIterator
2 import java.util.ListIterator;
3 import java.util.Vector;
4 public class Vector_11
5 {
6 public static void main(String[] args)
7 {
8     System.out.println("TRAVERSING VECTOR ( GENERIC APPROACH) USING 3)ListIterator");
9     Vector<Character> arr_obj = new Vector<Character>();
10
11     //NOW VECTOR will Contain Character values only
12     arr_obj.add('A');// ADD() METHOD ADDS ELEMENT IN THE VECTOR
13     arr_obj.add('I');
14     arr_obj.add('J');
15     arr_obj.add('A');
16     arr_obj.add('Y');
17
18     arr_obj.add(0,'I');
19     //ADD ( INDEX,VALUE) --ADD'S ELEMENT AT PARTICULAR INDEX AND SHIFT REST ELEMENTS OF VECTOR
20     arr_obj.add(1,'M');
21     arr_obj.set(2,'V');
22     //Set Replaces the element at the specified position in VECTOR with the specified element.
23     //*****USING LIST ITERATOR TO TRAVEL FORWARD[next()] & BACKWARD[previous()] IN VECTOR*****
24
25     ListIterator obj = arr_obj.listIterator();
26     System.out.println("USING LIST ITERATOR TO TRAVEL FORWARD BY USING next() METHOD");
27     while (obj.hasNext())
28     {
29         Character O1 = (Character) obj.next(); //****next Allows you to Travel Forward Direction
30         System.out.println(O1);
31     }
32     System.out.println("USING LIST ITERATOR TO TRAVEL BACKWARD BY USING previous() METHOD");
33     while (obj.hasPrevious())
34     {
35         Character O1 = (Character) obj.previous(); //****next Allows you to Travel Backward Direction
36         System.out.println(O1);
37     }
38 }
39 }
```

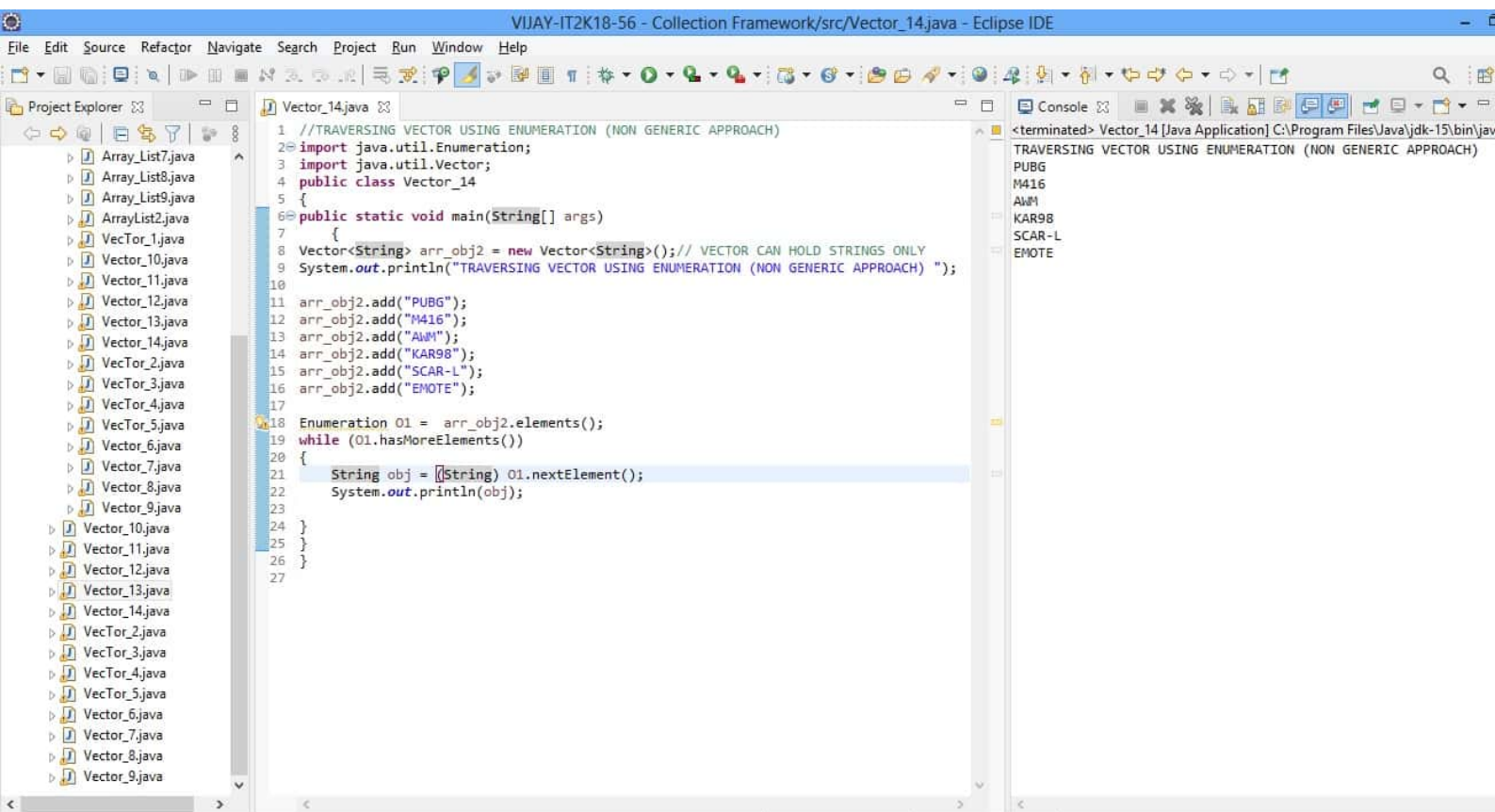
Console

```
<terminated> Vector_11 [Java Application] C:\Program Files\Java\jdk-15\bin\javaw.exe (21 Mar
TRAVERSING VECTOR ( GENERIC APPROACH) USING 3)ListIterator
USING LIST ITERATOR TO TRAVEL FORWARD BY USING next() METHOD
I
M
V
I
J
A
Y
USING LIST ITERATOR TO TRAVEL BACKWARD BY USING previous() METHOD
Y
A
J
I
V
M
I
```

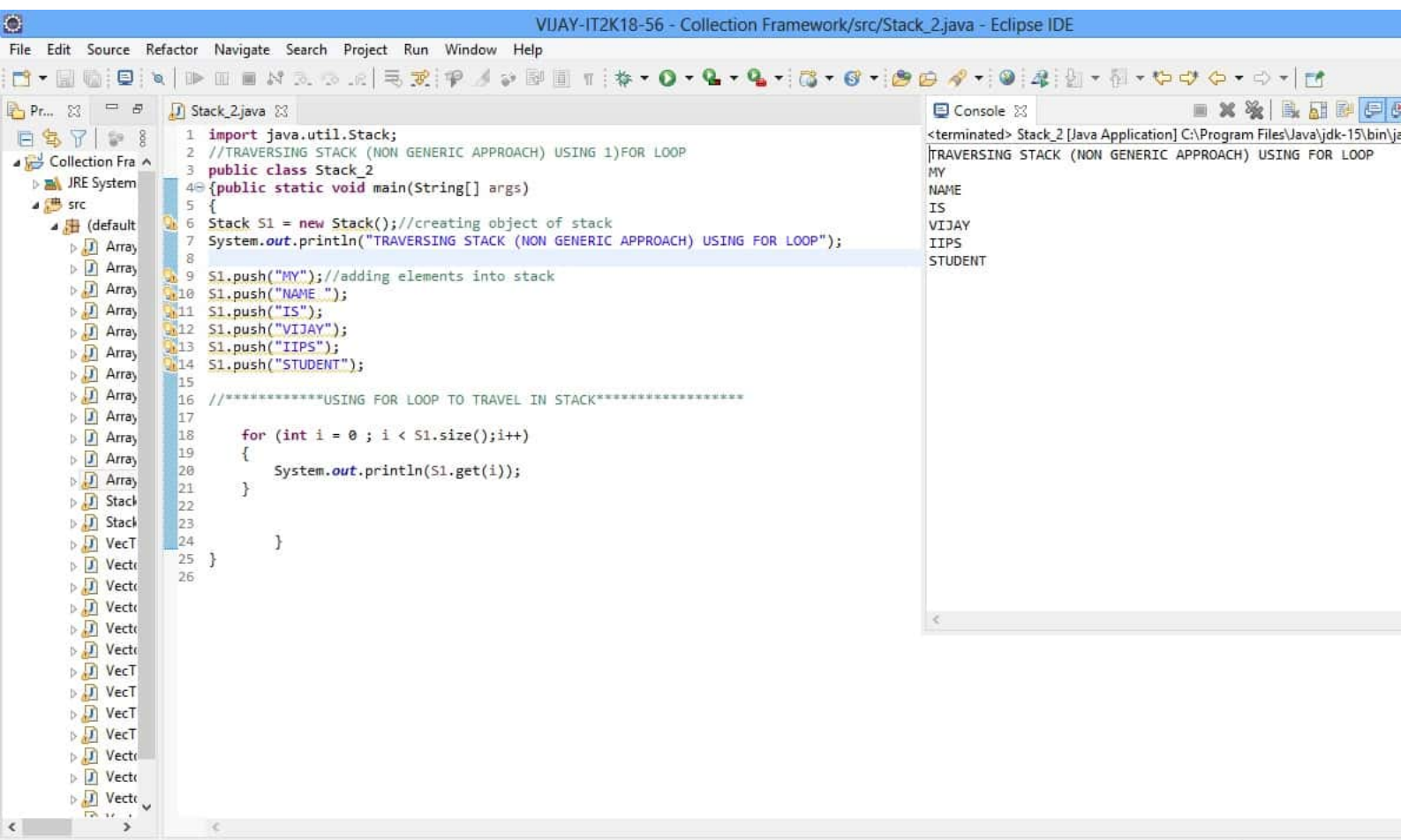


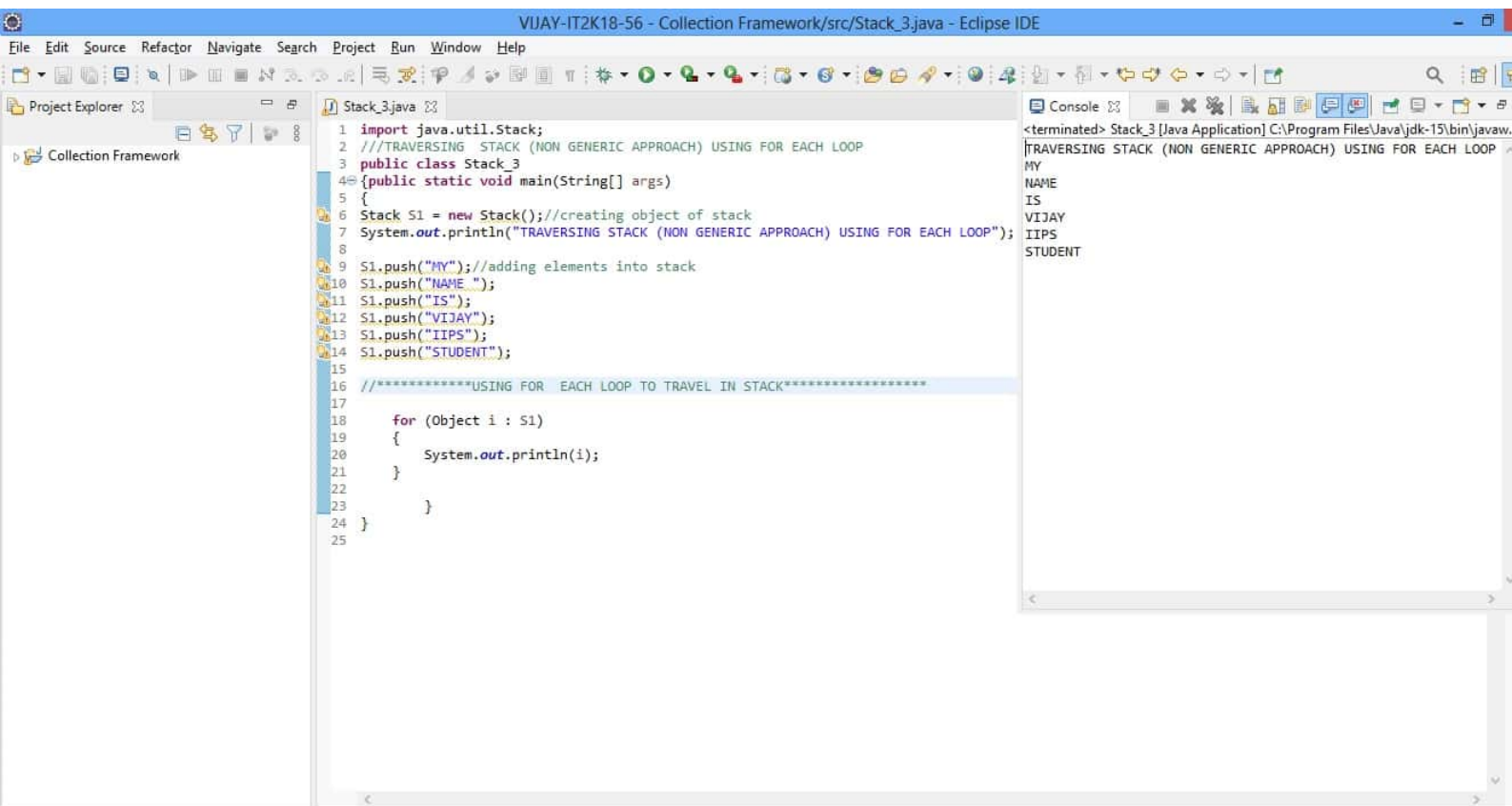










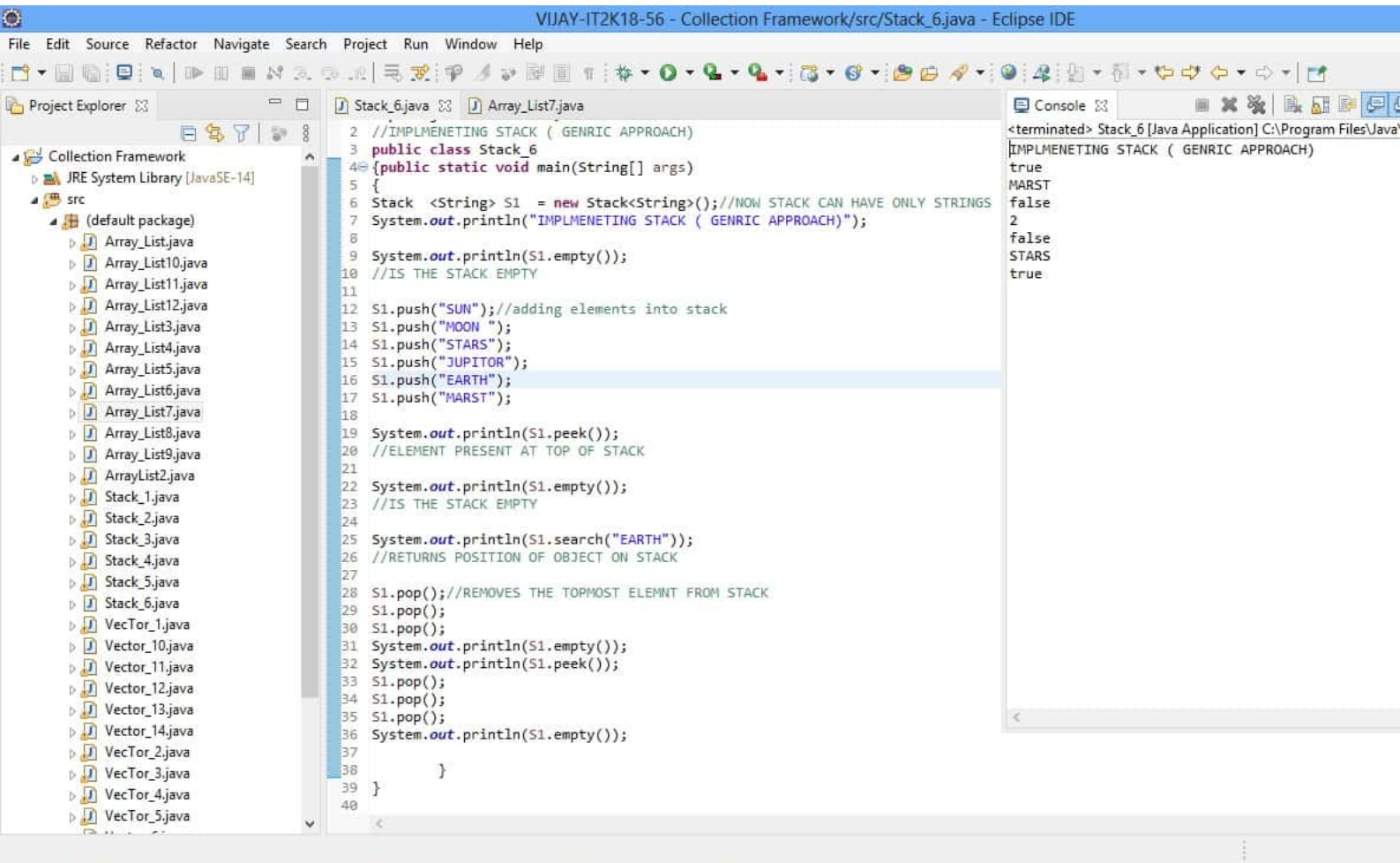


VIJAY-IT2K18-56 - Collection Framework/src/Stack_5.java - Eclipse IDE

```
1 import java.util.Iterator;
2 import java.util.ListIterator;
3 import java.util.Stack;
4 //TRAVERSING STACK (NON GENERIC APPROACH) USING LISTITERATOR
5 public class Stack_5
6 {
7     public static void main(String[] args)
8     {
9         Stack S1 = new Stack(); //creating object of stack
10        System.out.println("TRAVERSING STACK (NON GENERIC APPROACH) USING LIST ITERATOR");
11
12        S1.push("MY"); //adding elements into stack
13        S1.push("NAME ");
14        S1.push("IS");
15        S1.push("VIJAY");
16        S1.push("IIPS");
17        S1.push("STUDENT");
18
19        //*****USING LIST ITERATOR TO TRAVEL FORWARD[next()] & BACKWARD[previous()] IN STACK*****
20
21        ListIterator obj = S1.listIterator();
22
23        System.out.println("USING LIST ITERATOR TO TRAVEL FORWARD BY USING next() METHOD");
24        while (obj.hasNext())
25        {
26            Object O1 = (Object) obj.next(); //****next Allows you to Travel Forward Direction
27
28            System.out.println(O1);
29        }
30
31        System.out.println("USING LIST ITERATOR TO TRAVEL BACKWARD BY USING previous() METHOD");
32        while (obj.hasPrevious())
33        {
34            Object O1 = (Object) obj.previous(); //****next Allows you to Travel Backward Direction
35
36            System.out.println(O1);
37        }
38    }
39 }
```

Console

```
<terminated> Stack_5 [Java Application] C:\Program Files\Java\jdk-15\bin\javaw.exe (21
TRAVERSING STACK (NON GENERIC APPROACH) USING LIST ITERATOR
USING LIST ITERATOR TO TRAVEL FORWARD BY USING next() METHOD
MY
NAME
IS
VIJAY
IIPS
STUDENT
USING LIST ITERATOR TO TRAVEL BACKWARD BY USING previous() METHOD
STUDENT
IIPS
VIJAY
IS
NAME
MY
```



VIJAY-IT2K18-56 - Collection Framework/src/Stack_7.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

Project Explorer

- Array_List4.java
- Array_List5.java
- Array_List6.java
- Array_List7.java
- Array_List8.java
- Array_List9.java
- ArrayList2.java
- Stack_1.java
- Stack_2.java
- Stack_3.java
- Stack_4.java
- Stack_5.java
- Stack_6.java
- Stack_7.java
- VecTor_1.java
- Vector_10.java
- Vector_11.java
- Vector_12.java
- Vector_13.java
- Vector_14.java
- VecTor_2.java
- VecTor_3.java
- VecTor_4.java
- VecTor_5.java
- Vector_6.java
- Vector_7.java
- Vector_8.java
- Vector_9.java
- Stack_4.java
- Stack_5.java
- Stack_6.java
- Stack_7.java

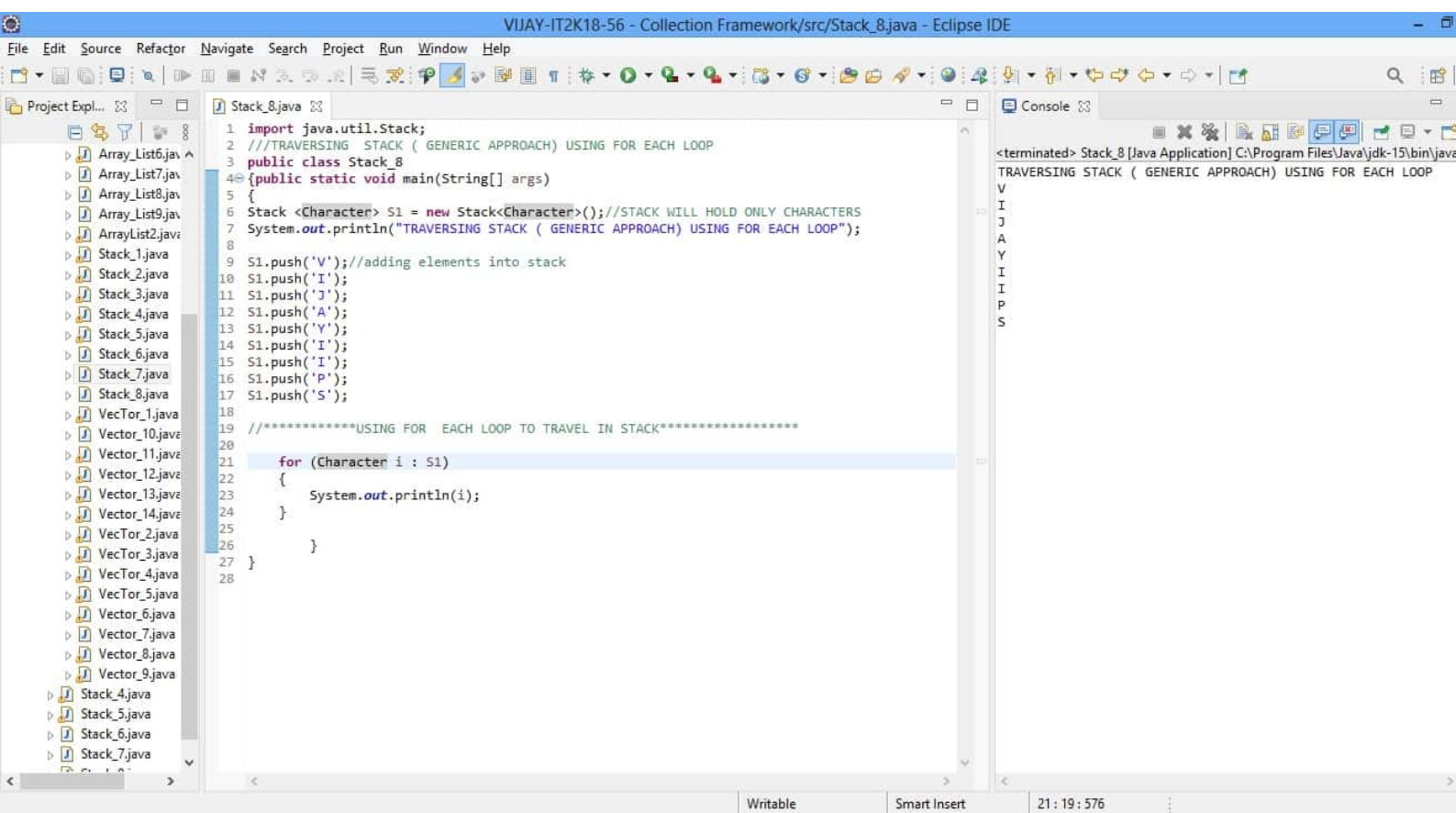
Stack_2.java Stack_7.java Stack_6.java

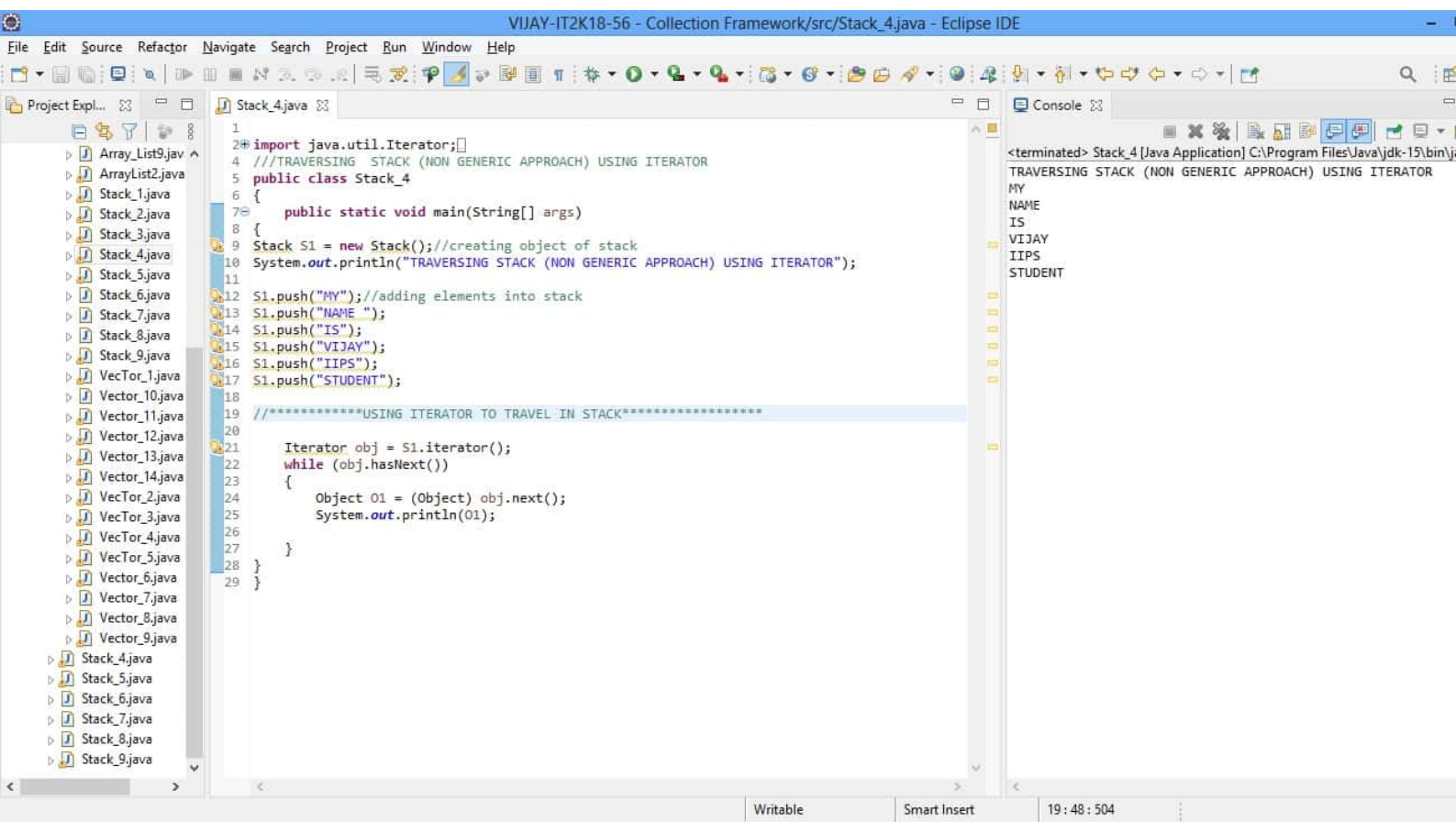
```
1 import java.util.Stack;
2 //TRAVERSING STACK ( GENERIC APPROACH) USING 1)FOR LOOP
3 public class Stack_7
4 {
5     public static void main(String[] args)
6     {
7         Stack <Integer> S1 = new Stack<Integer>();//NOW STACK CAN HAVE ONLY INTEGERS
8         System.out.println("TRAVERSING STACK (GENERIC APPROACH) USING FOR LOOP");
9
10        S1.push(56);//adding elements into stack
11        S1.push(569);
12        S1.push(69);
13        S1.push(182);
14        S1.push(94);
15        S1.push(129);
16
17        //*****USING FOR LOOP TO TRAVEL IN STACK*****
18
19        for (int i = 0 ; i < S1.size();i++)
20        {
21            System.out.println(S1.get(i));
22        }
23
24    }
25
26 }
27
```

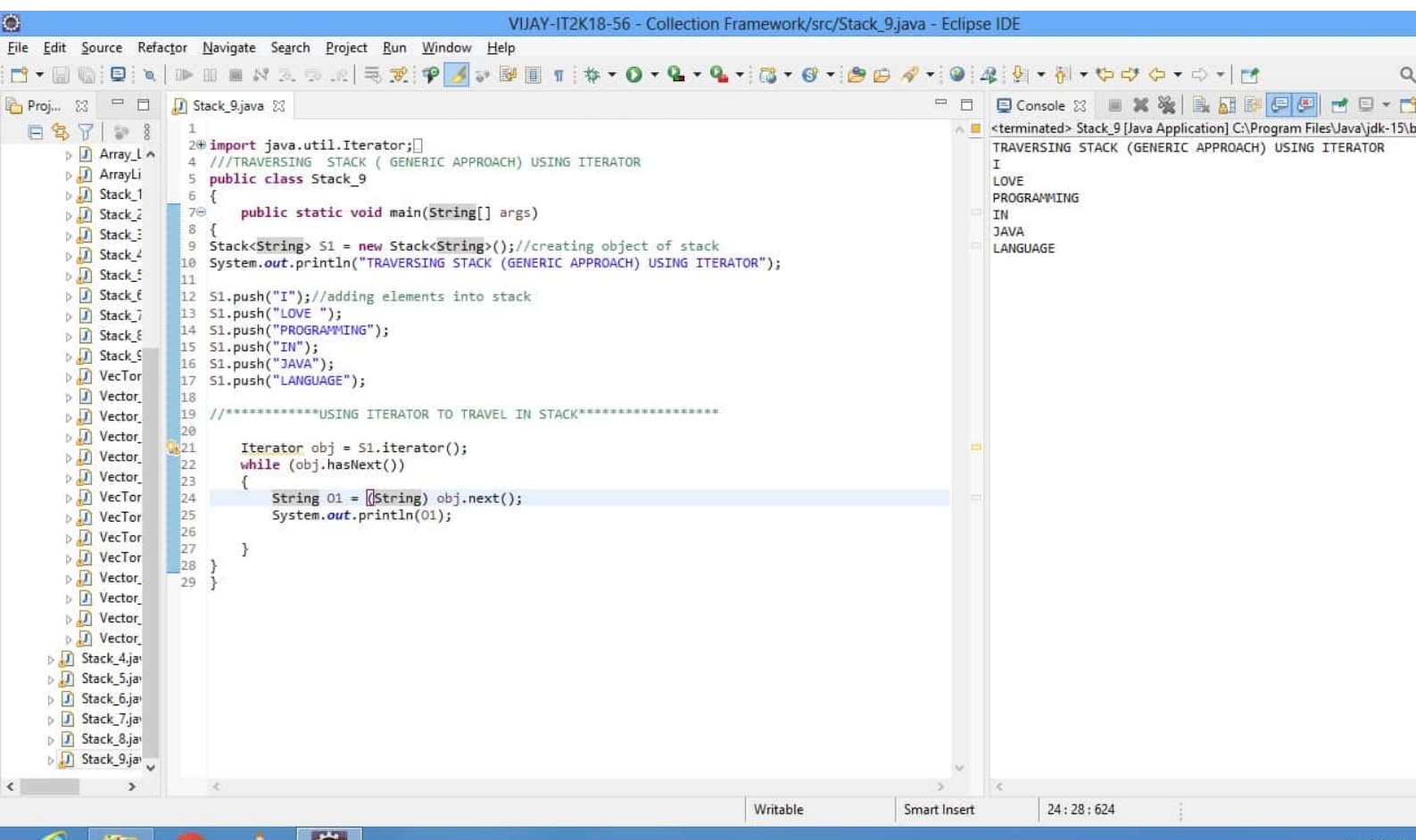
Console

Stack_7 [Java Application] C:\Program Files\Java\jdk-15\bin\javaw.exe
TRAVERSING STACK (GENERIC APPROACH) USING FOR LOOP
56
569
69
182
94
129

Writable Smart Insert 8 : 39 : 270





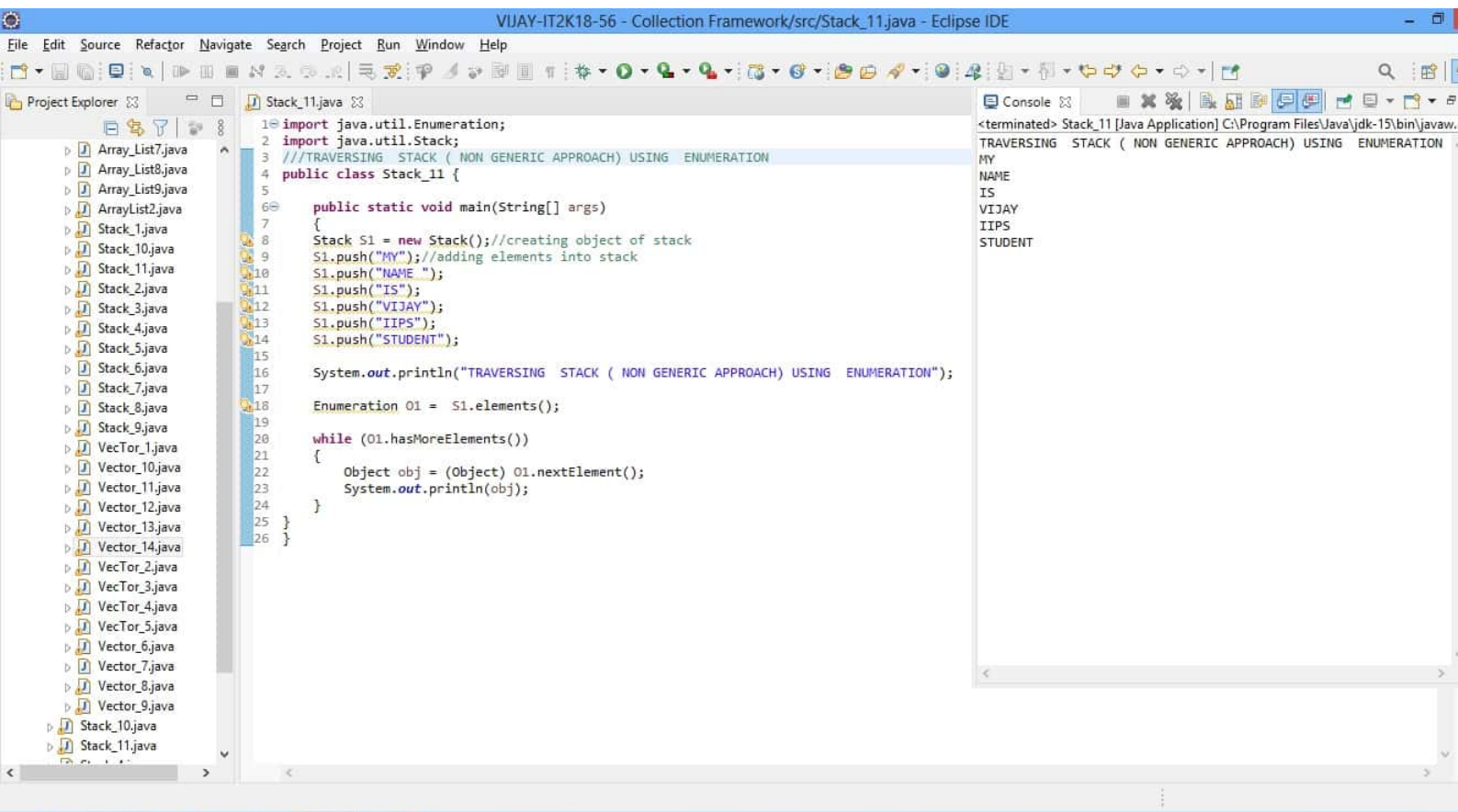


VIJAY-IT2K18-56 - Collection Framework/src/Stack_10.java - Eclipse IDE

```
File Edit Source Refactor Navigate Search Project Run Window Help
Stack_10.java
1 import java.util.ListIterator;
2 import java.util.Stack;
3 ///TRAVERSING STACK ( GENERIC APPROACH) USING LISTITERATOR
4 public class Stack_10
5 {
6     public static void main(String[] args)
7     {
8         Stack<Double> S1 = new Stack<Double>(); //Stack will hold only double values
9         System.out.println("TRAVERSING STACK ( GENERIC APPROACH) USING LIST ITERATOR");
10
11         S1.push(11.1); //adding elements into stack
12         S1.push(11.2);
13         S1.push(11.025);
14         S1.push(11.96);
15         S1.push(6.0222);
16         S1.push(52.1);
17
18         //*****USING LIST ITERATOR TO TRAVEL FORWARD[next()] & BACKWARD[previous()] IN STACK*****
19
20         ListIterator obj = S1.listIterator();
21
22         System.out.println("USING LIST ITERATOR TO TRAVEL FORWARD BY USING next() METHOD");
23         while (obj.hasNext())
24         {
25             Double O1 = (Double) obj.next(); //****next Allows you to Travel Forward Direction
26
27             System.out.println(O1);
28         }
29
30         System.out.println("USING LIST ITERATOR TO TRAVEL BACKWARD BY USING previous() METHOD");
31         while (obj.hasPrevious())
32         {
33             Double O1 = (Double) obj.previous(); //****next Allows you to Travel Backward Direction
34
35             System.out.println(O1);
36         }
37     }
38 }
```

Console

```
<terminated> Stack_10 [Java Application] C:\Program Files\Java\jdk-15\bin\javaw.exe (21 M
TRAVERSING STACK ( GENERIC APPROACH) USING LIST ITERATOR
USING LIST ITERATOR TO TRAVEL FORWARD BY USING next() METHOD
11.1
11.2
11.025
11.96
6.0222
52.1
USING LIST ITERATOR TO TRAVEL BACKWARD BY USING previous() METHOD
52.1
6.0222
11.96
11.025
11.2
11.1
```



VIJAY-IT2K18-56 - Collection Framework/src/Stack_12.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

Stack_12.java

```
1 import java.util.Enumeration;
2 import java.util.Stack;
3 ///TRAVERSING STACK ( GENERIC APPROACH) USING ENUMERATION
4 public class Stack_12 {
5
6     public static void main(String[] args)
7     {
8         Stack <Integer>S1 = new Stack<Integer>();//Stack can hold only integer values
9         S1.push(98);//adding elements into stack
10        S1.push(99);
11        S1.push(100);
12        S1.push(101);
13        S1.push(102);
14        S1.push(103);
15
16        System.out.println("TRAVERSING STACK ( GENERIC APPROACH) USING ENUMERATION");
17
18        Enumeration O1 = S1.elements();
19
20        while (O1.hasMoreElements())
21        {
22            Integer obj = (Integer) O1.nextElement();
23            System.out.println(obj);
24        }
25    }
26 }
```

Console

```
<terminated> Stack_12 [Java Application] C:\Program Files\Java\jdk-15\bin\java
TRAVERSING STACK ( GENERIC APPROACH) USING ENUMERATION
98
99
100
101
102
103
```

Writable Smart Insert 8 : 82 : 269

VIJAY-IT2K18-56 - Collection Framework/src/Linked_List1.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

Linked_List1.java

```
1 import java.util.LinkedList;
2
3 //IMPLEMENTING LINKEDLIST PROGRAM (NON GENERIC APPROACH)
4
5 public class Linked_List1
6 {
7     public static void main(String[] args)
8     {
9         LinkedList arr_obj = new LinkedList();//CREATING OBJECT OF LINKEDLIST
10        System.out.println("IMPLEMENTING LINKEDLIST PROGRAM (NON GENERIC APPROACH)");
11        System.out.println(arr_obj.isEmpty());//THIS METHOD CHECKS IS IT EMPTY AND RETURNS BOOLEAN VALUE
12        System.out.println(arr_obj.size());//RETURNS INTEGER COUNT OF NUMBER OF ELEMENTS IN LINKEDLIST
13
14        arr_obj.add("VIJAY");// ADD() METHOD ADDS ELEMENT IN THE LINKEDLIST
15        arr_obj.add("ROLL NO");
16        arr_obj.add(56);
17
18        arr_obj.add(0,"HI");
19        //ADD ( INDEX,VALUE) --ADD's ELEMENT AT PARTICULAR INDEX AND SHIFT REST ELEMENTS OF LINKEDLIST
20
21        System.out.println(arr_obj.isEmpty());//THIS METHOD CHECKS IS IT EMPTY AND RETURNS BOOLEAN VALUE
22        System.out.println(arr_obj);
23
24        arr_obj.set(0, "IIPS");
25        //Replaces the element at the specified position in LINKEDLIST with the specified element.
26        System.out.println(arr_obj);
27
28        System.out.println(arr_obj.size());
29        //RETURNS INT VALUE WHICH TELLS COUNT OF ELEMENT PRESENT IN LINKEDLIST
30
31        System.out.println(arr_obj.contains("AJAY"));
32        //RETURNS BOOLEAN VALUE THAT TELLS ELEMENT EXIST IN LINKEDLIST OR NOT ?
33        System.out.println(arr_obj.contains("VIJAY"));
34
35        System.out.println(arr_obj.get(3));//RETURNS ELEMENT AT PARTICULAR INDEX
36
37    }
38
39 }
```

Console

```
<terminated> Linked_List1 [Java Application] C:\Program Files\Java\jdk-15\bin\jav
IMPLEMENTING LINKEDLIST PROGRAM (NON GENERIC APPROACH)
true
0
false
[HI, VIJAY, ROLL NO, 56]
[IIPS, VIJAY, ROLL NO, 56]
4
false
true
56
```


VIJAY-IT2K18-56 - Collection Framework/src/Linked_List2.java - Eclipse IDE

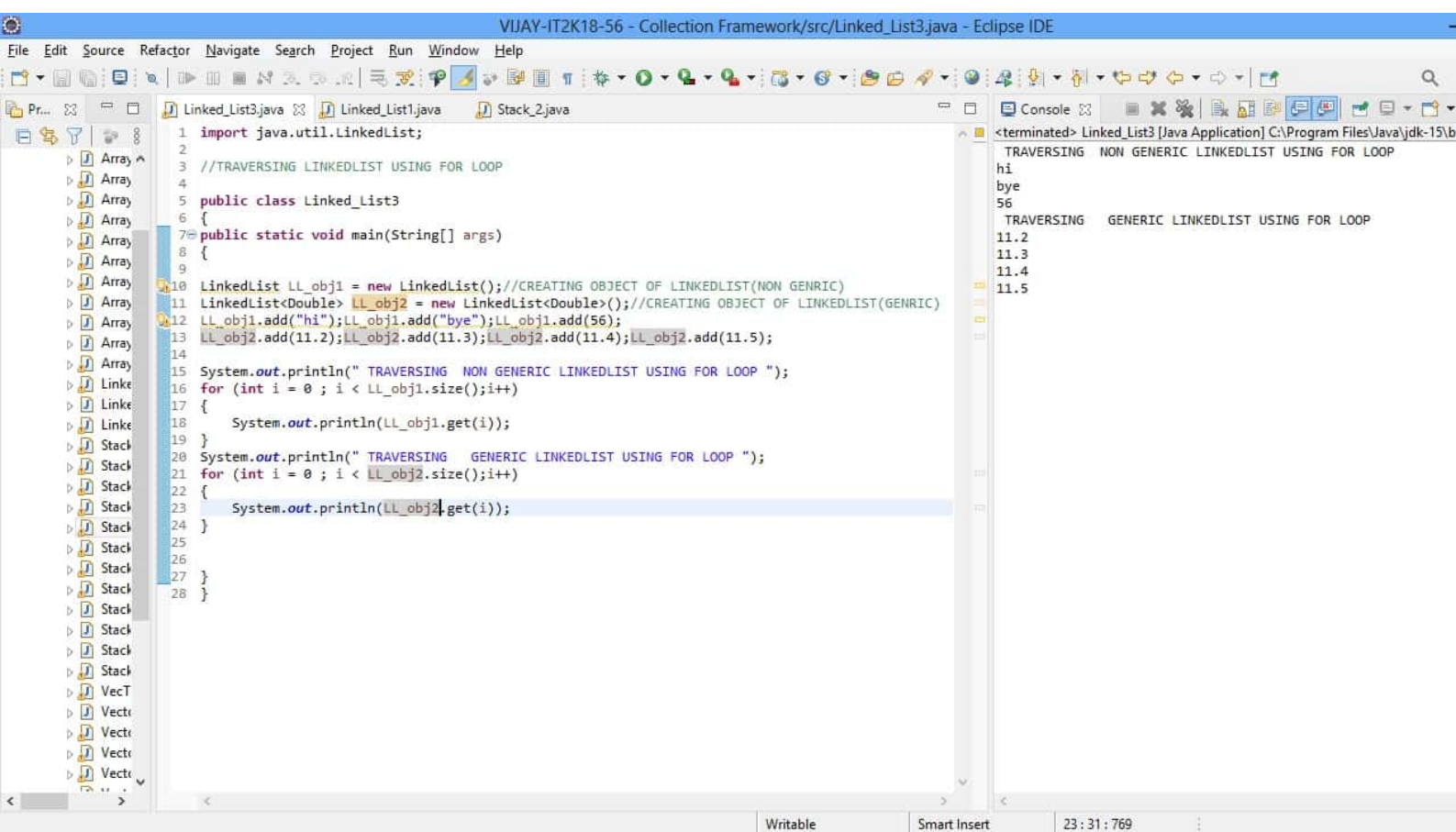
File Edit Source Refactor Navigate Search Project Run Window Help

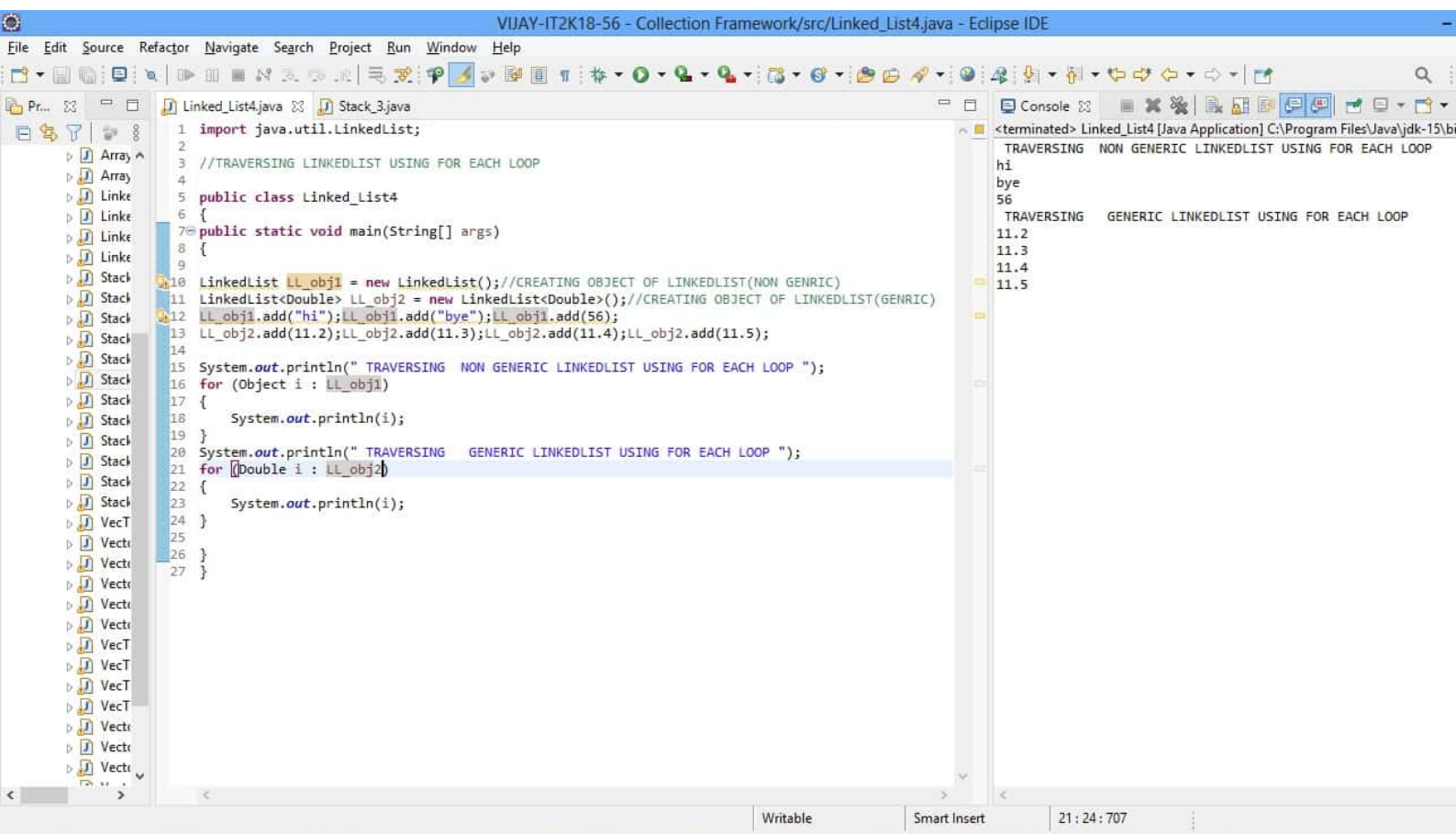
Linked_List2.java Linked_List1.java

```
1 import java.util.LinkedList;
2
3 //IMPLEMENTING LINKEDLIST PROGRAM ( GENERIC APPROACH)
4
5 public class Linked_List2
6 {
7     public static void main(String[] args)
8     {
9         LinkedList<Integer> arr_obj = new LinkedList<Integer>();//LINKEDLIST will hold INTEGER ONLY
10        System.out.println("IMPLEMENTING LINKEDLIST PROGRAM ( GENERIC APPROACH)");
11        System.out.println(arr_obj.isEmpty());//THIS METHOD CHECKS IS IT EMPTY AND RETURNS BOOLEAN VALUE
12        System.out.println(arr_obj.size());//RETURNS INTEGER COUNT OF NUMBER OF ELEMENTS IN LINKEDLIST
13
14        arr_obj.add(99);// ADD() METHOD ADDS ELEMENT IN THE LINKEDLIST
15        arr_obj.add(95);
16        arr_obj.add(56);
17
18        arr_obj.add(0,87);
19        //ADD ( INDEX,VALUE) --ADD'S ELEMENT AT PARTICULAR INDEX AND SHIFT REST ELEMENTS OF LINKEDLIST
20
21        System.out.println(arr_obj.isEmpty());//THIS METHOD CHECKS IS IT EMPTY AND RETURNS BOOLEAN VALUE
22        System.out.println(arr_obj);
23
24        arr_obj.set(0, 68);
25        //Replaces the element at the specified position in LINKEDLIST with the specified element.
26        System.out.println(arr_obj);
27
28        System.out.println(arr_obj.size());
29        //RETURNS INT VALUE WHICH TELLS COUNT OF ELEMENT PRESENT IN LINKEDLIST
30
31        System.out.println(arr_obj.contains(100));
32        //RETURNS BOOLEAN VALUE THAT TELLS ELEMENT EXIST IN LINKEDLIST OR NOT ?
33        System.out.println(arr_obj.contains(56));
34
35        System.out.println(arr_obj.get(3));//RETURNS ELEMENT AT PARTICULAR INDEX
36
37    }
38
39 }
```

Console

```
<terminated> Linked_List2 [Java Application] C:\Program Files\Java\jdk-15\bin
IMPLEMENTING LINKEDLIST PROGRAM ( GENERIC APPROACH)
true
0
false
[87, 99, 95, 56]
[68, 99, 95, 56]
4
false
true
56
```





VIJAY-IT2K18-56 - Collection Framework/src/Linked_List5.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

Stack_4.java Linked_List5.java

```
1 import java.util.Iterator;
2 import java.util.LinkedList;
3
4 //TRAVERSING LINKEDLIST USING FOR ITERATOR
5
6 public class Linked_List5
7 {
8     public static void main(String[] args)
9     {
10
11         LinkedList LL_obj1 = new LinkedList();//CREATING OBJECT OF LINKEDLIST(NON GENRIC)
12         LinkedList<Double> LL_obj2 = new LinkedList<Double>();//CREATING OBJECT OF LINKEDLIST(GENRIC)
13         LL_obj1.add("SUN");LL_obj1.add("MOON");LL_obj1.add(100);
14         LL_obj2.add(6.022);LL_obj2.add(60.22);LL_obj2.add(602.2);LL_obj2.add(6022.0);
15
16         System.out.println("TRAVERSING NON GENERIC LINKEDLIST USING ITERATOR ");
17         Iterator obj = LL_obj1.iterator();
18         while (obj.hasNext())
19         {
20             Object O1 = (Object) obj.next();
21             System.out.println(O1);
22         }
23         System.out.println(" TRAVERSING  GENERIC LINKEDLIST USING ITERATOR ");
24
25         Iterator obj1 = LL_obj2.iterator();
26         while (obj1.hasNext())
27         {
28             Double O1 = (Double) obj1.next();
29             System.out.println(O1);
30         }
31     }
32 }
33 }
34 }
```

Console

```
<terminated> Linked_List5 [Java Application] C:\Program Files\Java\jdk-15
TRAVERSING NON GENERIC LINKEDLIST USING ITERATOR
SUN
MOON
100
TRAVERSING  GENERIC LINKEDLIST USING ITERATOR
6.022
60.22
602.2
6022.0
```

Writable Smart Insert 29 : 24 : 865

VIJAY-IT2K18-56 - Collection Framework/src/Linked_List6.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

Linked_List6.java Stack_5.java

```
1 import java.util.Iterator;
2 import java.util.LinkedList;
3 import java.util.ListIterator;
4 //TRAVERSING LINKEDLIST USING FOR LIST ITERATOR
5 public class Linked_List6
6 {
7     public static void main(String[] args)
8     {
9         LinkedList LL_obj1 = new LinkedList();//CREATING OBJECT OF LINKEDLIST(NON GENERIC)
10        LinkedList<Character> LL_obj2 = new LinkedList<Character>();//CREATING OBJECT OF LINKEDLIST(GENERIC)
11        LL_obj1.add("I AM ");LL_obj1.add("STUDENT ");LL_obj1.add("OF IIPS");
12        LL_obj2.add('V');LL_obj2.add('I');LL_obj2.add('J');LL_obj2.add('A');LL_obj2.add('Y');
13        System.out.println("*****TRAVERSING NON GENERIC LINKEDLIST USING LIST ITERATOR*****");
14        ListIterator obj = LL_obj1.listIterator();
15        System.out.println("USING LIST ITERATOR TO TRAVEL FORWARD BY USING next() METHOD");
16        while (obj.hasNext())
17        {Object O1 = (Object) obj.next(); //****next Allows you to Travel Forward Direction
18        System.out.println(O1);
19        }
20        System.out.println("USING LIST ITERATOR TO TRAVEL BACKWARD BY USING previous() METHOD");
21        while (obj.hasPrevious())
22        {Object O1 = (Object) obj.previous(); //****next Allows you to Travel Backward Direction
23        System.out.println(O1);
24        }
25        System.out.println();System.out.println();System.out.println();
26        System.out.println("*****TRAVERSING GENERIC LINKEDLIST USING LIST ITERATOR*****");
27        ListIterator obj2 = LL_obj2.listIterator();
28        System.out.println("USING LIST ITERATOR TO TRAVEL FORWARD BY USING next() METHOD");
29        while (obj2.hasNext())
30        {Character O1 = (Character) obj2.next(); //****next Allows you to Travel Forward Direction
31        System.out.println(O1);
32        }
33        System.out.println("USING LIST ITERATOR TO TRAVEL BACKWARD BY USING previous() METHOD");
34        while (obj2.hasPrevious())
35        {Character O1 = (Character) obj2.previous(); //****next Allows you to Travel Backward Direction
36        System.out.println(O1);
37        }
38        }
39    }
```

Console

```
<terminated> Linked_List6 [Java Application] C:\Program Files\Java\jdk-15\bin\javaw.exe (2
*****TRAVERSING NON GENERIC LINKEDLIST USING LIST ITERATOR*****
USING LIST ITERATOR TO TRAVEL FORWARD BY USING next() METHOD
I AM
STUDENT
OF IIPS
USING LIST ITERATOR TO TRAVEL BACKWARD BY USING previous() METHOD
OF IIPS
STUDENT
I AM

*****TRAVERSING GENERIC LINKEDLIST USING LIST ITERATOR*****
USING LIST ITERATOR TO TRAVEL FORWARD BY USING next() METHOD
V
I
J
A
Y
USING LIST ITERATOR TO TRAVEL BACKWARD BY USING previous() METHOD
Y
A
J
I
V
```

Writable Smart Insert 25:64:1218

vijayyy - COLLECTION/src/Array_Deque1.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

```
Array_Deque1.java
1 import java.util.ArrayDeque;
2 import java.util.Iterator;
3 public class Array_Deque1
4 {
5     public static void main(String[] args)
6     {
7         System.out.println("PROGRAM ON ARRAY DEQUE (GENERIC APPROACH)");
8         ArrayDeque<String> deque = new ArrayDeque<String>();
9         //Creating Deque and adding elements
10        deque.add("HELLOW");
11        deque.add("Vijay");
12        deque.add("Ajay");
13        System.out.println(" Traversing elements using FOR EACH LOOP");
14        for (String str : deque)
15        { System.out.println(str); }
16
17        System.out.println(" Traversing elements using ITERATOR");
18        Iterator obj = deque.iterator();
19        while (obj.hasNext()) {
20            String O1 = (String) obj.next();
21            System.out.println(O1);
22        }
23        System.out.println();System.out.println();System.out.println();
24        System.out.println("PROGRAM ON ARRAY DEQUE ( NON GENERIC APPROACH)");
25        ArrayDeque deque1 = new ArrayDeque();
26        //Creating Deque and adding elements
27        deque1.add("HELLOW");
28        deque1.add("Vijay");
29        deque1.add(56);
30        System.out.println(" Traversing elements using FOR EACH LOOP");
31        for (Object objj : deque1)
32        { System.out.println(objj); }
33        System.out.println(" Traversing elements using ITERATOR");
34        Iterator obj2 = deque1.iterator();
35        while (obj2.hasNext()) {
36            Object O1 = (Object) obj2.next();
37            System.out.println(O1);
38        }
39    }
}
```

Problems @ Javadoc Declaration Console

<terminated> Array_Deque1 [Java Application] D:\JAVA 15\bin\javaw.exe (22-Mar-2021, 9

PROGRAM ON ARRAY DEQUE (GENERIC APPROACH)

Traversing elements using FOR EACH LOOP

HELLOW

Vijay

Ajay

Traversing elements using ITERATOR

HELLOW

Vijay

Ajay

PROGRAM ON ARRAY DEQUE (NON GENERIC APPROACH)

Traversing elements using FOR EACH LOOP

HELLOW

Vijay

56

Traversing elements using ITERATOR

HELLOW

Vijay

56

vijjayyy - COLLECTION/src/Array_Deque2.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

Array_Deque1.java Array_Deque2.java

```
1 import java.util.ArrayDeque;
2
3
4 public class Array_Deque2
5 {
6     public static void main(String[] args) {
7         ArrayDeque<String> deque = new ArrayDeque<String>();
8         deque.offer("good");
9         deque.offer("bad");
10        deque.add("worst");
11        deque.offerFirst("best");
12        System.out.println("Java ArrayDeque Example: offerFirst() and pollLast()");
13        System.out.println();System.out.println();
14        System.out.println("After offerFirst Traversal...");
15        for(String s:deque)
16        {
17            System.out.println(s);
18        }
19        //deque.poll();
20        //deque.pollFirst();//it is same as poll()
21        System.out.println();System.out.println();
22        deque.pollLast();
23        System.out.println("After pollLast() Traversal...");
24        for(String s:deque){
25            System.out.println(s);
26        }
27    }
28 }
29
30
```

Problems Javadoc Declaration Console

<terminated> Array_Deque2 [Java Application] D:\JAVA 15\bin\javaw.exe

Java ArrayDeque Example: offerFirst() and pollLast()

After offerFirst Traversal...

best
good
bad
worst

After pollLast() Traversal...

best
good
bad

vijayyy - COLLECTION/src/Priority_Queue.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

Priority_Queue.java Array_Deque1.java

```
1 import java.util.Iterator;
2 import java.util.PriorityQueue;
3
4
5
6 public class Priority_Queue
7 {
8
9     public static void main(String args[]){
10         PriorityQueue<String> queue=new PriorityQueue<String>();
11         System.out.println("PROGRAM ON Priority Queue (Genric Approach)");
12         queue.add("SUN");
13         queue.add("MOON");
14         queue.add("STAR");
15         queue.add("EARTH");
16         queue.add("JUPITOR");
17         System.out.println(" Traversing elements using FOR EACH LOOP");
18         for (String str : queue)
19             { System.out.println(str); }
20         System.out.println();System.out.println();
21         System.out.println("head:"+queue.element());
22         System.out.println("head:"+queue.peek());
23         System.out.println();System.out.println();
24         System.out.println("Traversing elements using Iterator");
25         Iterator itr=queue.iterator();
26         while(itr.hasNext()){
27             System.out.println(itr.next());
28         }
29         queue.remove();
30         queue.poll();
31         System.out.println();System.out.println();
32         System.out.println("ON REMOVING 2 ELEMENTS");
33         Iterator<String> itr2=queue.iterator();
34         while(itr2.hasNext()){
35             System.out.println(itr2.next());
36         }
37     }
38 }
39
```

Problems Javadoc Declaration Console

<terminated> Priority_Queue [Java Application] D:\JAVA 15\bin\javaw.exe (

PROGRAM ON Priority Queue (Genric Approach)

Traversing elements using FOR EACH LOOP

EARTH
JUPITOR
STAR
SUN
MOON

head:EARTH
head:EARTH

Traversing elements using Iterator

EARTH
JUPITOR
STAR
SUN
MOON

ON REMOVING 2 ELEMENTS

MOON
SUN
STAR

vijjayyy - COLLECTION/src/Priority_Queue2.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

Priority_Queue2.java

```
1 import java.util.Iterator;
2 import java.util.PriorityQueue;
3 public class Priority_Queue2
4 {
5     public static void main(String args[]){
6         PriorityQueue queue=new PriorityQueue();
7         System.out.println("PROGRAM ON Priority Queue ( NON Genric Approach)");
8         queue.add("100");
9         queue.add("MOON");
10        queue.add("56");
11        queue.add("EARTH");
12        queue.add("99");
13        System.out.println(" Traversing elements using FOR EACH LOOP");
14        for (Object str : queue)
15            { System.out.println(str); }
16        System.out.println();System.out.println();
17        System.out.println("head:"+queue.element());
18        System.out.println("head:"+queue.peek());
19        System.out.println();System.out.println();
20        System.out.println("Traversing elements using Iterator");
21        Iterator obj = queue.iterator();
22        while (obj.hasNext()) {
23            Object O1 = (Object) obj.next();
24            System.out.println(O1);
25        }
26        queue.remove();
27        queue.poll();
28        System.out.println();System.out.println();
29        System.out.println("ON REMOVING 2 ELEMENTS");
30        Iterator obj2 = queue.iterator();
31        while (obj2.hasNext()) {
32            Object O1 = (Object) obj2.next();
33            System.out.println(O1);
34        }
35    }
36 }
37 }
38 }
39 }
```

Console

<terminated> Priority_Queue2 [Java Application] D:\JAVA 15\bin\javaw.exe

PROGRAM ON Priority Queue (NON Genric Approach)

Traversing elements using FOR EACH LOOP

100
99
56
MOON
EARTH

head:100
head:100

Traversing elements using Iterator

100
99
56
MOON
EARTH

ON REMOVING 2 ELEMENTS

99
MOON
EARTH

vijjayyy - COLLECTION/src/Hash_Set1.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

Hash_Set1.java

```
1 import java.util.HashSet;
2 import java.util.Iterator;
3
4 public class Hash_Set1
5 {
6     public static void main(String args[])
7     {
8         System.out.println("PROGRAM ON HASHSET (GENRIC APPROACH)");
9         System.out.println();System.out.println();System.out.println();
10
11         HashSet<String> set=new HashSet();
12         //Creating HashSet and adding elements
13         set.add("HI");
14         set.add("HII");
15         set.add("HIII");
16         set.add("HIIII");
17         set.add("HIIIIII");
18
19         System.out.println(" Traversing elements using FOR EACH LOOP");
20         for (String str : set)
21         { System.out.println(str); }
22         System.out.println();System.out.println();System.out.println();
23
24         System.out.println(" Traversing elements using ITERATOR");
25         Iterator<String> i=set.iterator();
26         while(i.hasNext())
27         {
28             System.out.println(i.next());
29         }
30         System.out.println();System.out.println();System.out.println();
31         System.out.println("REMOVING ELEMENTS HI and HIII");
32         set.remove("HI");
33         set.remove("HIII");
34         System.out.println(" NOW ELEMENTS LEFT ARE ");
35         for (String str : set)
36         { System.out.println(str); }
37     }
38 }
39
```

Console

<terminated> Hash_Set1 [Java Application] D:\JAVA 15\bin\javaw.exe (22-Ma
PROGRAM ON HASHSET (GENRIC APPROACH)

Traversing elements using FOR EACH LOOP

HIII
HIIII
HI
HIIIIII
HII

Traversing elements using ITERATOR

HIII
HIIII
HI
HIIIIII
HII

REMOVING ELEMENTS HI and HIII
NOW ELEMENTS LEFT ARE

HIIII
HIIIIII
HII

vijjayyy - COLLECTION/src/Tree_Set1.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

Tree_Set1.java

```
1 import java.util.Iterator;
2
3 class Tree_Set1
4 {
5     public static void main(String args[]){
6         TreeSet<String> set=new TreeSet<String>();
7
8         System.out.println("PROGRAM ON TREESSET (GENRIC APPROACH)");
9         System.out.println();System.out.println();System.out.println();
10         set.add("I");
11         set.add("LOVE");
12         set.add("INDIA");
13
14         System.out.println(" Traversing elements using FOR EACH LOOP");
15
16         for (String str : set)
17         { System.out.println(str); }
18         System.out.println();System.out.println();System.out.println();
19
20         System.out.println("Traversing element through Iterator in descending order");
21         Iterator i=set.descendingIterator();
22         while(i.hasNext())
23         {
24             System.out.println(i.next());
25         }
26     }
27 }
28
29
```

Console

<terminated> Tree_Set1 [Java Application] D:\JAVA 15\bin\javaw.exe (22-Mar-2022 10:00:00 AM)

PROGRAM ON TREESSET (GENRIC APPROACH)

Traversing elements using FOR EACH LOOP

I

INDIA

LOVE

Traversing element through Iterator in descending order

LOVE

INDIA

I

Writable Smart Insert 5:1:74

vijayyy - COLLECTION/src/Linked_HashSet.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

```
Linked_HashSet.java
1 import java.util.Iterator;
2 import java.util.LinkedHashSet;
3 public class Linked_HashSet
4 { public static void main(String[] args)
5 {
6     LinkedHashSet linkedset = new LinkedHashSet();
7     System.out.println("PROGRAM ON LINKED HASH-SET ( NON  GENERIC APPROACH)");
8     System.out.println();System.out.println();
9     // Adding element to LinkedHashSet
10    linkedset.add("HELLOW");
11    linkedset.add("ROLL");
12    linkedset.add("NO");
13    linkedset.add(56);
14    linkedset.add("VIJAY");
15    System.out.println("Size of LinkedHashSet = " + linkedset.size());
16    System.out.println();System.out.println();
17    System.out.println("Original LinkedHashSet: " + linkedset);
18    System.out.println("Removing HELLOW from LinkedHashSet: " + linkedset.remove("HELLOW"));
19    System.out.println();System.out.println();
20    System.out.println("Trying to Remove AJAY which is not "+
21        "present: " + linkedset.remove("AJAY"));
22    System.out.println();System.out.println();
23    System.out.println("Checking if VIJAY is present=" +
24        linkedset.contains("VIJAY"));
25    System.out.println();System.out.println();
26    System.out.println("Updated LinkedHashSet: " + linkedset);
27    System.out.println();System.out.println();
28    System.out.println(" Traversing elements using FOR EACH LOOP");
29    for (Object objj : linkedset)
30    { System.out.println(objj); }
31
32    System.out.println(" Traversing elements using ITERATOR");
33    Iterator obj2 = linkedset.iterator();
34    while (obj2.hasNext()) {
35        Object o1 = (Object) obj2.next();
36        System.out.println(o1);
37    }
38 }
39 }
```

Console

<terminated> Linked_HashSet [Java Application] D:\JAVA 15\bin\javaw.exe (22-Mar-2021)
PROGRAM ON LINKED HASH-SET (NON GENERIC APPROACH)

Size of LinkedHashSet = 5

Original LinkedHashSet:[HELLOW, ROLL, NO, 56, VIJAY]
Removing HELLOW from LinkedHashSet: true

Trying to Remove AJAY which is not present: false

Checking if VIJAY is present=true

Updated LinkedHashSet: [ROLL, NO, 56, VIJAY]

Traversing elements using FOR EACH LOOP
ROLL
NO
56
VIJAY
Traversing elements using ITERATOR
ROLL
NO
56
VIJAY

vijjayyy - COLLECTION/src/Linked_HashSet.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help



Linked_HashSet.java

```
1 import java.util.Iterator;
2 import java.util.LinkedHashSet;
3 public class Linked_HashSet
4 { public static void main(String[] args)
5 {
6     LinkedHashSet<Character> linkedset = new LinkedHashSet<Character>();
7     System.out.println("PROGRAM ON LINKED HASH-SET ( GENRIC APPROACH)");
8     System.out.println();System.out.println();
9     // Adding element to LinkedHashSet
10     linkedset.add('V');
11     linkedset.add('I');
12     linkedset.add('J');
13     linkedset.add('A');
14     linkedset.add('Y');
15     System.out.println("Size of LinkedHashSet = " + linkedset.size());
16     System.out.println();System.out.println();
17     System.out.println("Original LinkedHashSet:" + linkedset);
18     System.out.println("Removing Y from LinkedHashSet: " + linkedset.remove('Y'));
19     System.out.println();System.out.println();
20     System.out.println("Trying to Remove Z which is not "+
21         "present: " + linkedset.remove('Z'));
22     System.out.println();System.out.println();
23     System.out.println("Checking if A is present=" +
24         linkedset.contains('A'));
25     System.out.println();System.out.println();
26     System.out.println("Updated LinkedHashSet: " + linkedset);
27     System.out.println();System.out.println();
28     System.out.println(" Traversing elements using FOR EACH LOOP");
29     for (Character objj : linkedset)
30     { System.out.println(objj); }
31
32     System.out.println(" Traversing elements using ITERATOR");
33     Iterator obj2 = linkedset.iterator();
34     while (obj2.hasNext()) {
35         Character O1 = ( Character ) obj2.next();
36         System.out.println(O1);
37     }
38 }
39 }
```

Console

<terminated> Linked_HashSet [Java Application] D:\JAVA 15\bin\javaw.exe
PROGRAM ON LINKED HASH-SET (GENRIC APPROACH)

Size of LinkedHashSet = 5

Original LinkedHashSet:[V, I, J, A, Y]
Removing Y from LinkedHashSet: true

Trying to Remove Z which is not present: false

Checking if A is present=true

Updated LinkedHashSet: [V, I, J, A]

Traversing elements using FOR EACH LOOP

V
I
J
A

Traversing elements using ITERATOR

V
I
J
A