*Array_List.java — NAME -VIJAY BORATE — ROLL NO IT-2K18-56

```java
//IMPLEMENTING ARRAYLIST  PROGRAM (NON GENERIC APPROACH)
import java.util.ArrayList;
public class Array_List
{
public static void main(String[] args)
{
    ArrayList arr_obj = new ArrayList();//CREATING OBJECT OF ARRAY LIST
    System.out.println("IMPLEMENTING ARRAYLIST  PROGRAM (NON GENERIC APPROACH)");
    System.out.println(arr_obj.isEmpty());//THIS METHOD CHECKS IS IT EMPTY AND RETURNS BOOLEAN VALUE
    System.out.println(arr_obj.size());//RETURNS INTEGER COUNT OF NUMBER OF ELEMENTS IN ARRAYLIST

    arr_obj.add("VIJAY");// ADD() METHOD ADD;s ELEMENT IN THE ARRAY LIST
    arr_obj.add("ROLL NO");
    arr_obj.add(56);
    arr_obj.add(5.6);

    arr_obj.add(0,"HI");
    //ADD ( INDEX,VALUE) --ADD's ELEMENT AT PARTICULAR INDEX AND SHIFT REST ELEMENTS OF ARRAYLIST

    System.out.println(arr_obj.isEmpty());//THIS METHOD CHECKS IS IT EMPTY AND RETURNS BOOLEAN VALUE
    System.out.println(arr_obj);

    arr_obj.set(0, "IIPS");
    //Replaces the element at the specified position in this list with the specified element.
    System.out.println(arr_obj);

    System.out.println(arr_obj.size());
    //RETURNS INT VALUE WHICH TELLS COUNT OF ELEMENT PRESENT IN ARRAYLIST

    System.out.println(arr_obj.contains("AJAY"));
    //RETURNS BOOLEAN VALUE THAT TELLS ELEMENT EXIST IN ARRAY LIST OR NOT ?
    System.out.println(arr_obj.contains("VIJAY"));

    System.out.println(arr_obj.get(3));//RETURNS ELEMENT AT PARTICULAR INDEX

}

}
```

Writable | Smart Insert | 36 : 1 : 1418

Console

<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

```java

//TRAVERSING  ARRAYLIST (NON GENERIC APPROACH) USING 1)FOR LOOP
import java.util.ArrayList;
public class ArrayList2
{
public static void main(String[] args)
{
    ArrayList arr_obj = new ArrayList();//CREATING OBJECT OF ARRAY LIST
    System.out.println("TRAVERSING  ARRAYLIST (NON GENERIC APPROACH) USING FOR LOOP");

    arr_obj.add("HELLOW");// ADD() METHOD ADD;s ELEMENT IN THE ARRAY LIST
    arr_obj.add("JAVA");
    arr_obj.add("PROGRAMMER");
    arr_obj.add(56);

    arr_obj.add(0,"HI");
    //ADD ( INDEX,VALUE) --ADD's ELEMENT AT PARTICULAR INDEX AND SHIFT REST ELEMENTS OF ARRAYLIST

    arr_obj.set(0,"IIPS");
    //Replaces the element at the specified position in this list with the specified element.

    //***********USING FOR LOOP TO TRAVEL IN ARRAYLIST*****************

    for (int i = 0 ; i < arr_obj.size();i++)
    {
        System.out.println(arr_obj.get(i));
    }



}

}

```

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 ADD;s 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)-ADD's 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

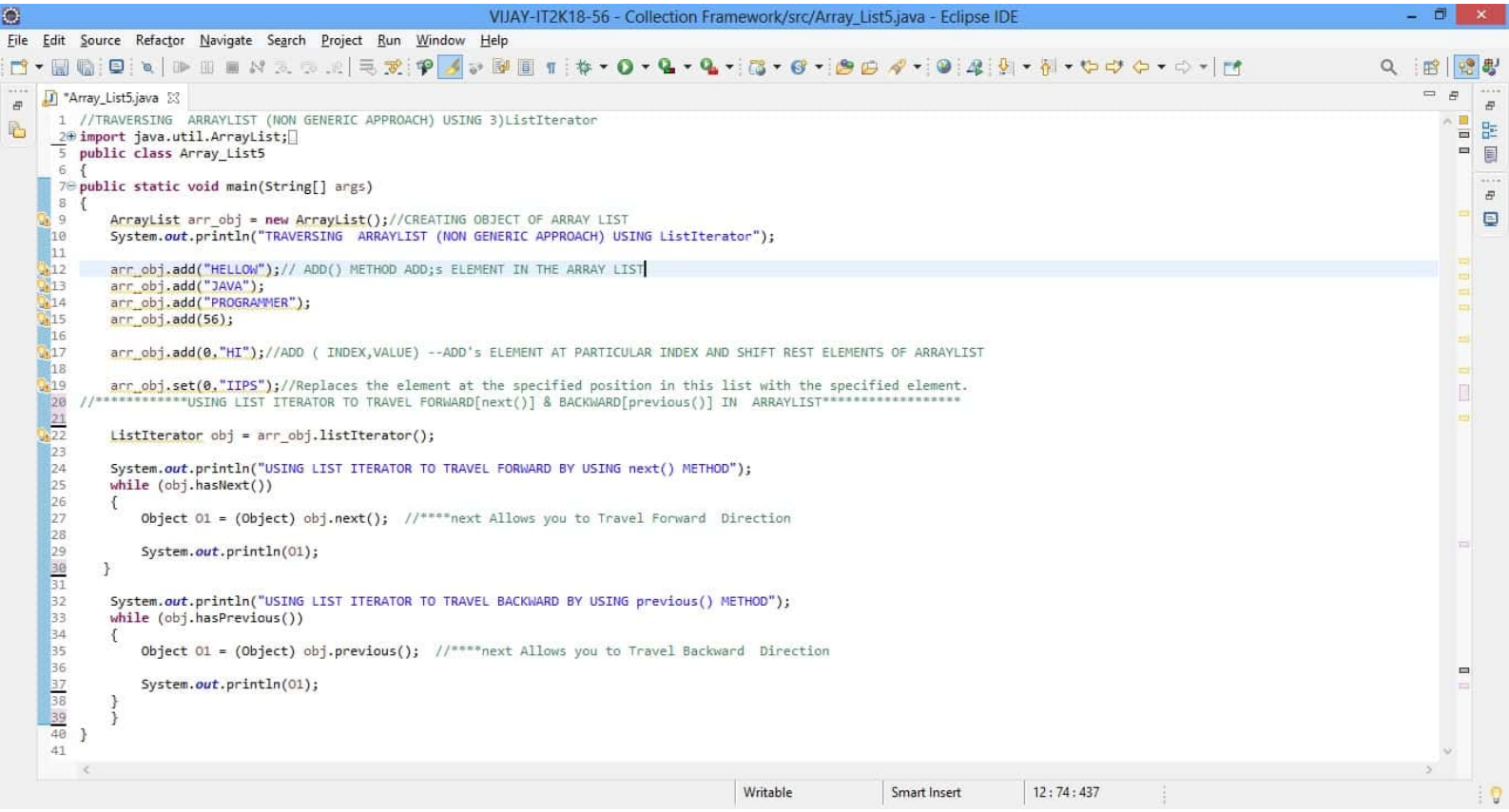
```
<terminated> Array_List3 [Java Application] C:\Program Files\Java\jdk-15\bin\javaw.exe (2
TRAVERSING  ARRAYLIST (NON GENERIC APPROACH) USING FOR EACH LOOP
IIPS
HELLOW
JAVA
PROGRAMMER
56
```

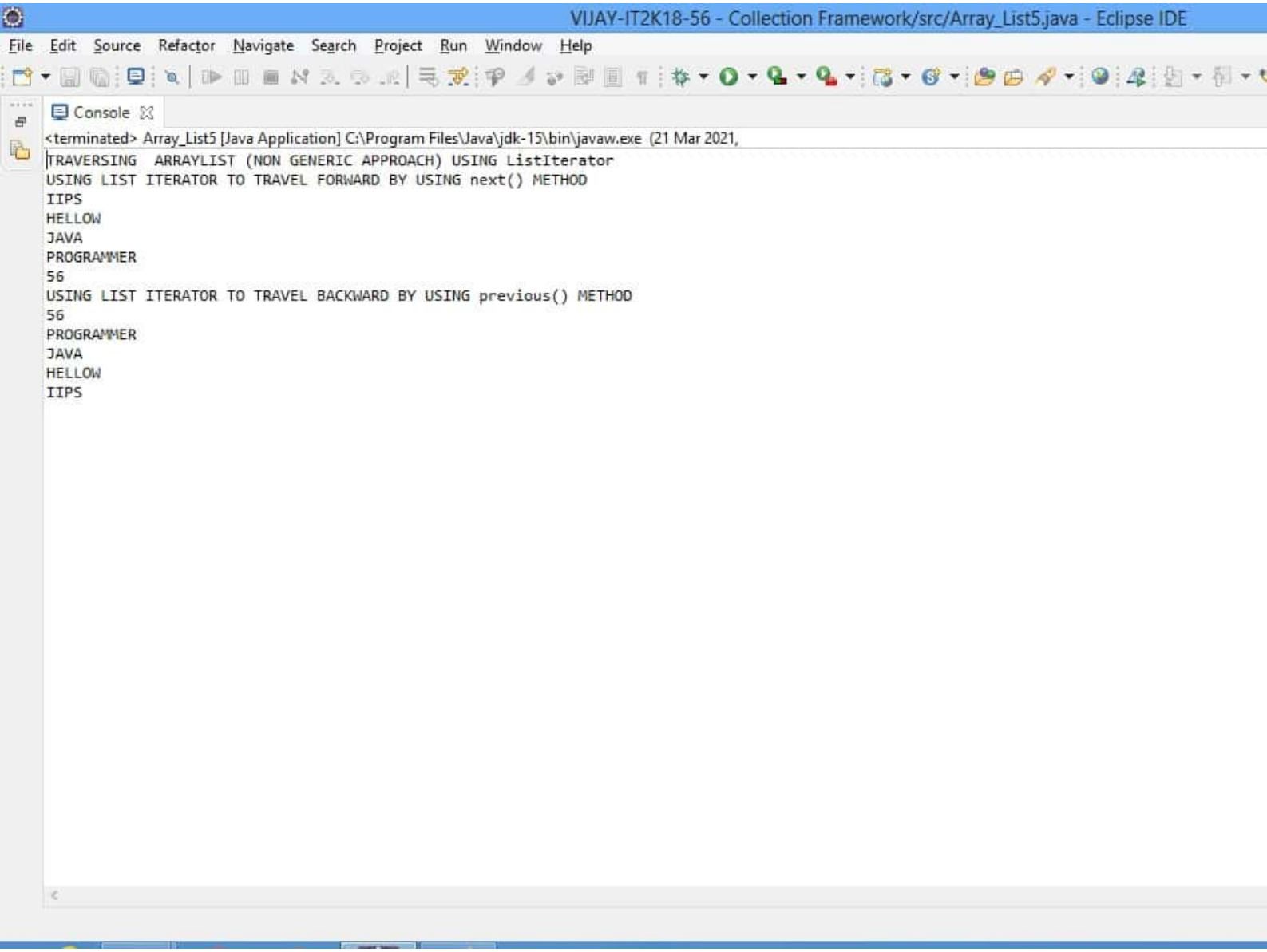
Array_List4.java NAME - VIJAY BORATE ROLL NO - IT2K18-56

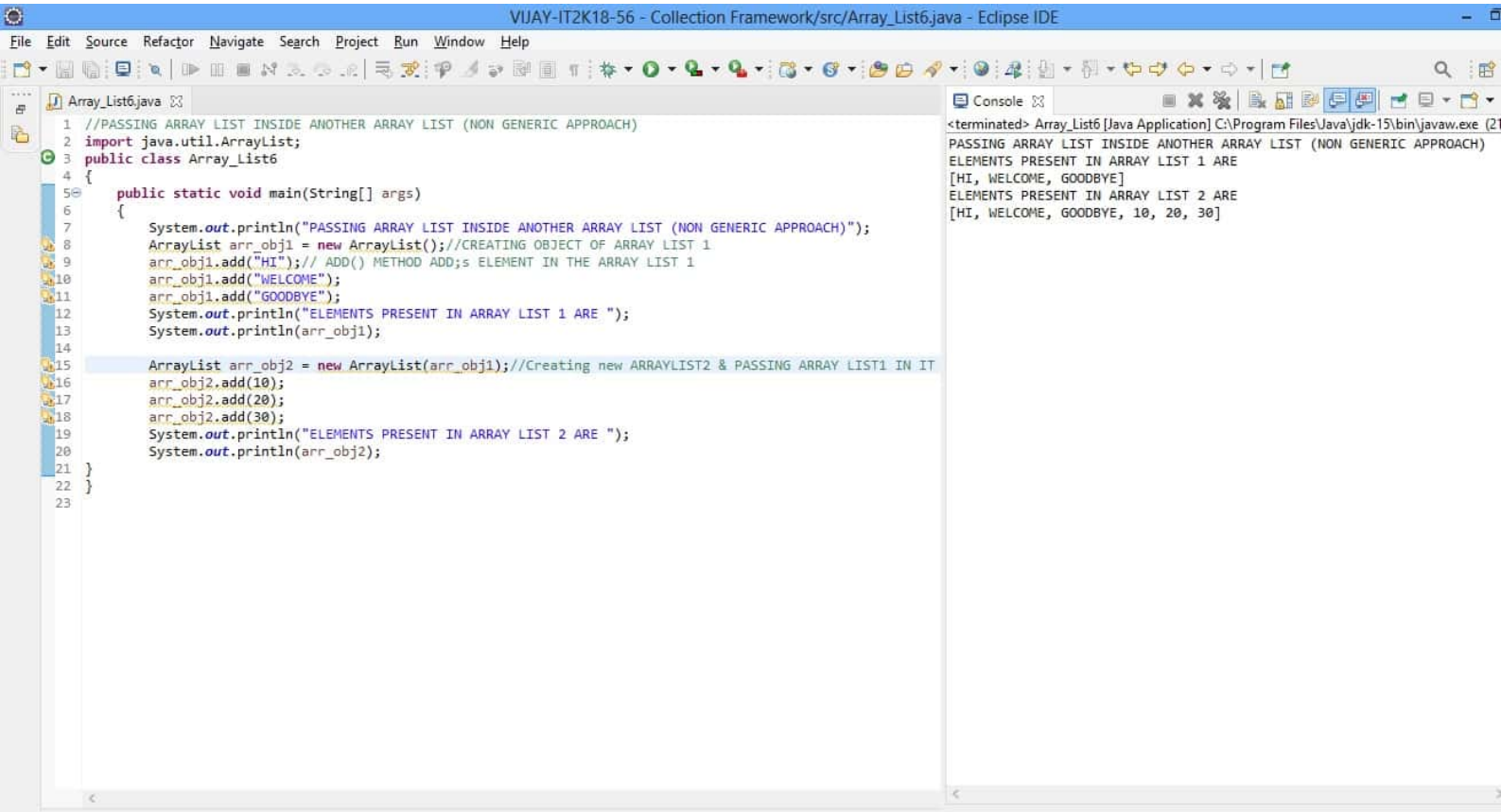
```
1  //TRAVERSING  ARRAYLIST (NON GENERIC APPROACH) USING 3)Iterator
2  import java.util.ArrayList;
4
5
6  public class Array_List4
7  {
8  public static void main(String[] args)
9  {
10     ArrayList arr_obj = new ArrayList();//CREATING OBJECT OF ARRAY LIST
11     System.out.println("TRAVERSING ARRAYLIST   (NON GENERIC APPROACH) USING 3)Iterator");
12     arr_obj.add("HELLOW");// ADD() METHOD ADD;s 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) --ADD's 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
```

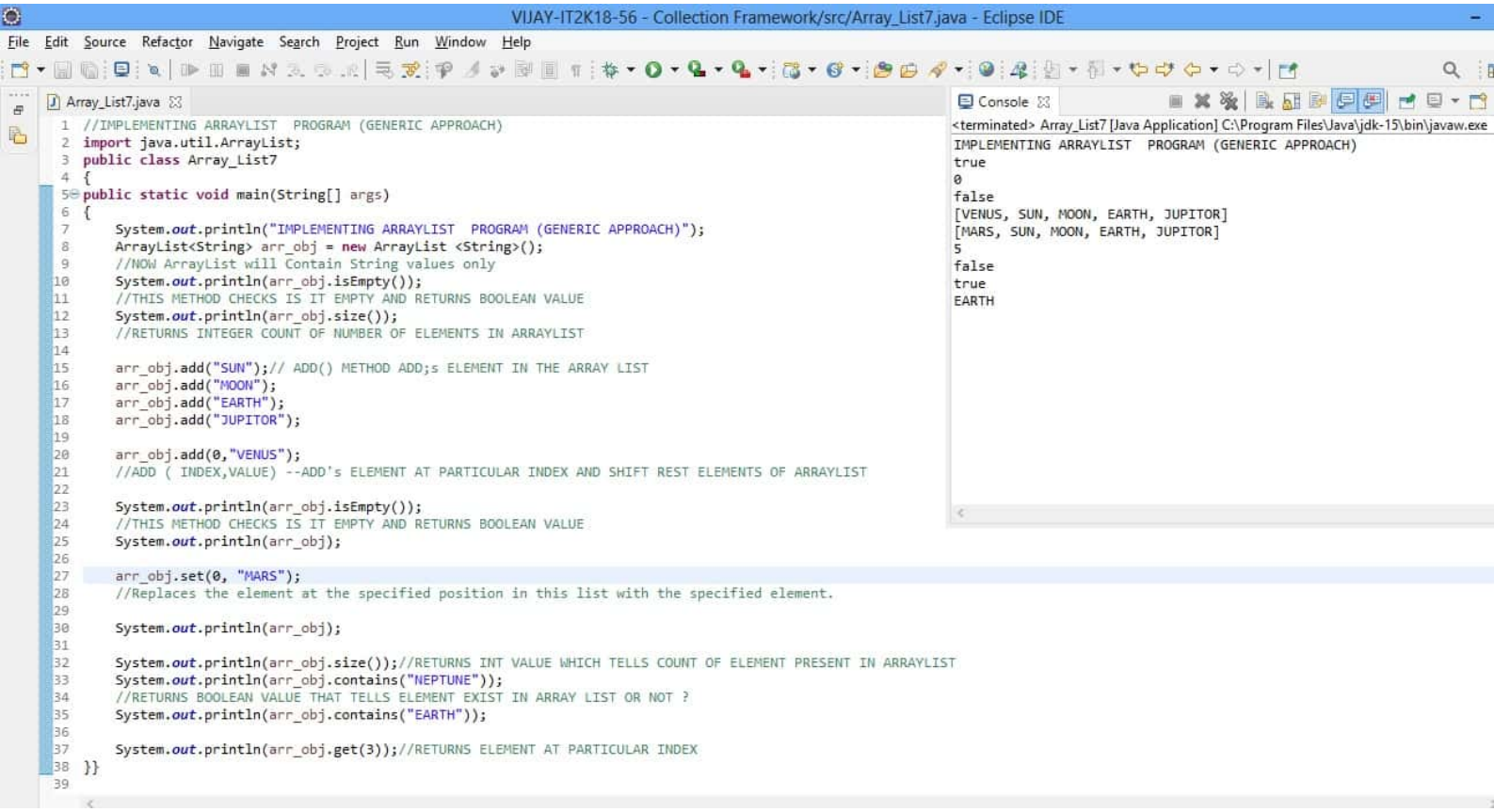
Console

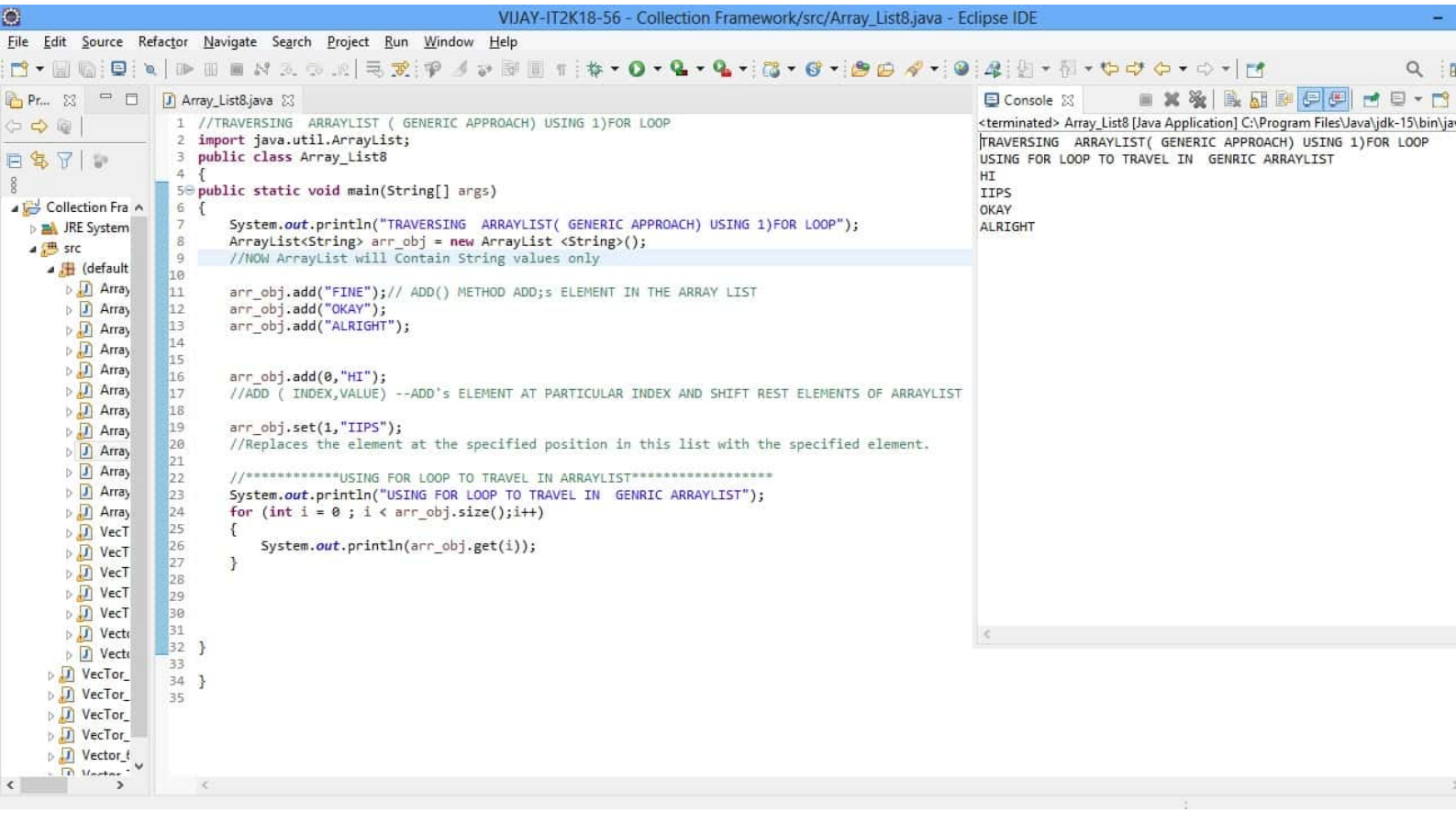
```
<terminated> Array_List4 [Java Application] C:\Program Files\Java\jdk-15\bin\javaw.e
TRAVERSING ARRAYLIST   (NON GENERIC APPROACH) USING 3)Iterator
IIPS
HELLOW
JAVA
PROGRAMMER
56
```

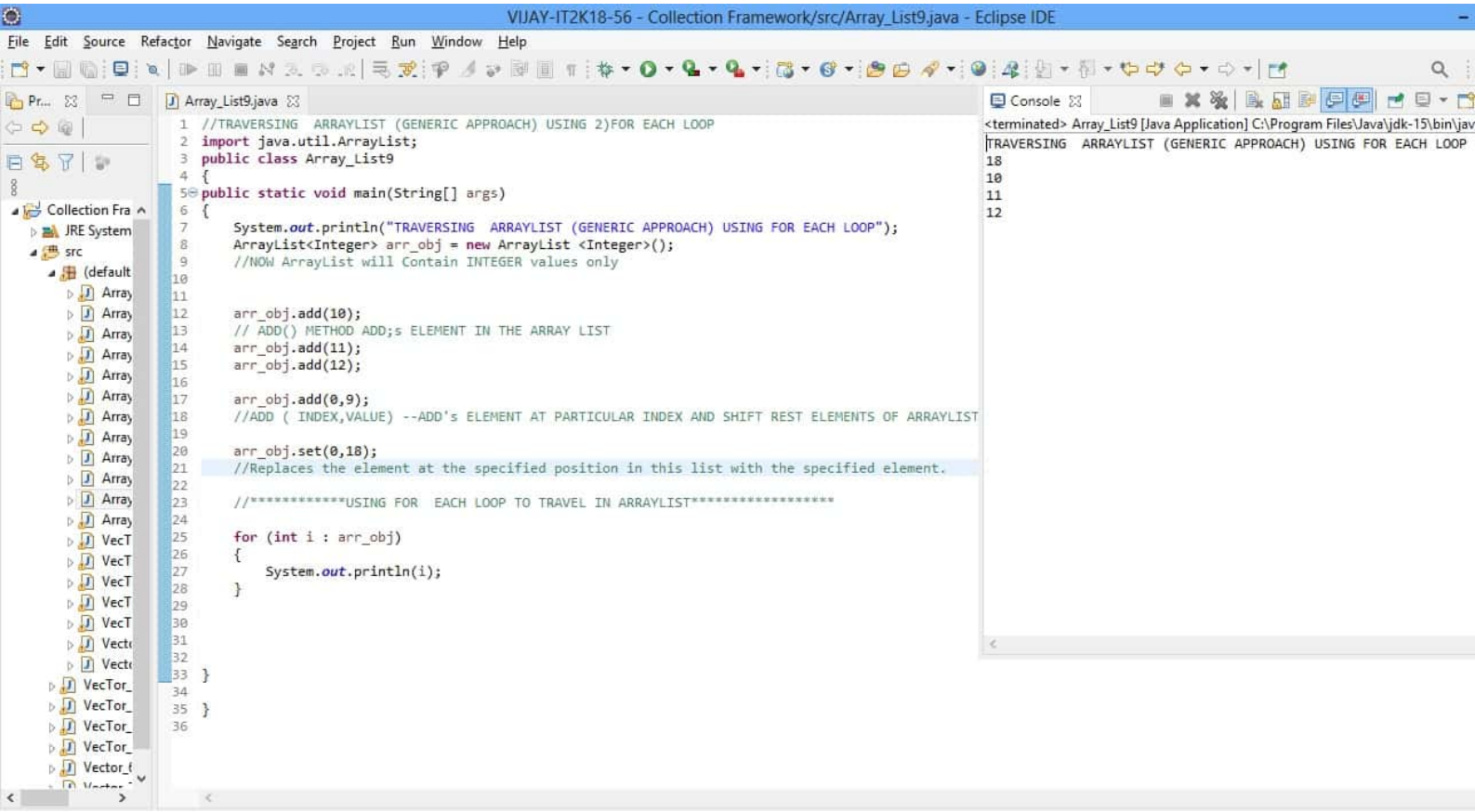


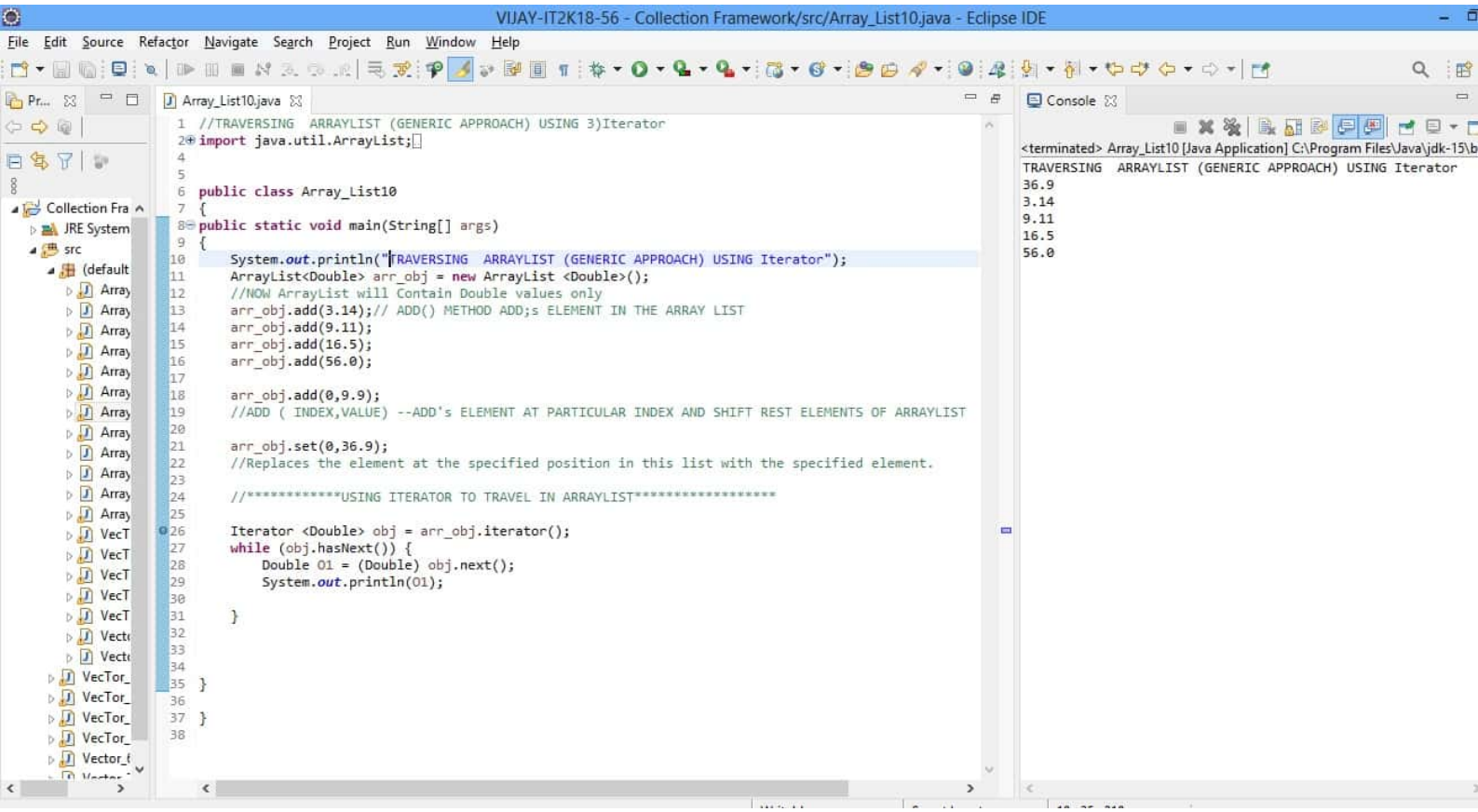


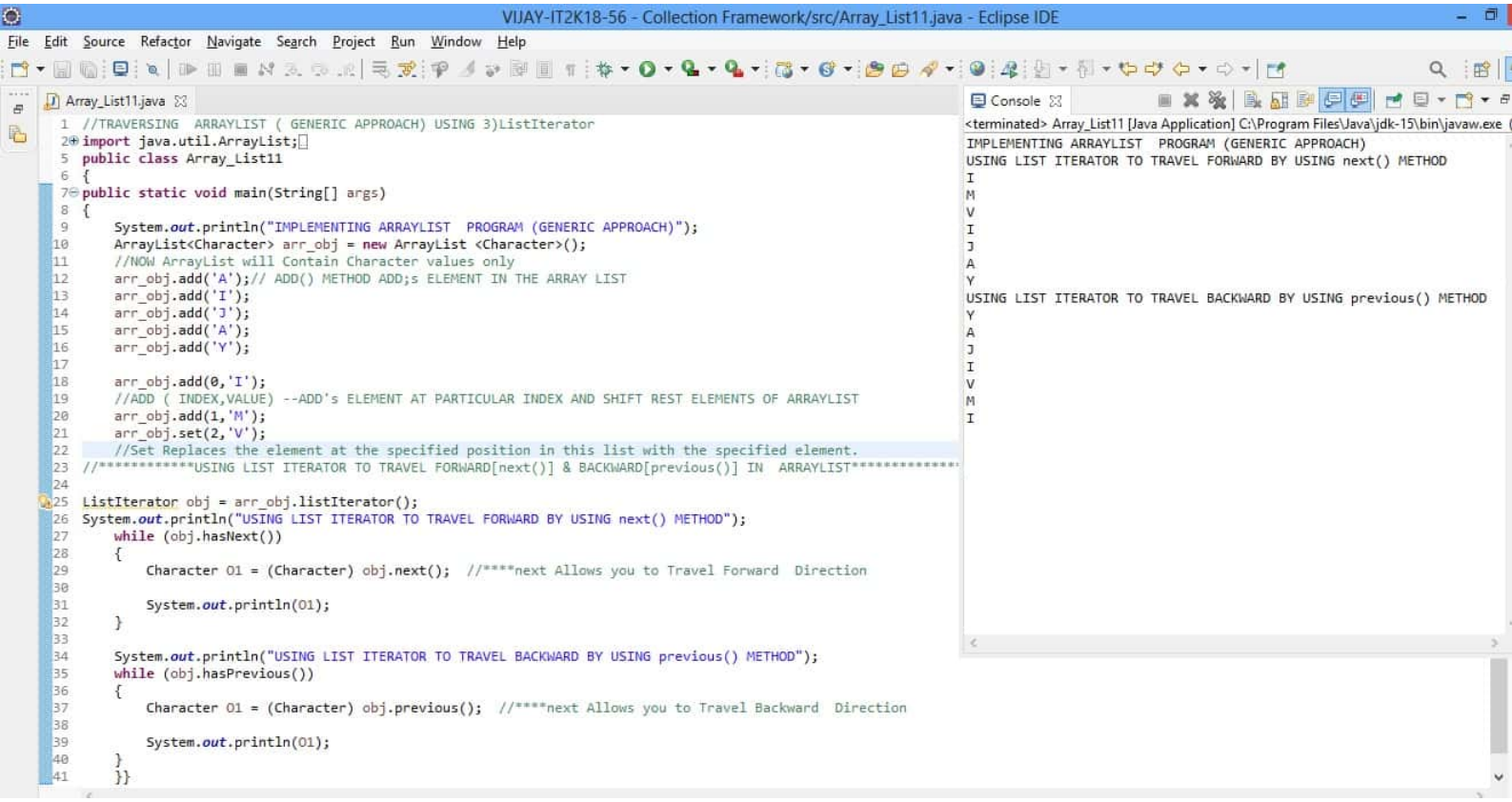


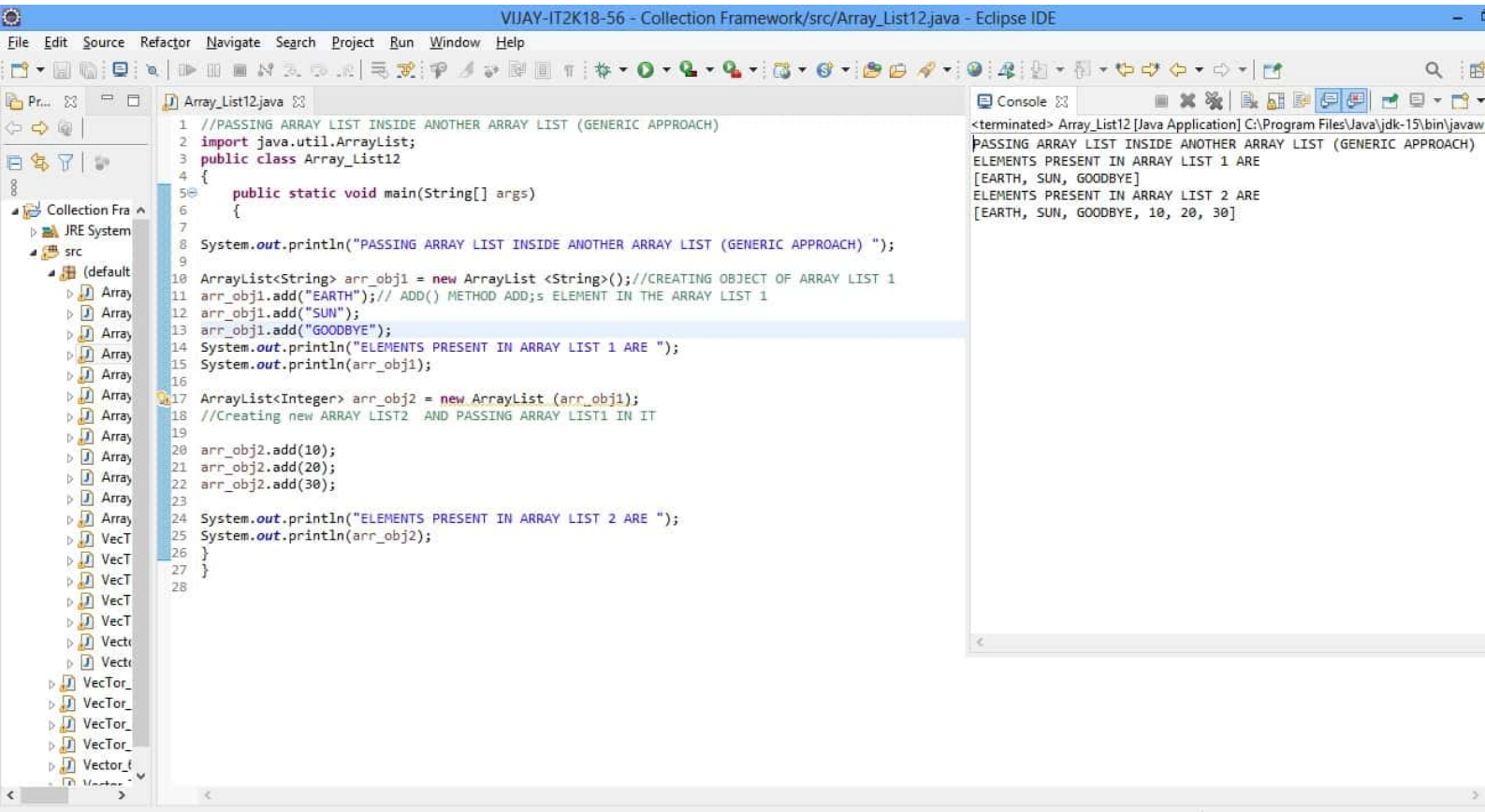


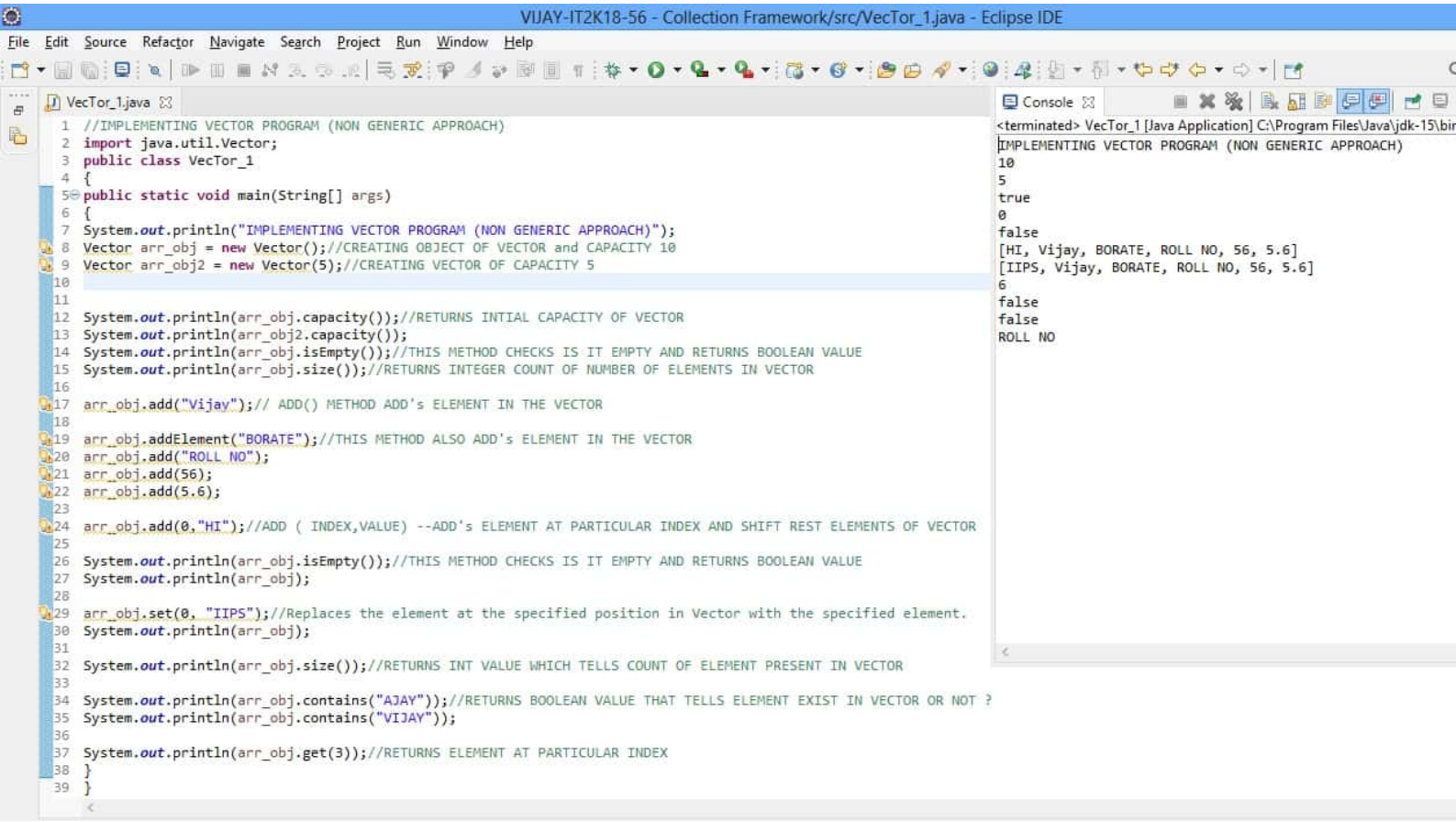




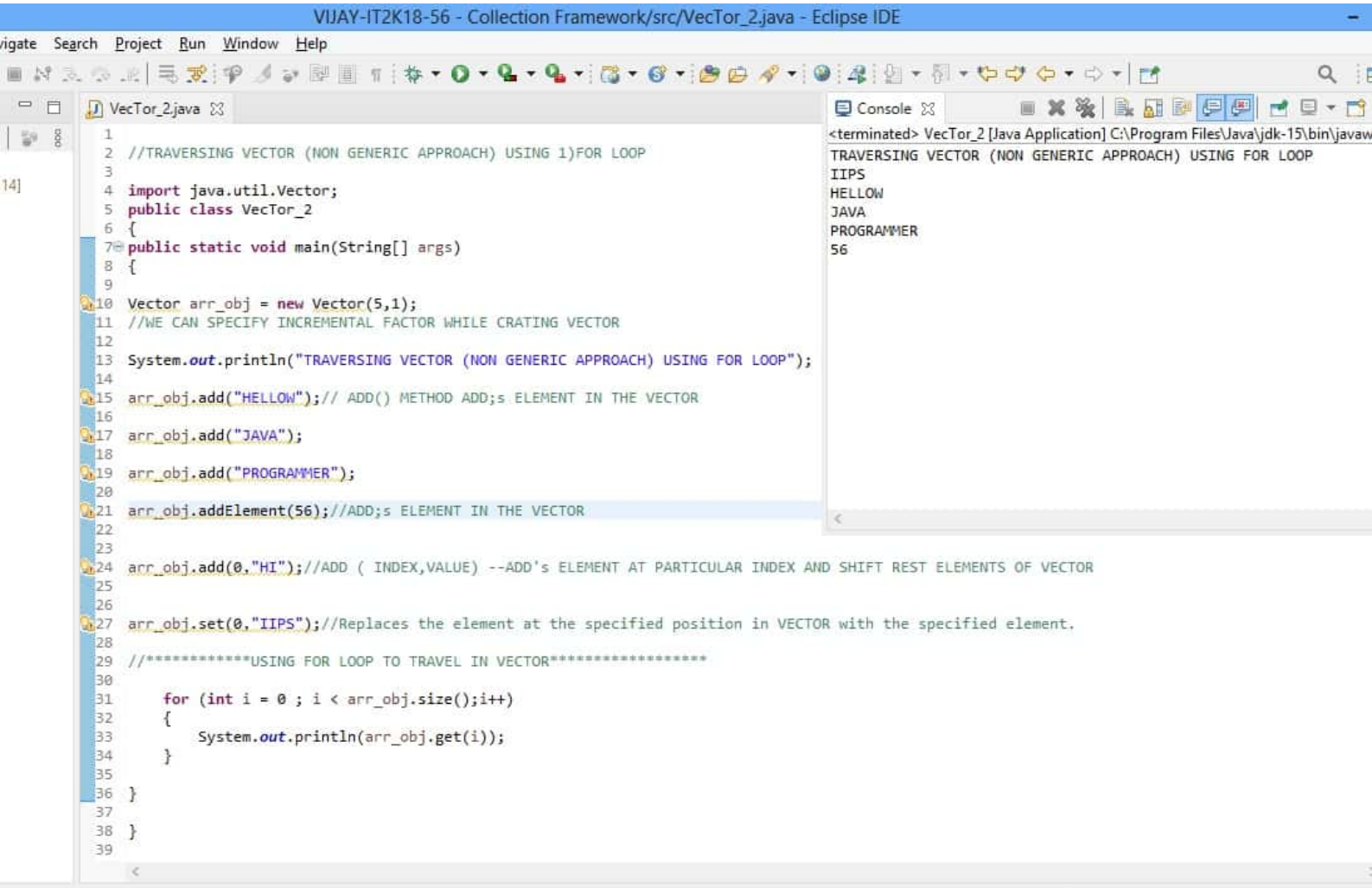




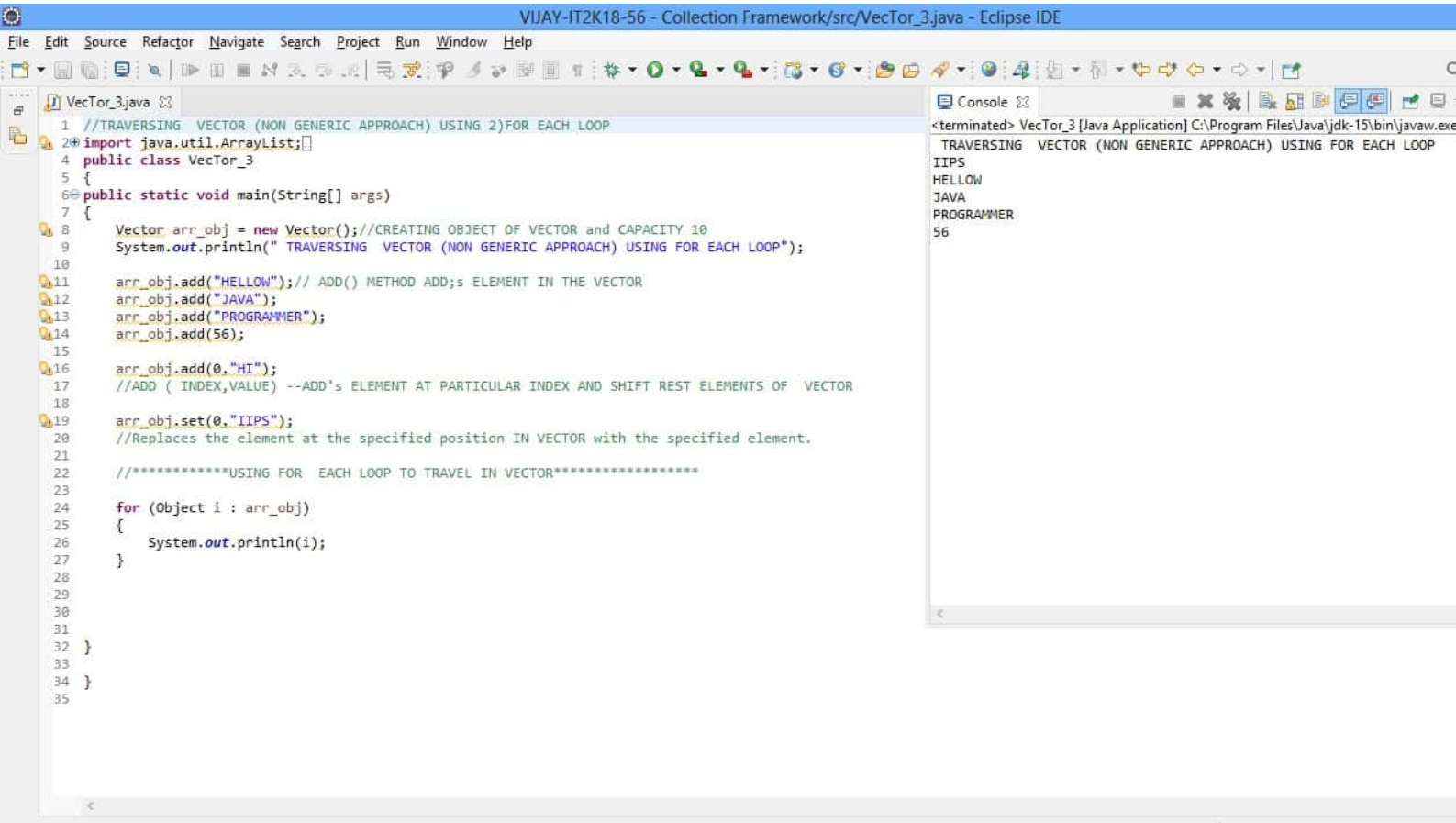


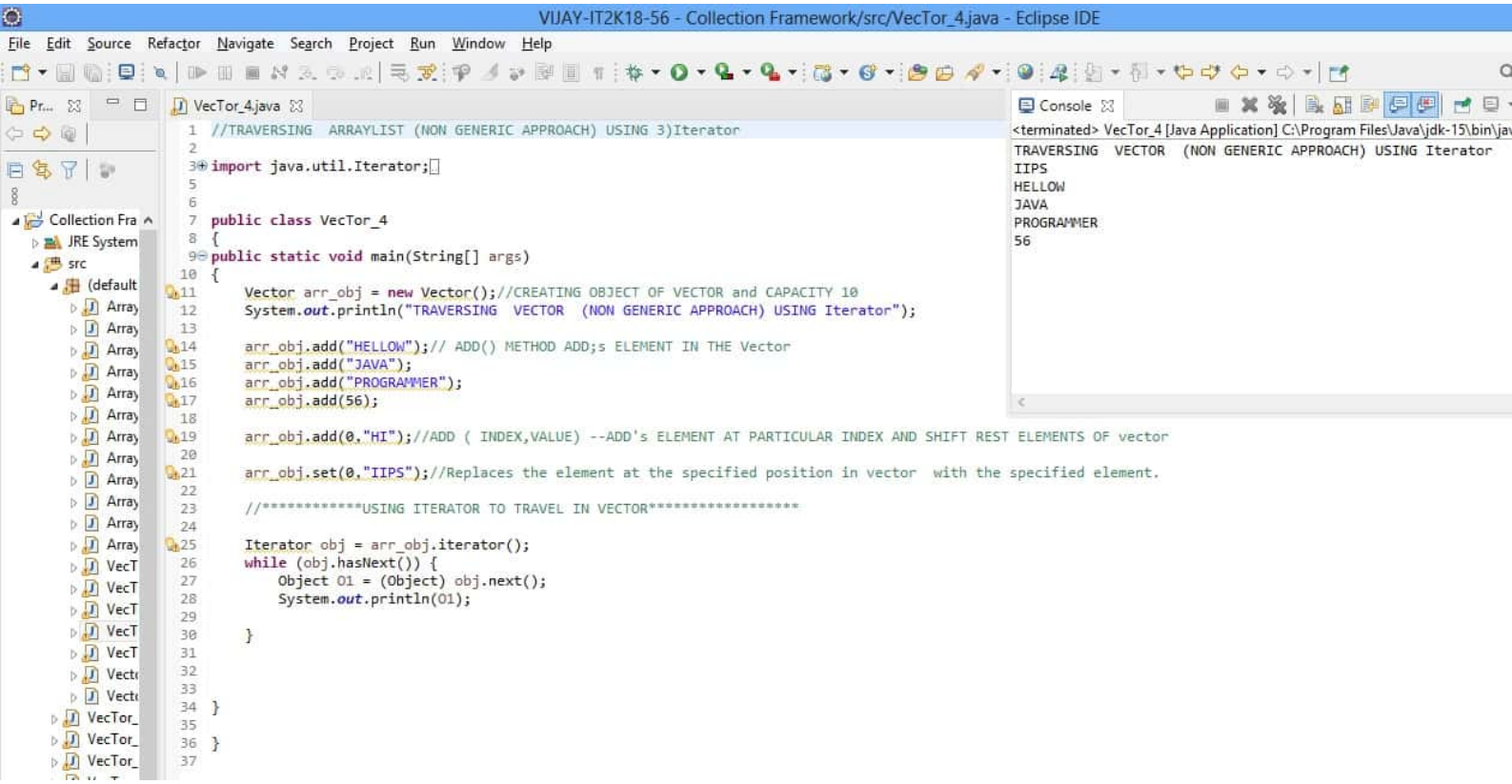



```
1  //IMPLEMENTING VECTOR PROGRAM (NON GENERIC APPROACH)
2  import java.util.Vector;
3  public class VecTor_1
4  {
5  public static void main(String[] args)
6  {
7  System.out.println("IMPLEMENTING VECTOR PROGRAM (NON GENERIC APPROACH)");
8  Vector arr_obj = new Vector();//CREATING OBJECT OF VECTOR 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 INTIAL 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 }
```

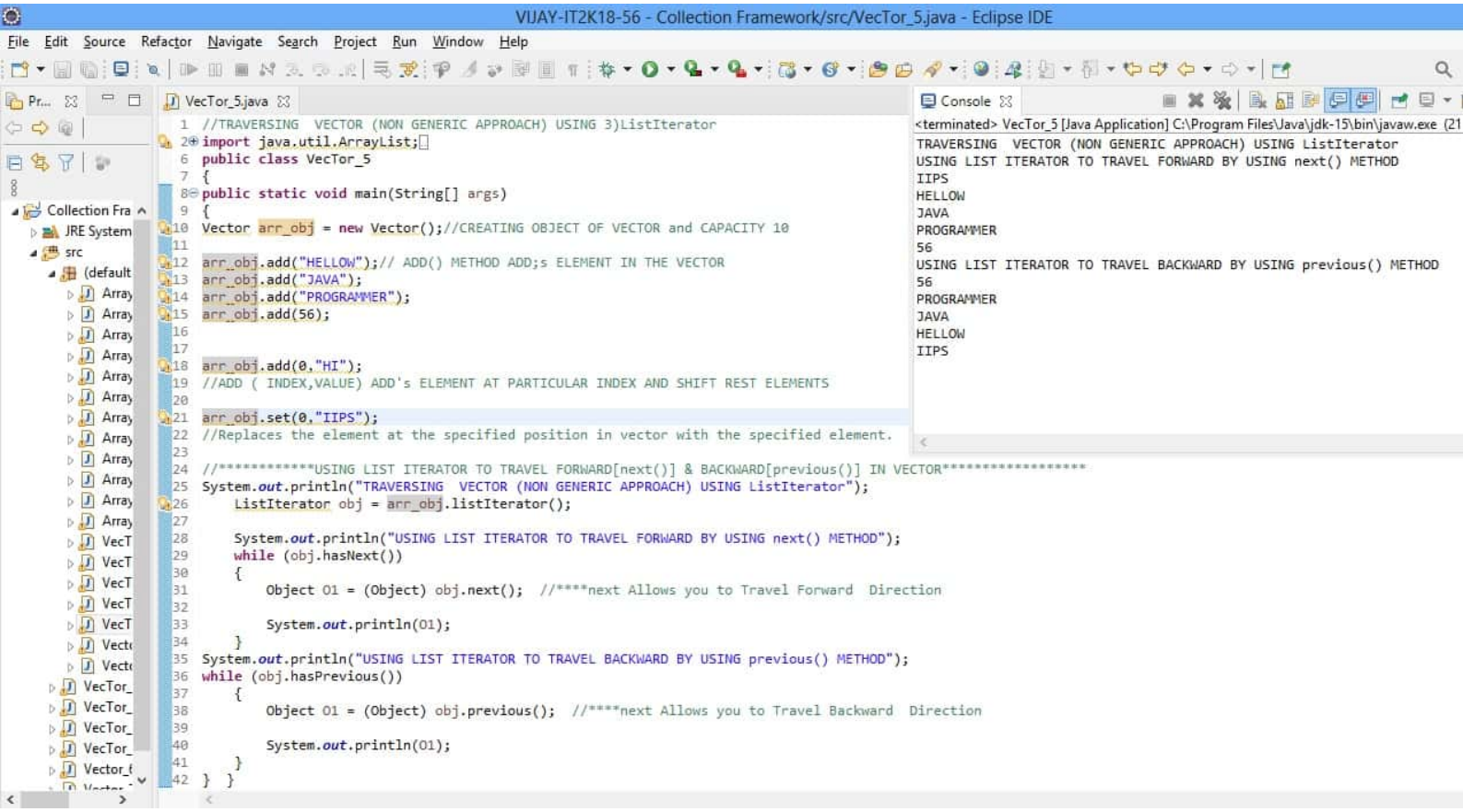



```java
//TRAVERSING VECTOR (NON GENERIC APPROACH) USING 1)FOR LOOP

import java.util.Vector;
public class VecTor_2
{
public static void main(String[] args)
{

Vector arr_obj = new Vector(5,1);
//WE CAN SPECIFY INCREMENTAL FACTOR WHILE CRATING VECTOR

System.out.println("TRAVERSING VECTOR (NON GENERIC APPROACH) USING FOR LOOP");

arr_obj.add("HELLOW");// ADD() METHOD ADD;s ELEMENT IN THE VECTOR

arr_obj.add("JAVA");

arr_obj.add("PROGRAMMER");

arr_obj.addElement(56);//ADD;s ELEMENT IN THE VECTOR


arr_obj.add(0,"HI");//ADD ( INDEX,VALUE) --ADD's ELEMENT AT PARTICULAR INDEX AND SHIFT REST ELEMENTS OF VECTOR


arr_obj.set(0,"IIPS");//Replaces the element at the specified position in VECTOR with the specified element.

//***********USING FOR LOOP TO TRAVEL IN VECTOR*****************

    for (int i = 0 ; i < arr_obj.size();i++)
    {
        System.out.println(arr_obj.get(i));
    }

}

}
```

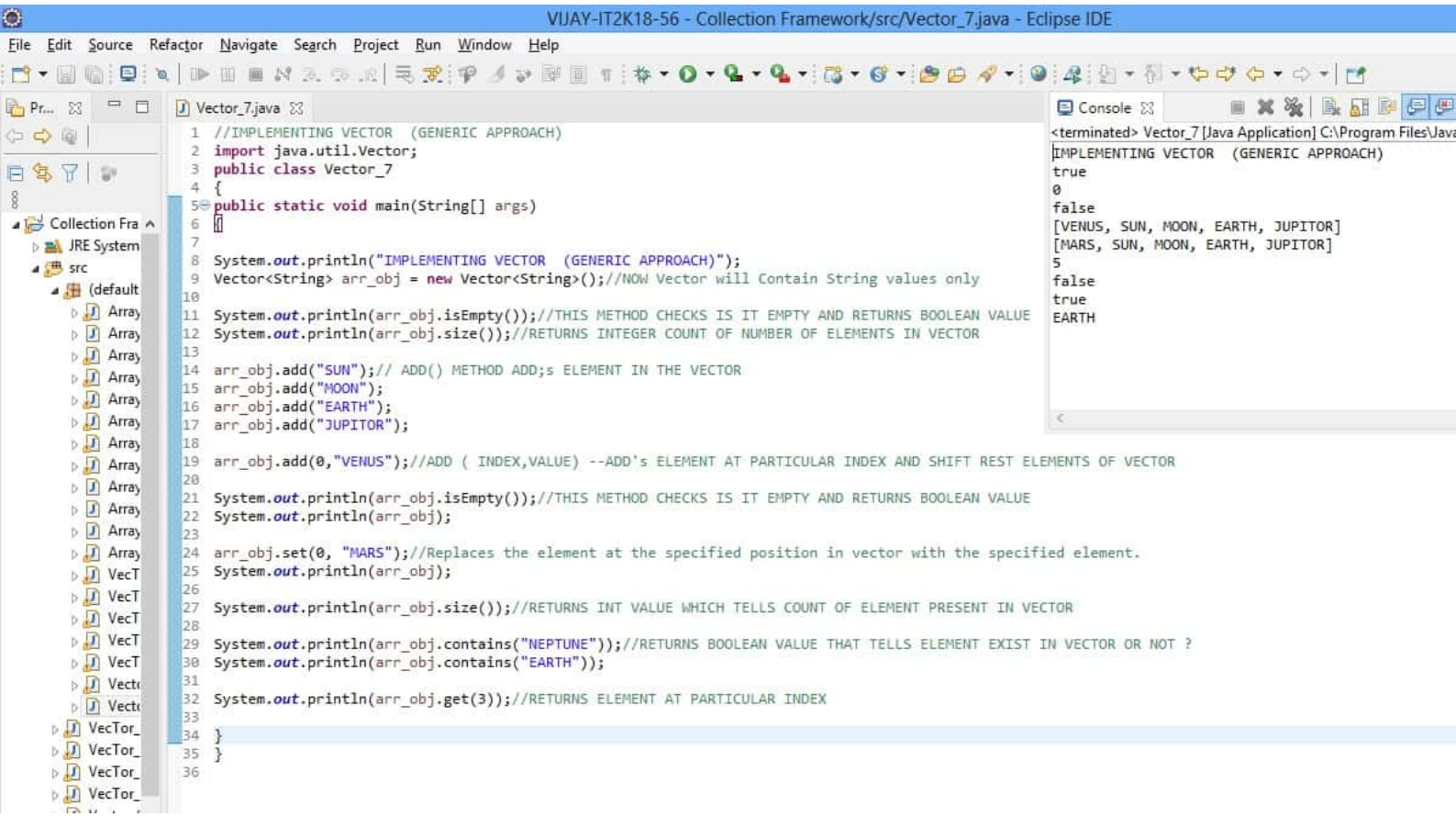




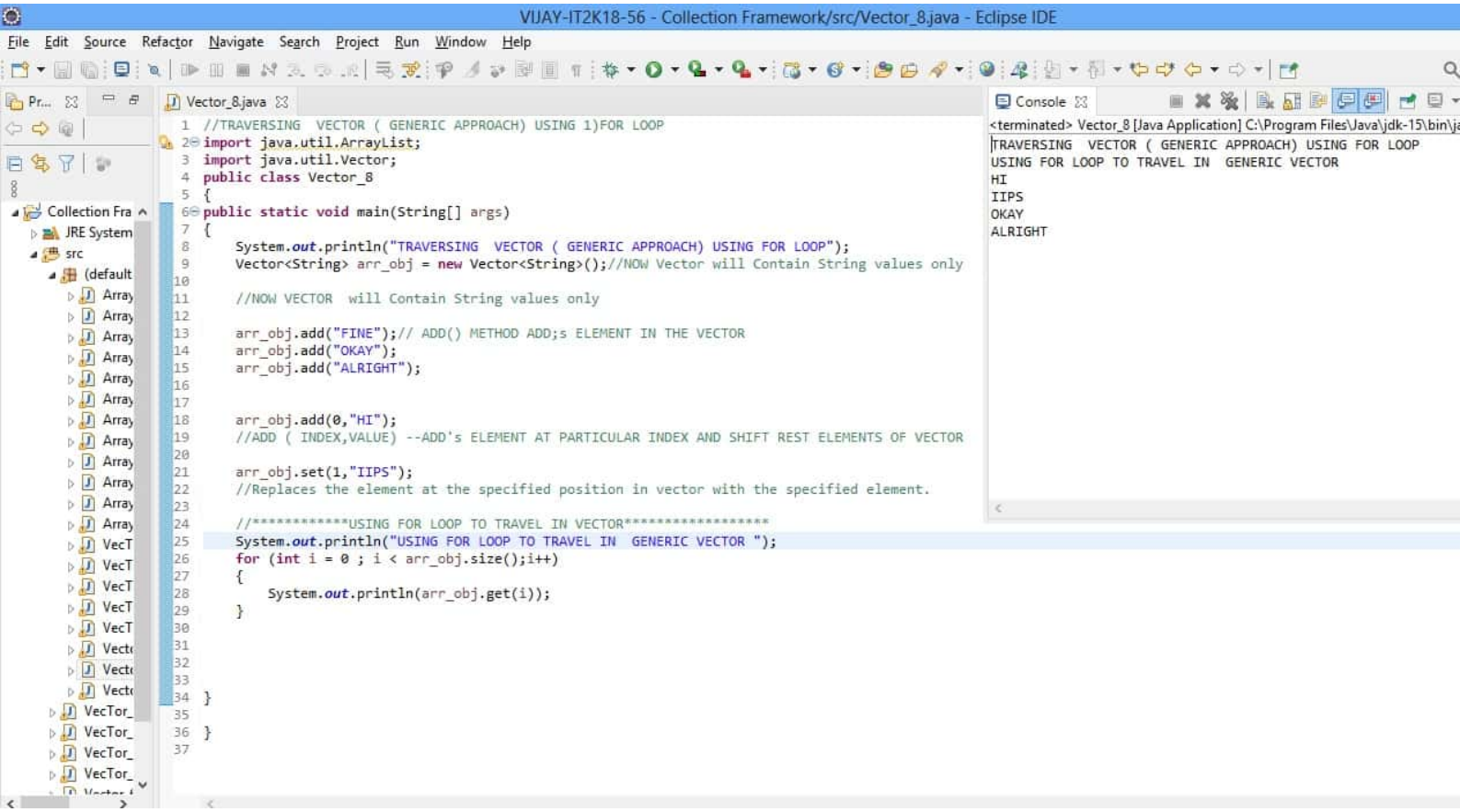

```java
//TRAVERSING  ARRAYLIST (NON GENERIC APPROACH) USING 3)Iterator

import java.util.Iterator;


public class VecTor_4
{
public static void main(String[] args)
{
    Vector arr_obj = new Vector();//CREATING OBJECT OF VECTOR and CAPACITY 10
    System.out.println("TRAVERSING  VECTOR  (NON GENERIC APPROACH) USING Iterator");

    arr_obj.add("HELLOW");// ADD() METHOD ADD;s ELEMENT IN THE Vector
    arr_obj.add("JAVA");
    arr_obj.add("PROGRAMMER");
    arr_obj.add(56);

    arr_obj.add(0,"HI");//ADD ( INDEX,VALUE) --ADD's ELEMENT AT PARTICULAR INDEX AND SHIFT REST ELEMENTS OF vector

    arr_obj.set(0,"IIPS");//Replaces the element at the specified position in vector  with the specified element.

    //***********USING ITERATOR TO TRAVEL IN VECTOR*****************

    Iterator obj = arr_obj.iterator();
    while (obj.hasNext()) {
        Object O1 = (Object) obj.next();
        System.out.println(O1);

    }


}

}
```



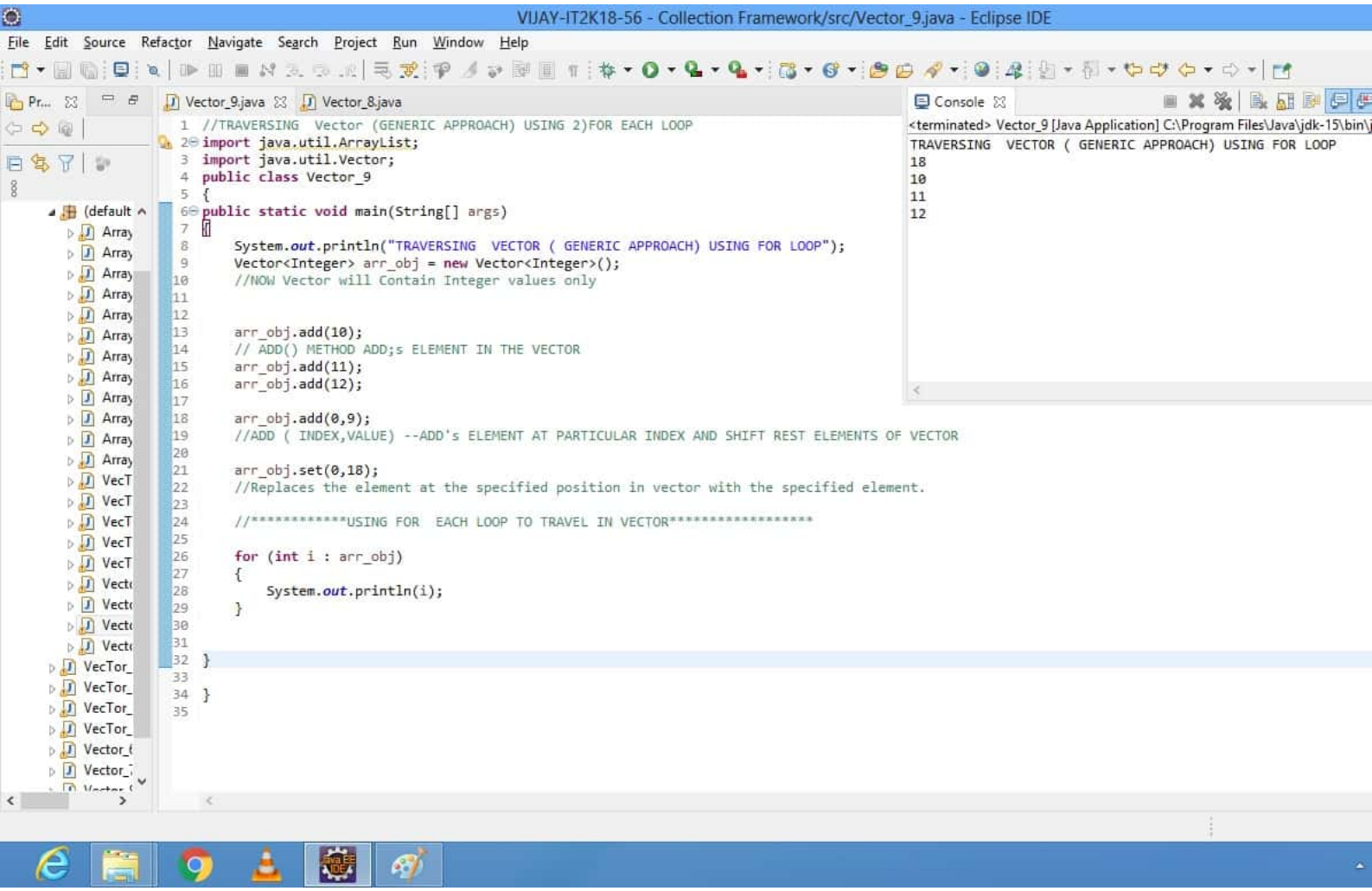

```
1  //TRAVERSING  VECTOR (NON GENERIC APPROACH) USING 3)ListIterator
2  import java.util.ArrayList;
6  public class VecTor_5
7  {
8  public static void main(String[] args)
9  {
10 Vector arr_obj = new Vector();//CREATING OBJECT OF VECTOR and CAPACITY 10
11
12 arr_obj.add("HELLOW");// ADD() METHOD ADD;s ELEMENT IN THE VECTOR
13 arr_obj.add("JAVA");
14 arr_obj.add("PROGRAMMER");
15 arr_obj.add(56);
16
17
18 arr_obj.add(0,"HI");
19 //ADD ( INDEX,VALUE) ADD's ELEMENT AT PARTICULAR INDEX AND SHIFT REST ELEMENTS
20
21 arr_obj.set(0,"IIPS");
22 //Replaces the element at the specified position in vector with the specified element.
23
24 //***********USING LIST ITERATOR TO TRAVEL FORWARD[next()] & BACKWARD[previous()] IN VECTOR*****************
25 System.out.println("TRAVERSING  VECTOR (NON GENERIC APPROACH) USING ListIterator");
26     ListIterator obj = arr_obj.listIterator();
27
28     System.out.println("USING LIST ITERATOR TO TRAVEL FORWARD BY USING next() METHOD");
29     while (obj.hasNext())
30     {
31         Object O1 = (Object) obj.next();  //****next Allows you to Travel Forward  Direction
32
33         System.out.println(O1);
34     }
35 System.out.println("USING LIST ITERATOR TO TRAVEL BACKWARD BY USING previous() METHOD");
36 while (obj.hasPrevious())
37     {
38         Object O1 = (Object) obj.previous();  //****next Allows you to Travel Backward  Direction
39
40         System.out.println(O1);
41     }
42 } }
```

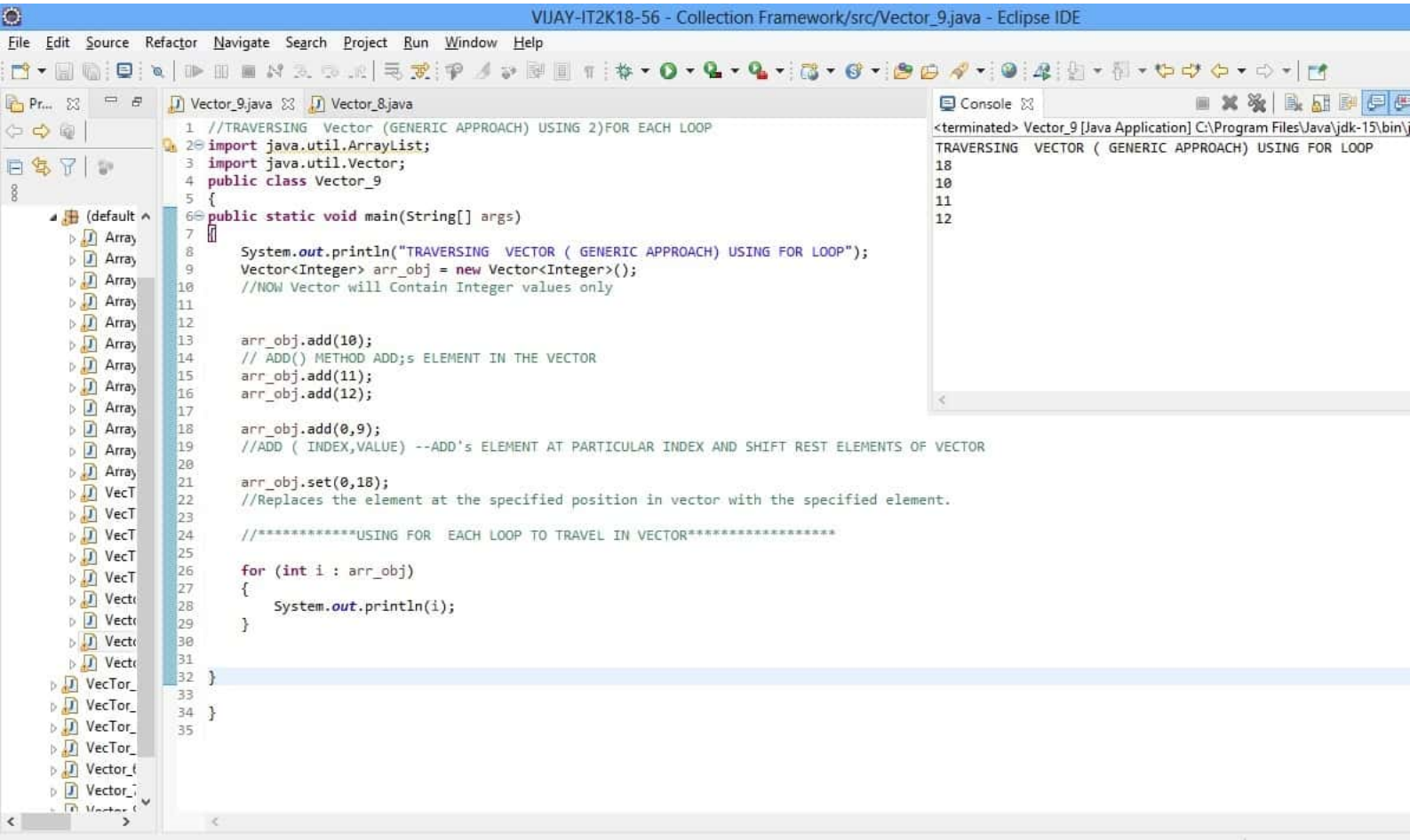


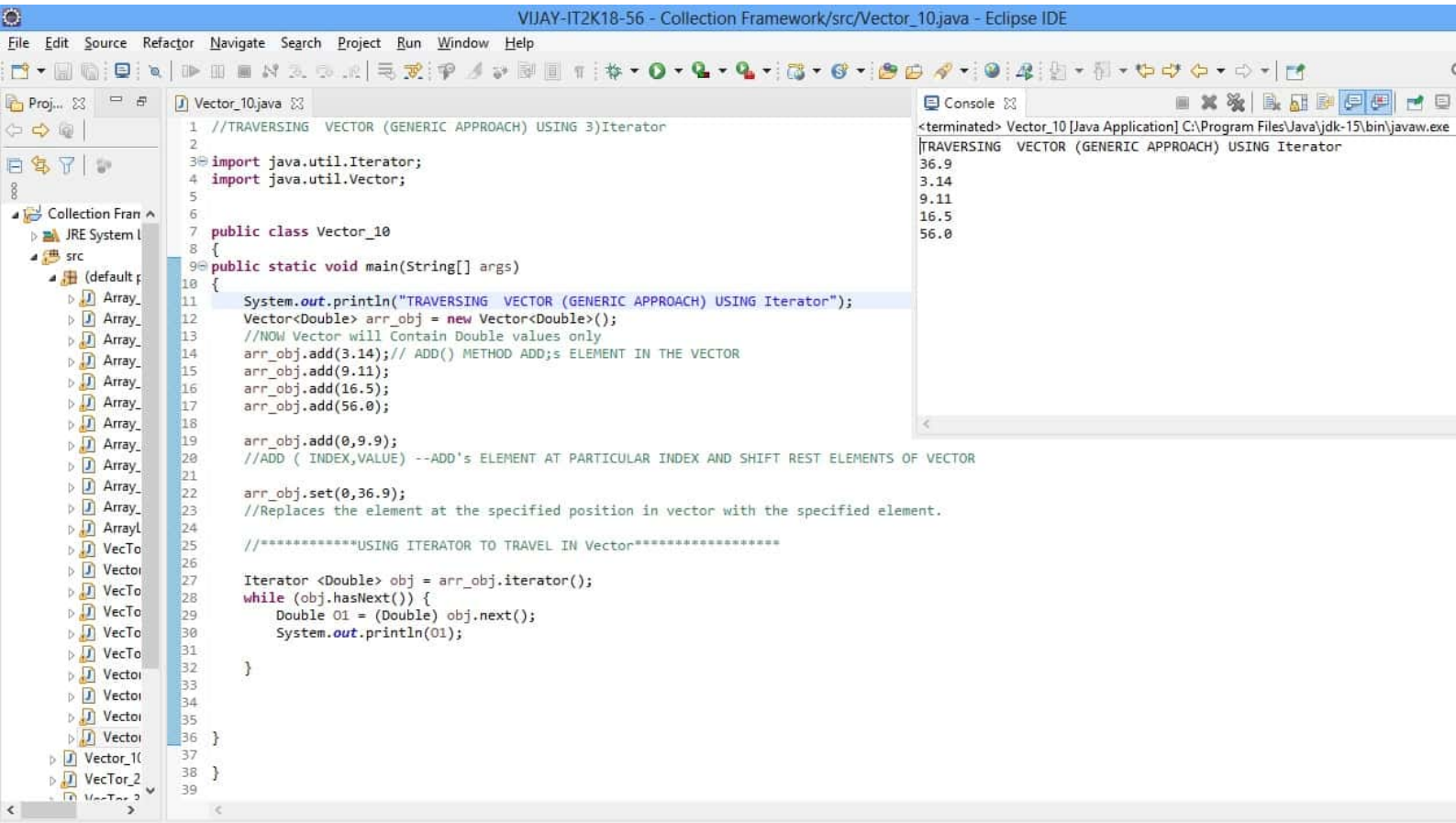

```java
//IMPLEMENTING VECTOR  (GENERIC APPROACH)
import java.util.Vector;
public class Vector_7
{
public static void main(String[] args)
{

System.out.println("IMPLEMENTING VECTOR  (GENERIC APPROACH)");
Vector<String> arr_obj = new Vector<String>();//NOW Vector will Contain String values only

System.out.println(arr_obj.isEmpty());//THIS METHOD CHECKS IS IT EMPTY AND RETURNS BOOLEAN VALUE
System.out.println(arr_obj.size());//RETURNS INTEGER COUNT OF NUMBER OF ELEMENTS IN VECTOR

arr_obj.add("SUN");// ADD() METHOD ADD;s ELEMENT IN THE VECTOR
arr_obj.add("MOON");
arr_obj.add("EARTH");
arr_obj.add("JUPITOR");

arr_obj.add(0,"VENUS");//ADD ( INDEX,VALUE) --ADD's ELEMENT AT PARTICULAR INDEX AND SHIFT REST ELEMENTS OF VECTOR

System.out.println(arr_obj.isEmpty());//THIS METHOD CHECKS IS IT EMPTY AND RETURNS BOOLEAN VALUE
System.out.println(arr_obj);

arr_obj.set(0, "MARS");//Replaces the element at the specified position in vector with the specified element.
System.out.println(arr_obj);

System.out.println(arr_obj.size());//RETURNS INT VALUE WHICH TELLS COUNT OF ELEMENT PRESENT IN VECTOR

System.out.println(arr_obj.contains("NEPTUNE"));//RETURNS BOOLEAN VALUE THAT TELLS ELEMENT EXIST IN VECTOR OR NOT ?
System.out.println(arr_obj.contains("EARTH"));

System.out.println(arr_obj.get(3));//RETURNS ELEMENT AT PARTICULAR INDEX

}
}
```



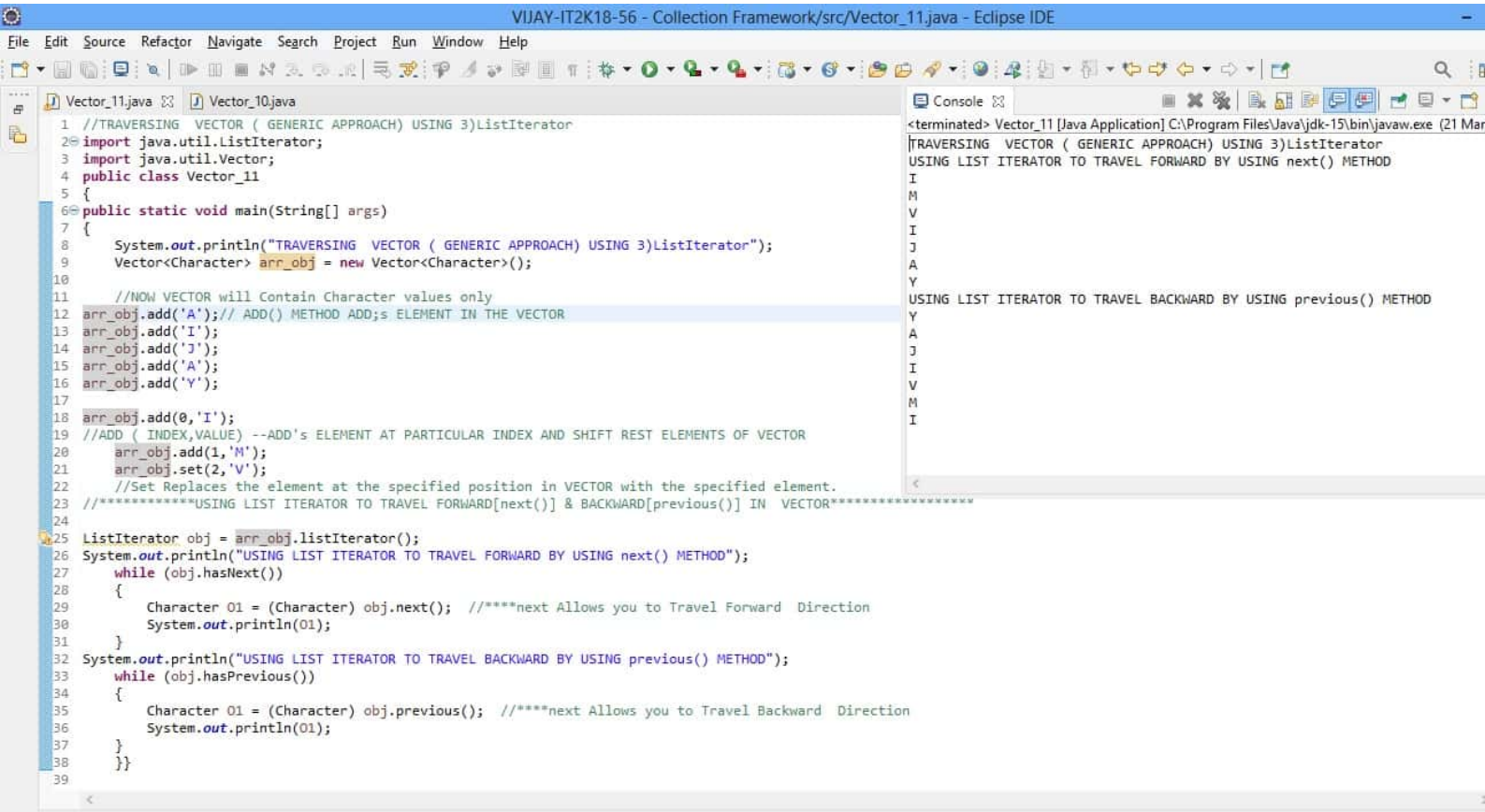

```java
//TRAVERSING  VECTOR ( GENERIC APPROACH) USING 1)FOR LOOP
import java.util.ArrayList;
import java.util.Vector;
public class Vector_8
{
public static void main(String[] args)
{
    System.out.println("TRAVERSING  VECTOR ( GENERIC APPROACH) USING FOR LOOP");
    Vector<String> arr_obj = new Vector<String>();//NOW Vector will Contain String values only

    //NOW VECTOR  will Contain String values only

    arr_obj.add("FINE");// ADD() METHOD ADD;s ELEMENT IN THE VECTOR
    arr_obj.add("OKAY");
    arr_obj.add("ALRIGHT");


    arr_obj.add(0,"HI");
    //ADD ( INDEX,VALUE) --ADD's ELEMENT AT PARTICULAR INDEX AND SHIFT REST ELEMENTS OF VECTOR

    arr_obj.set(1,"IIPS");
    //Replaces the element at the specified position in vector with the specified element.

    //***********USING FOR LOOP TO TRAVEL IN VECTOR*****************
    System.out.println("USING FOR LOOP TO TRAVEL IN  GENERIC VECTOR ");
    for (int i = 0 ; i < arr_obj.size();i++)
    {
        System.out.println(arr_obj.get(i));
    }



}
}
```

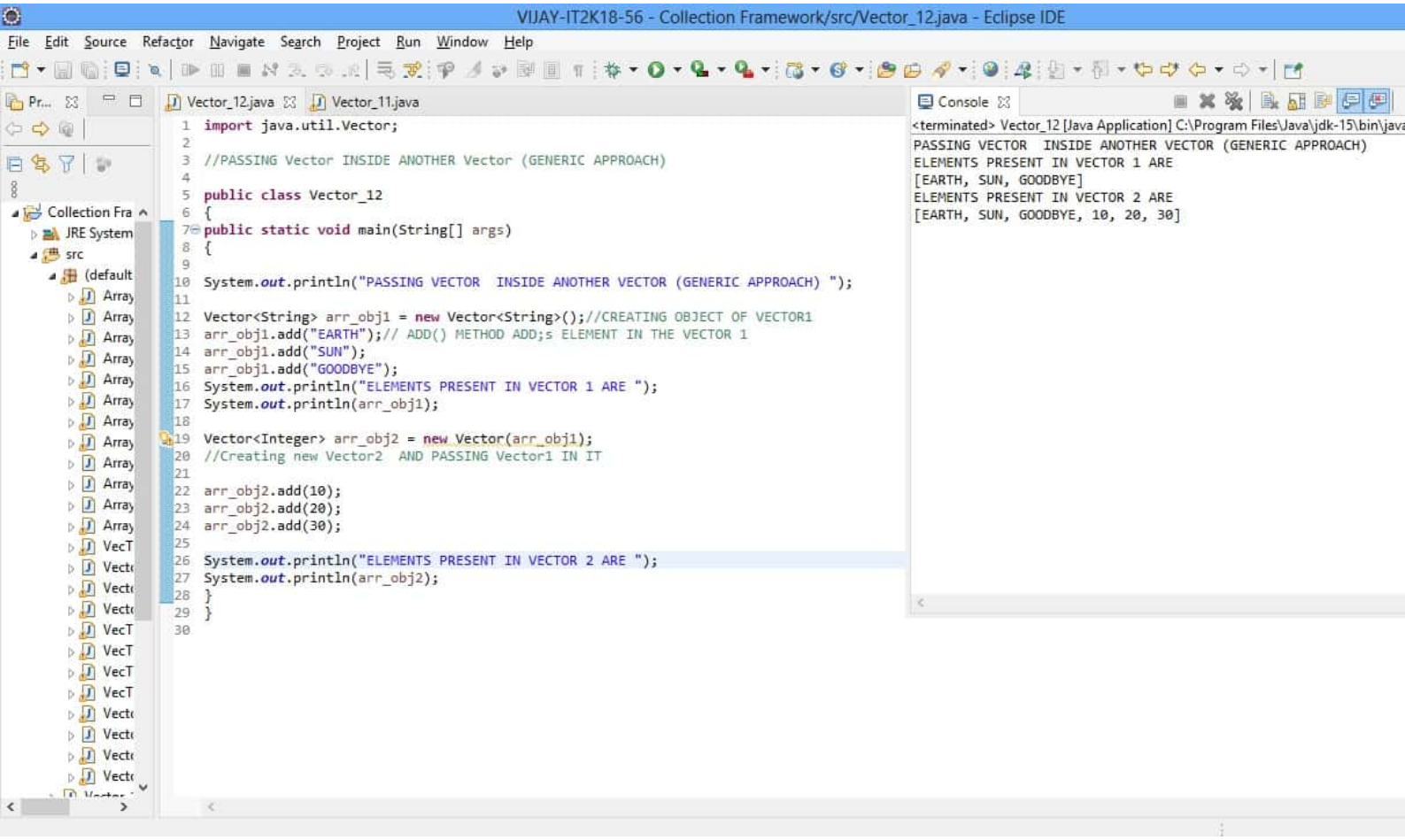


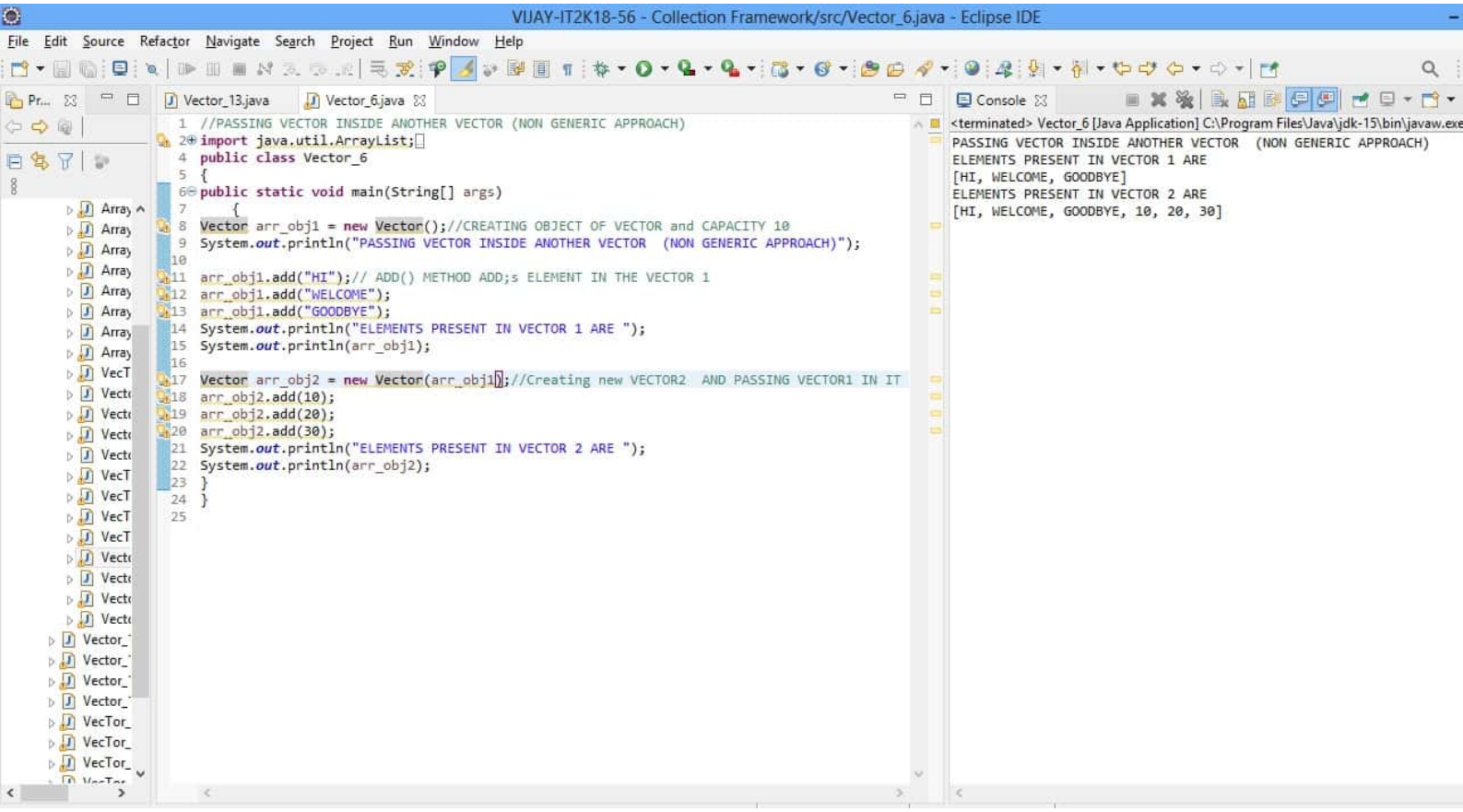


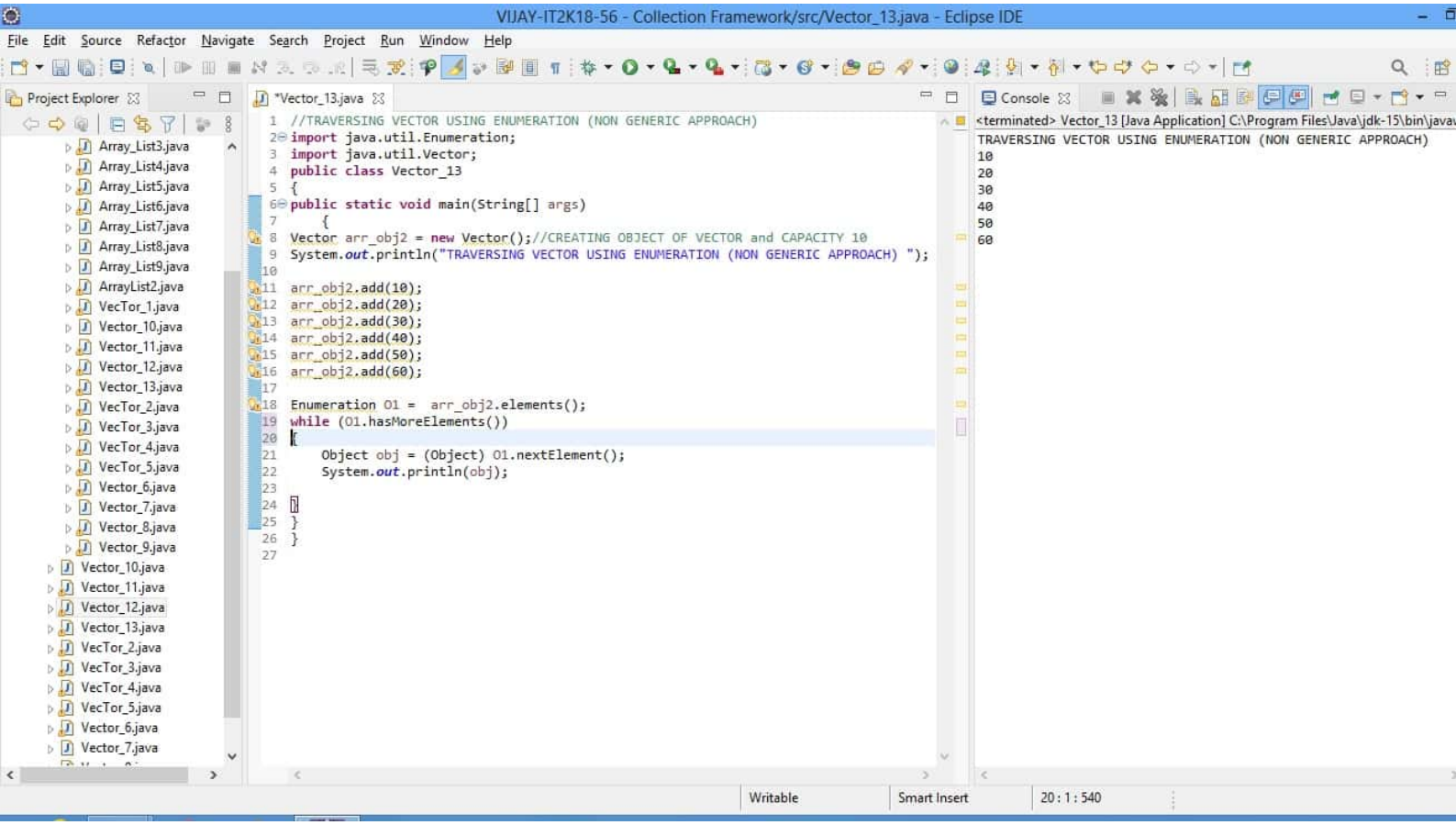


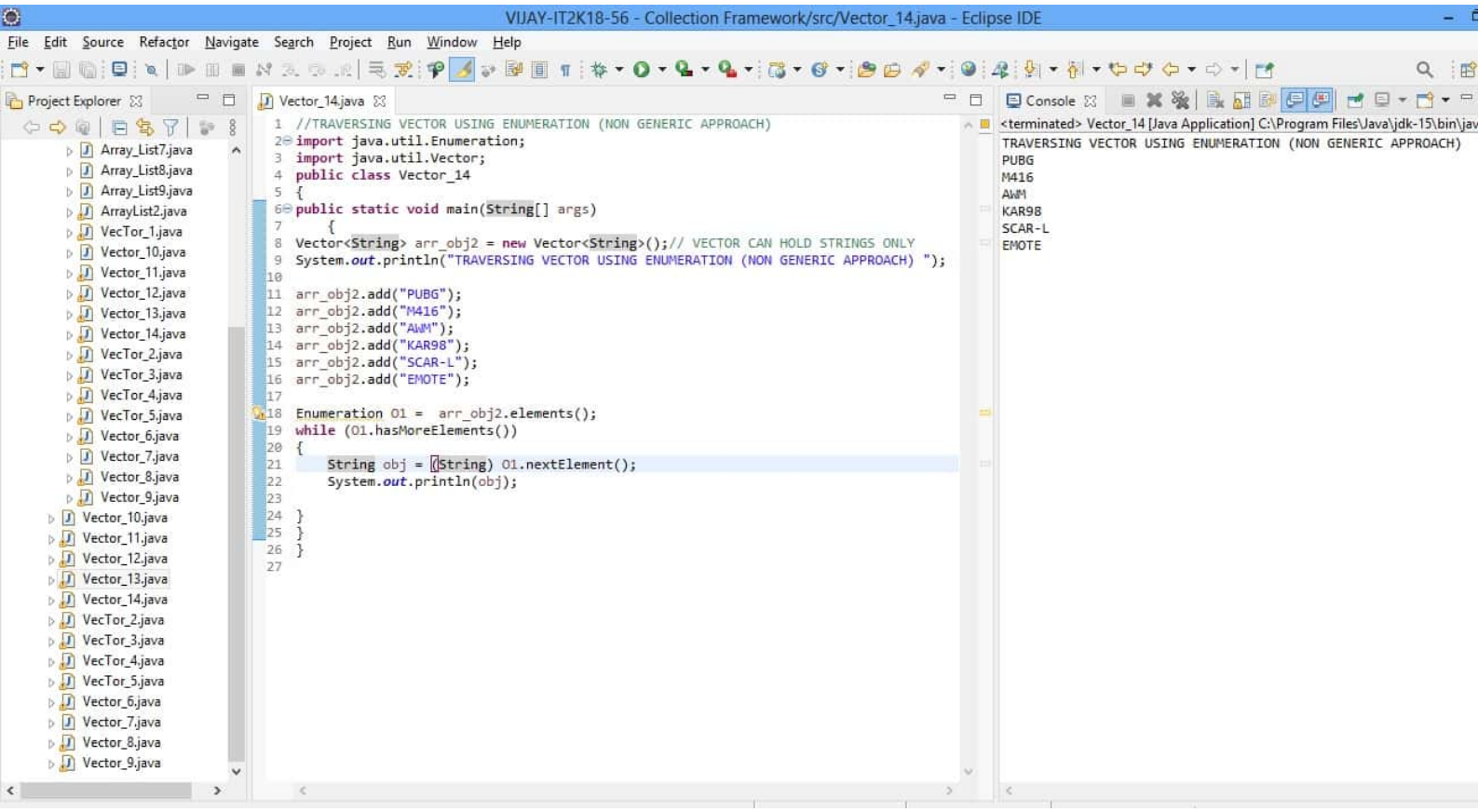

```java
//TRAVERSING  VECTOR (GENERIC APPROACH) USING 3)Iterator

import java.util.Iterator;
import java.util.Vector;


public class Vector_10
{
public static void main(String[] args)
{
    System.out.println("TRAVERSING  VECTOR (GENERIC APPROACH) USING Iterator");
    Vector<Double> arr_obj = new Vector<Double>();
    //NOW Vector will Contain Double values only
    arr_obj.add(3.14);// ADD() METHOD ADD;s ELEMENT IN THE VECTOR
    arr_obj.add(9.11);
    arr_obj.add(16.5);
    arr_obj.add(56.0);

    arr_obj.add(0,9.9);
    //ADD ( INDEX,VALUE) --ADD's ELEMENT AT PARTICULAR INDEX AND SHIFT REST ELEMENTS OF VECTOR

    arr_obj.set(0,36.9);
    //Replaces the element at the specified position in vector with the specified element.

    //***********USING ITERATOR TO TRAVEL IN Vector*****************

    Iterator <Double> obj = arr_obj.iterator();
    while (obj.hasNext()) {
        Double O1 = (Double) obj.next();
        System.out.println(O1);

    }



}

}
```

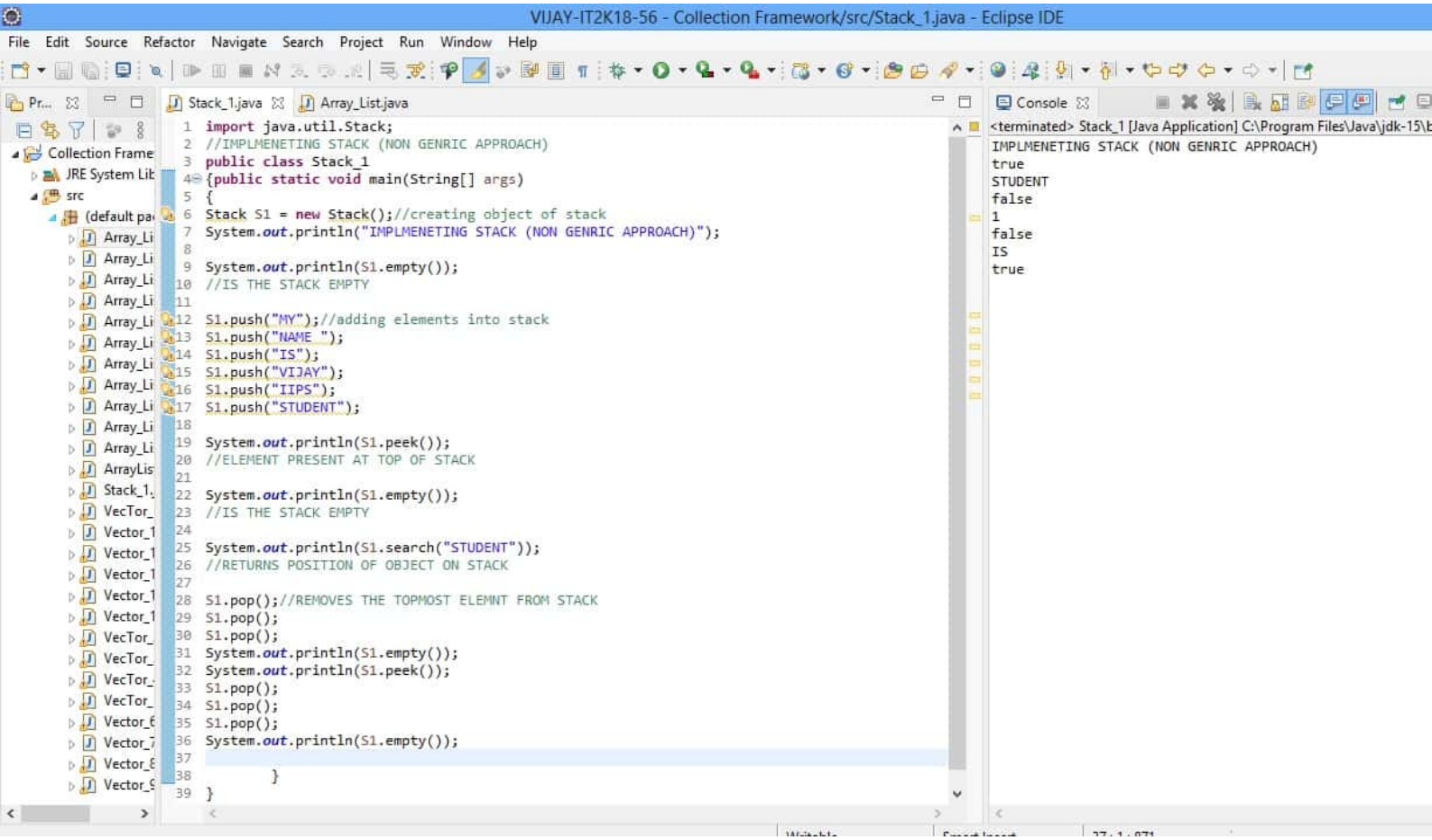




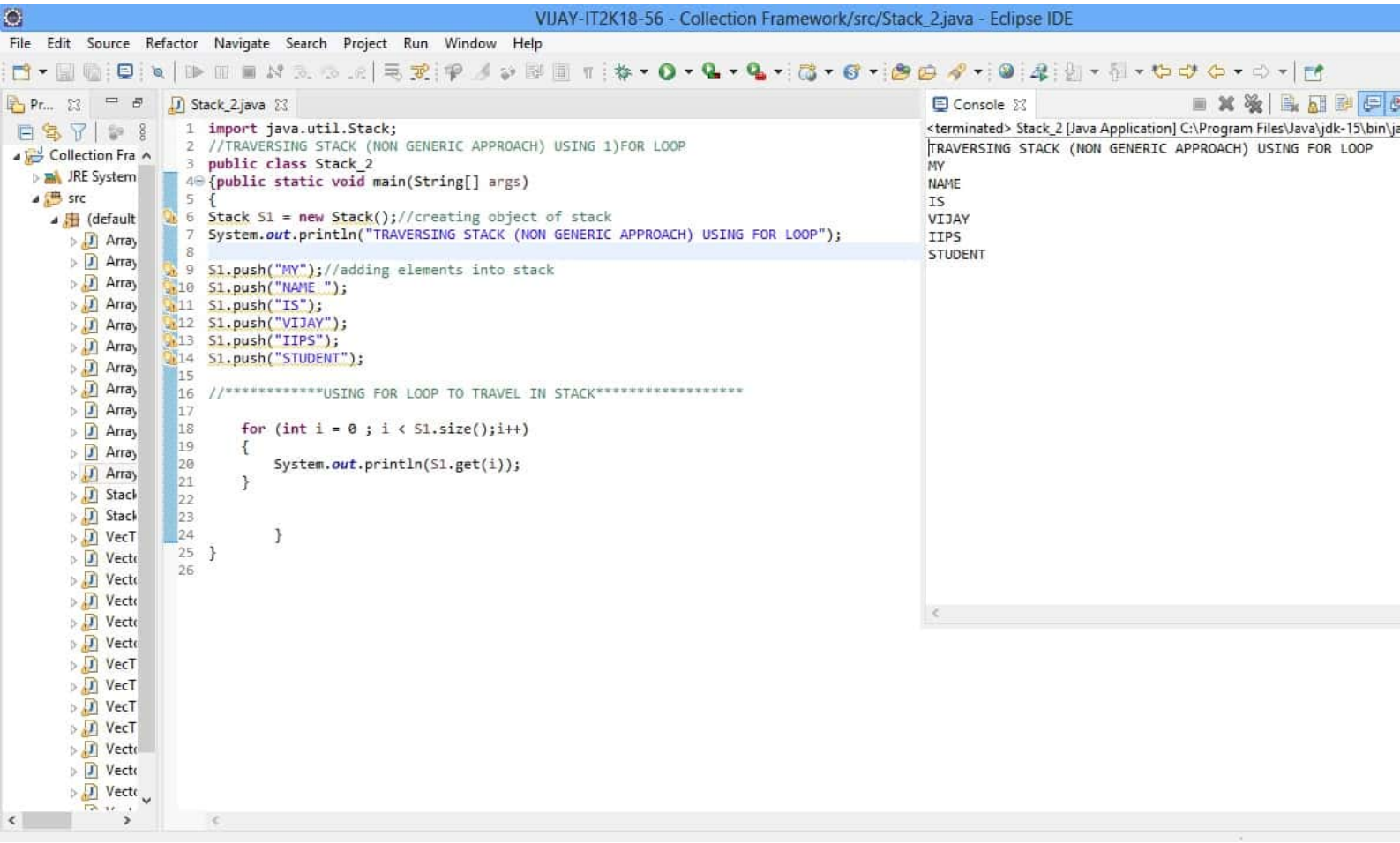


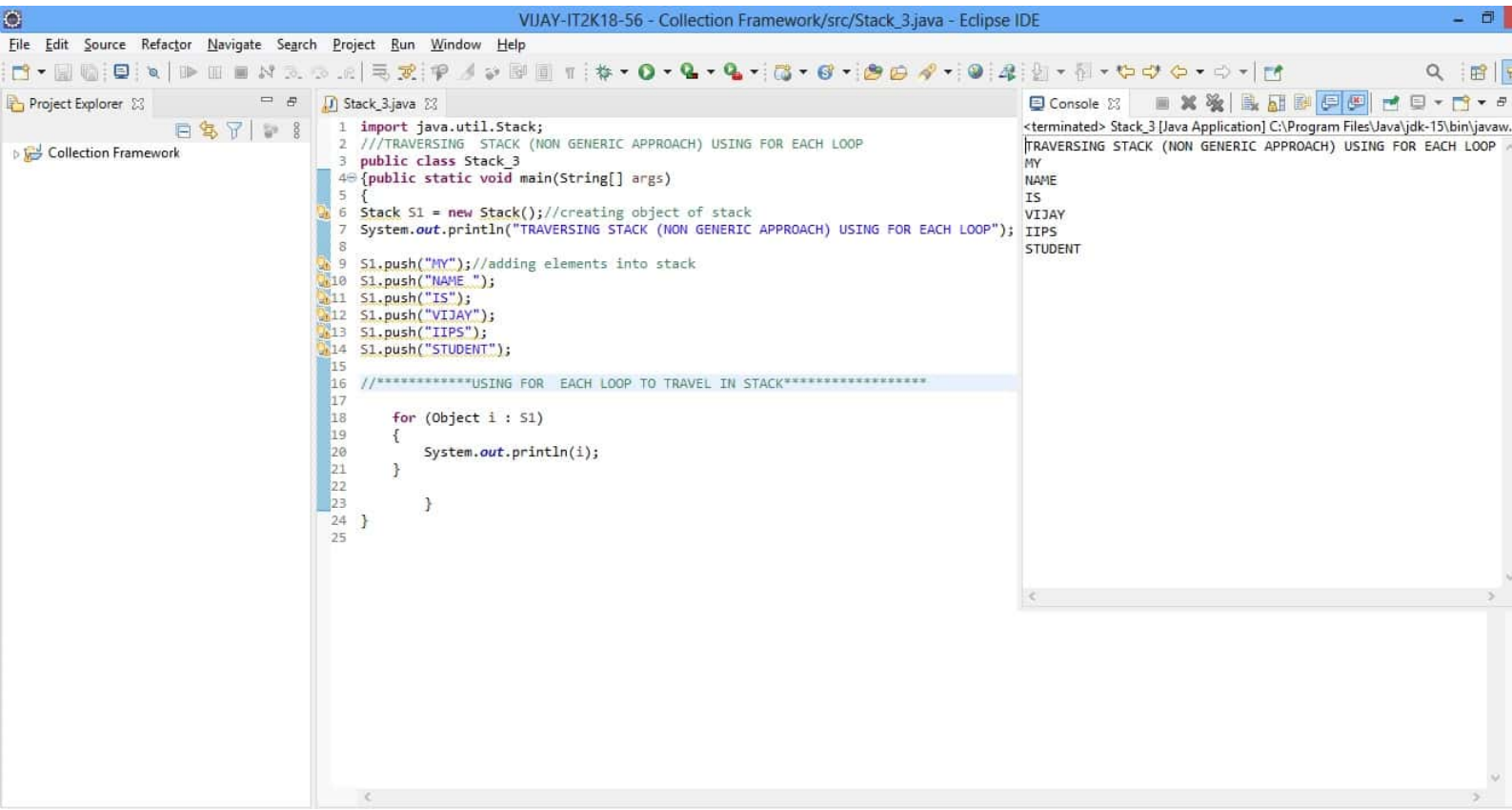


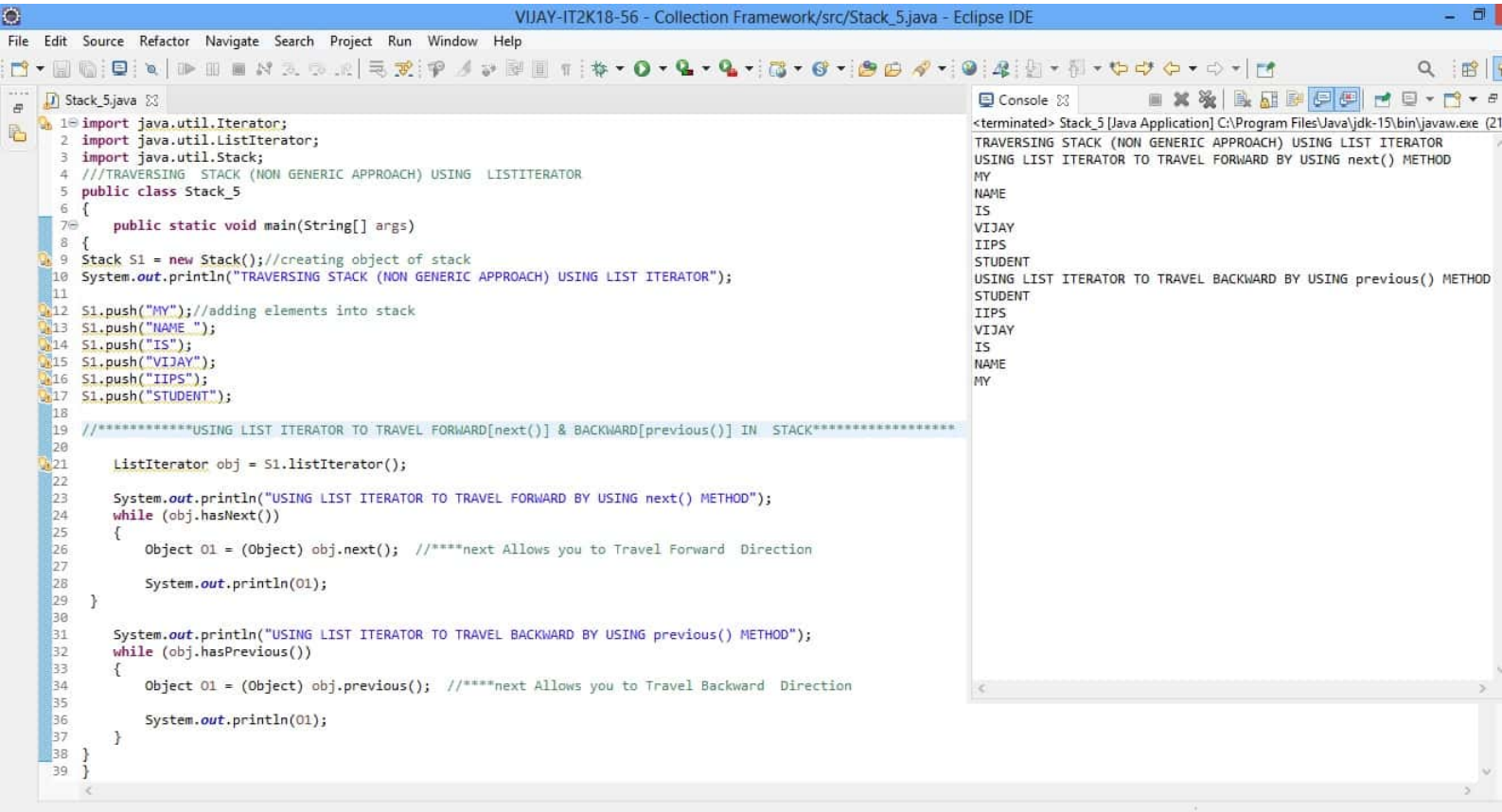


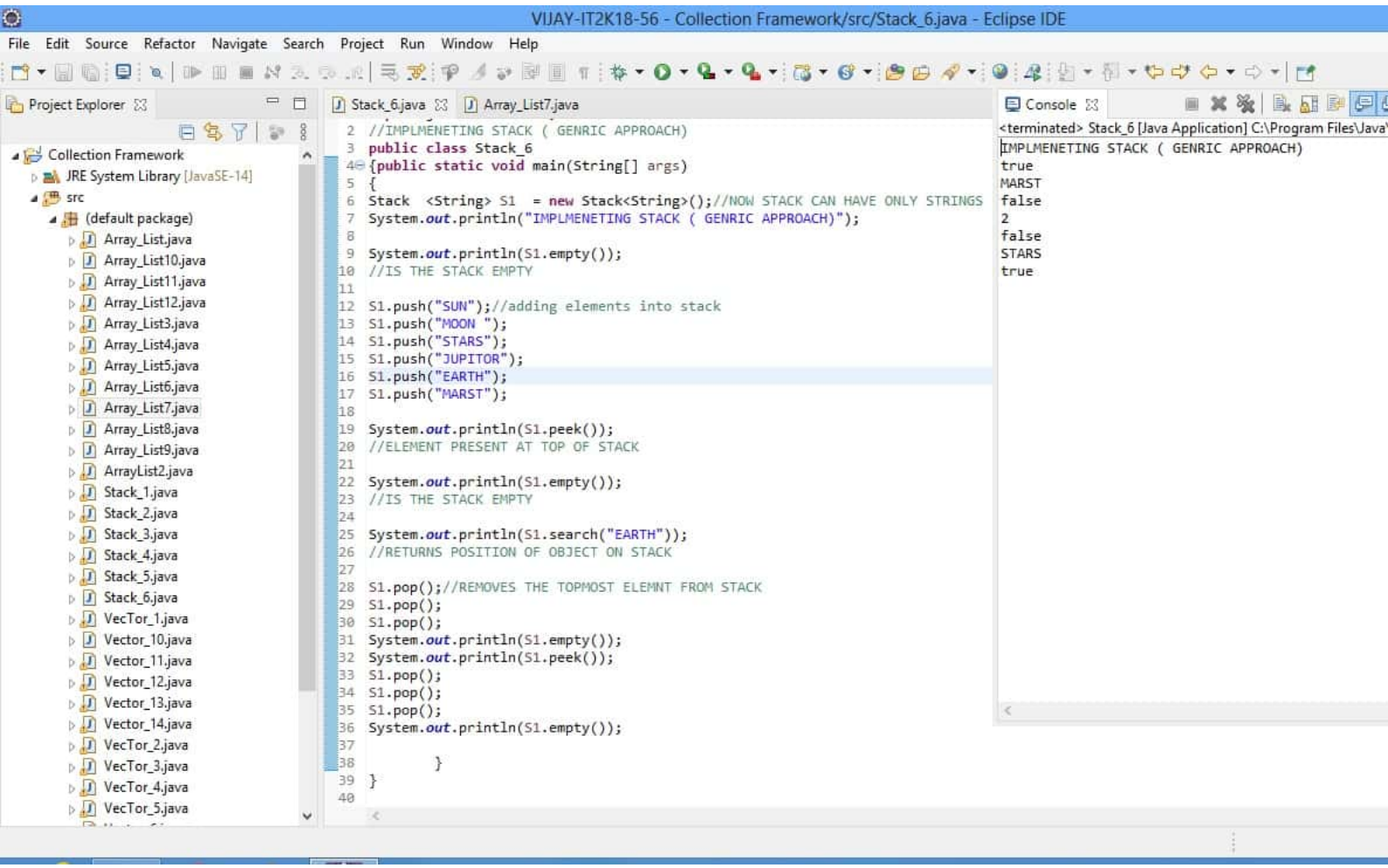


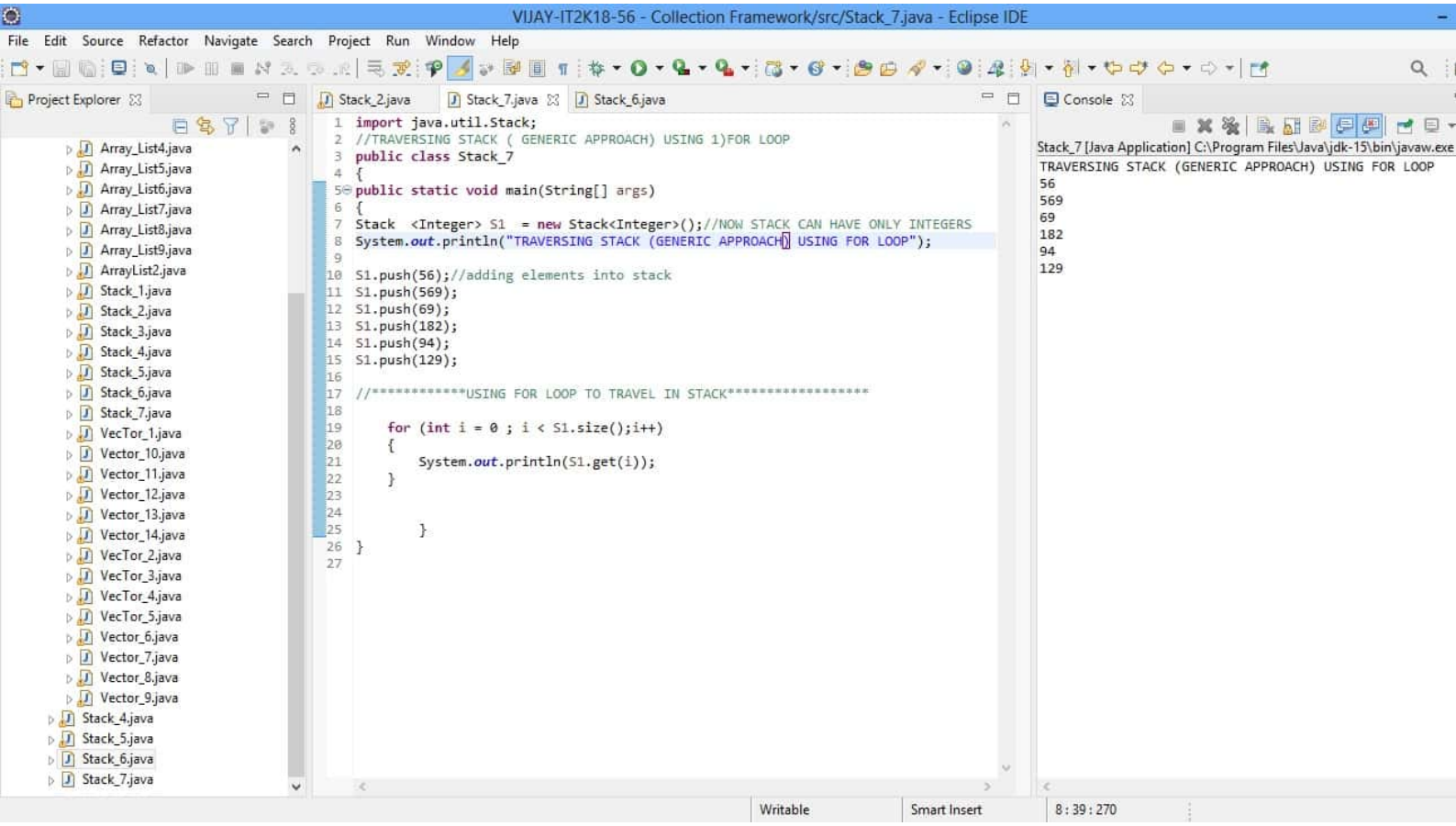

```java
import java.util.Stack;
//IMPLMENETING STACK (NON GENRIC APPROACH)
public class Stack_1
{public static void main(String[] args)
{
Stack S1 = new Stack();//creating object of stack
System.out.println("IMPLMENETING STACK (NON GENRIC APPROACH)");

System.out.println(S1.empty());
//IS THE STACK EMPTY

S1.push("MY");//adding elements into stack
S1.push("NAME ");
S1.push("IS");
S1.push("VIJAY");
S1.push("IIPS");
S1.push("STUDENT");

System.out.println(S1.peek());
//ELEMENT PRESENT AT TOP OF STACK

System.out.println(S1.empty());
//IS THE STACK EMPTY

System.out.println(S1.search("STUDENT"));
//RETURNS POSITION OF OBJECT ON STACK

S1.pop();//REMOVES THE TOPMOST ELEMNT FROM STACK
S1.pop();
S1.pop();
System.out.println(S1.empty());
System.out.println(S1.peek());
S1.pop();
S1.pop();
S1.pop();
System.out.println(S1.empty());

        }
}
```

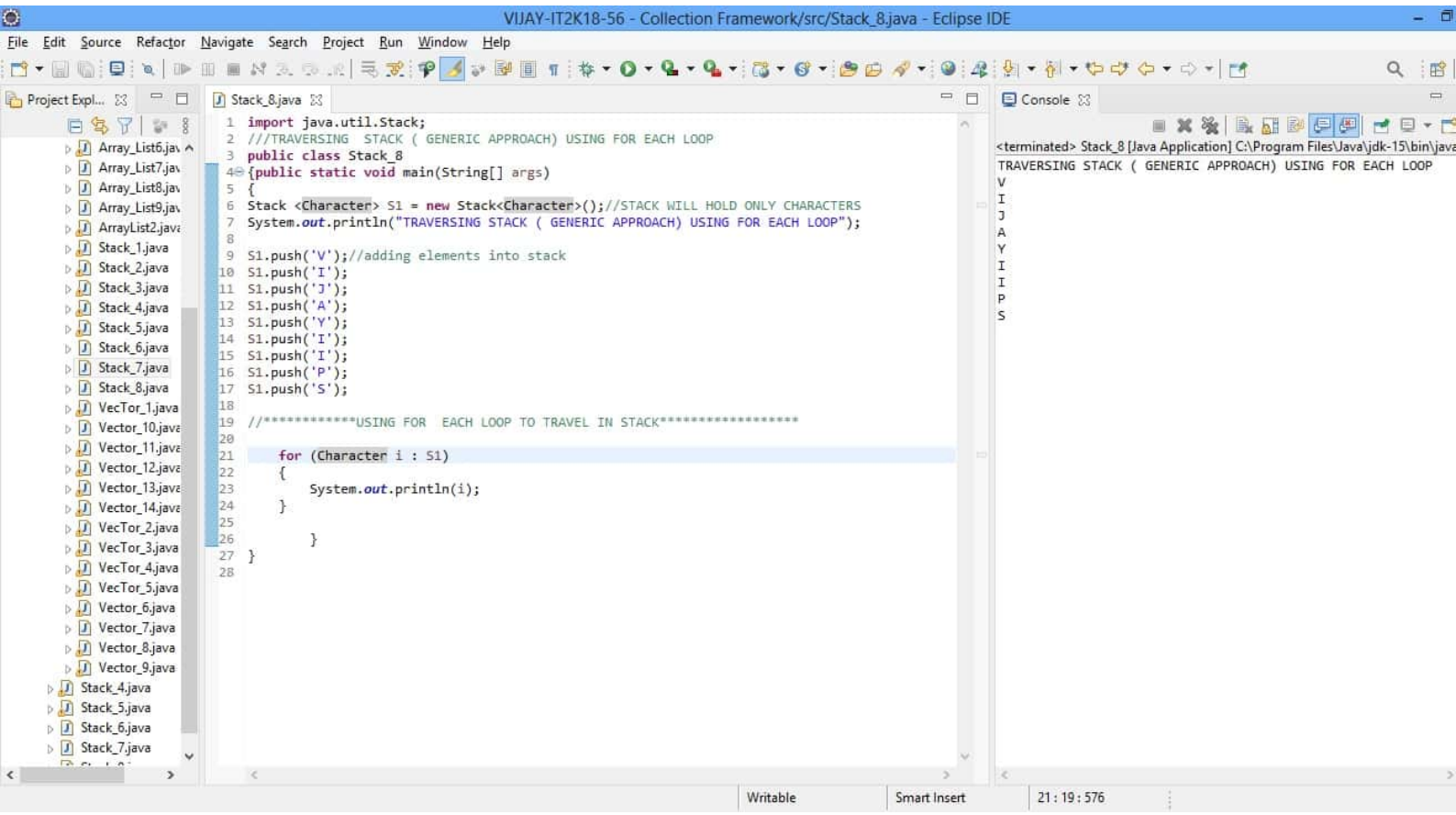


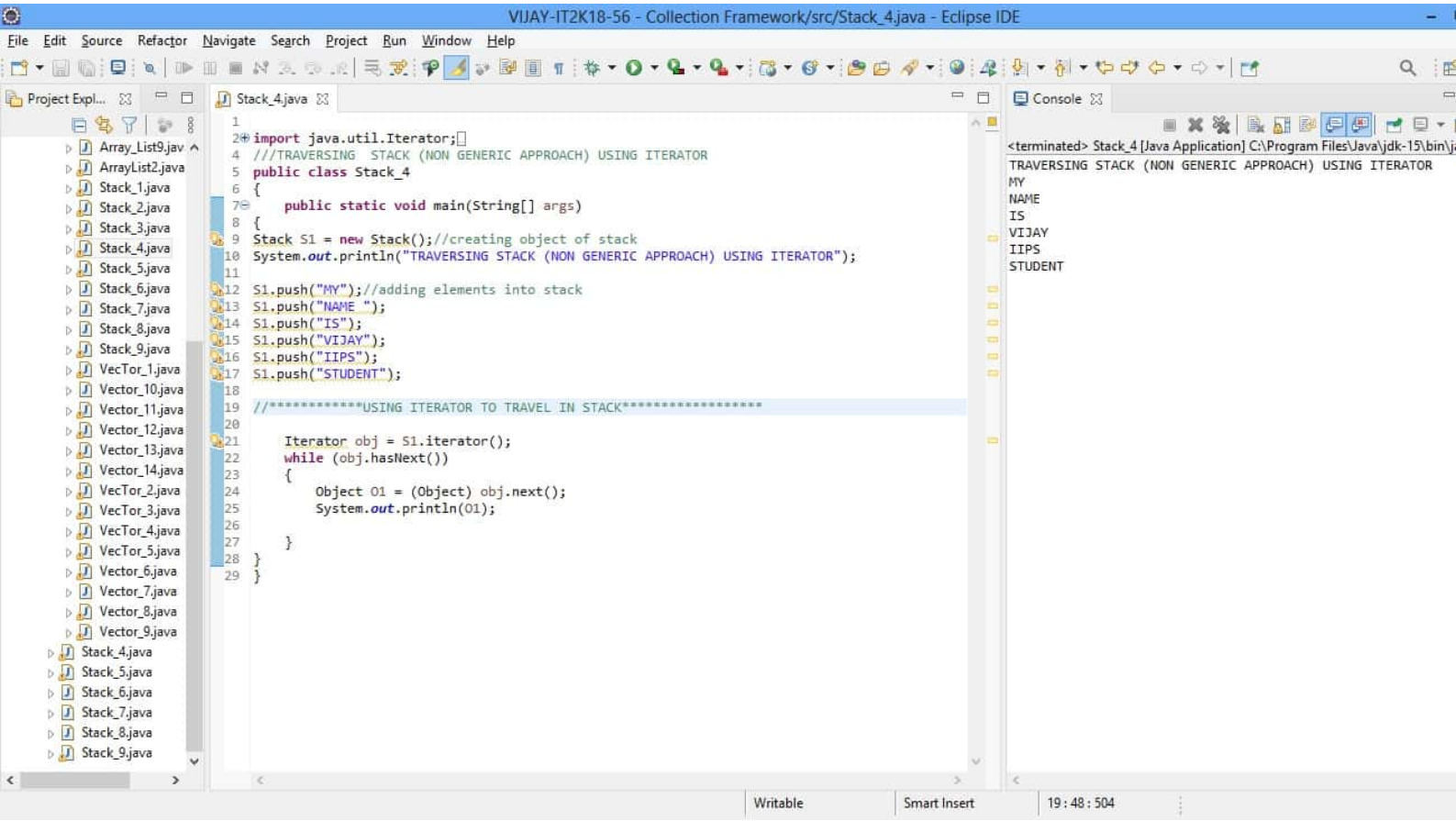


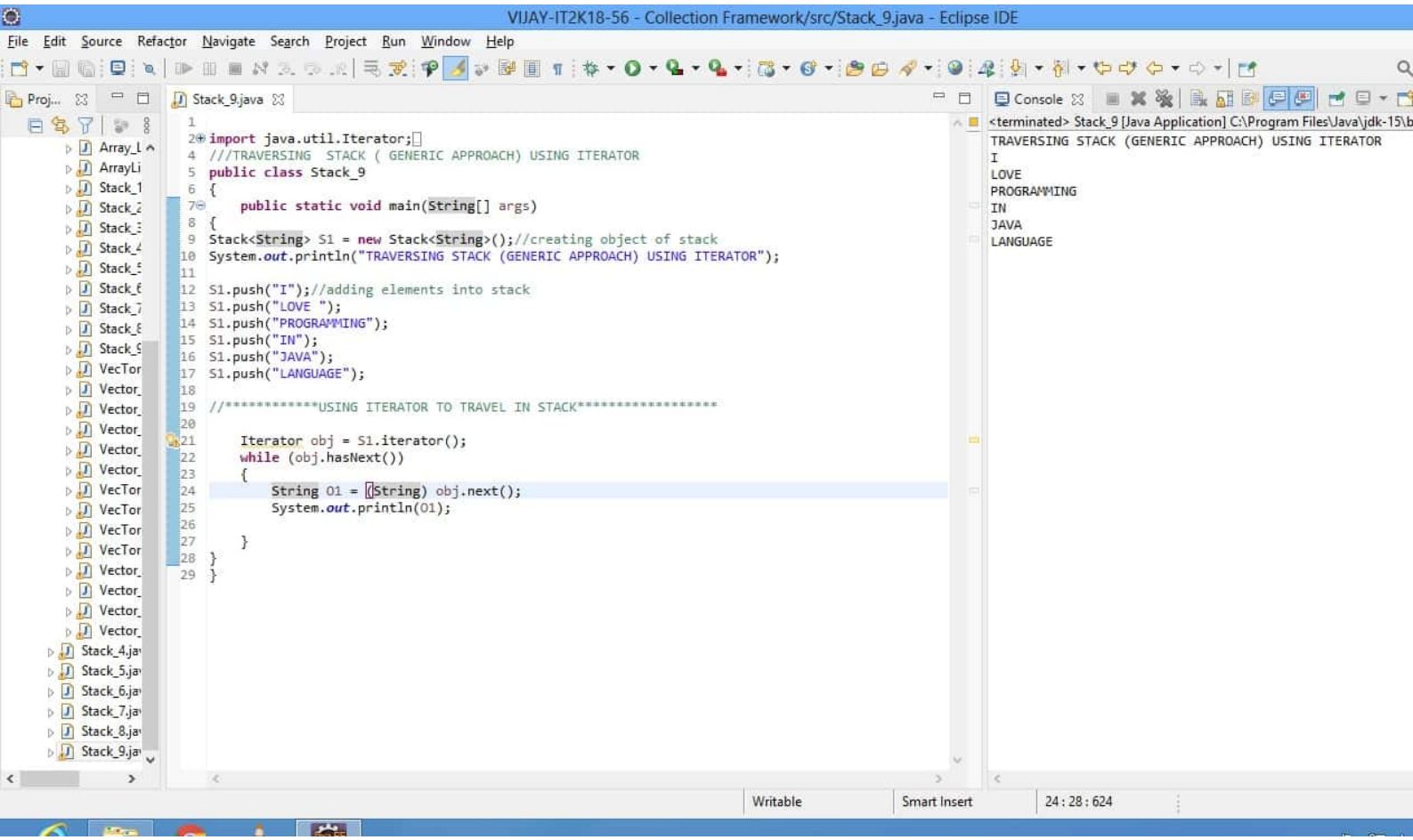


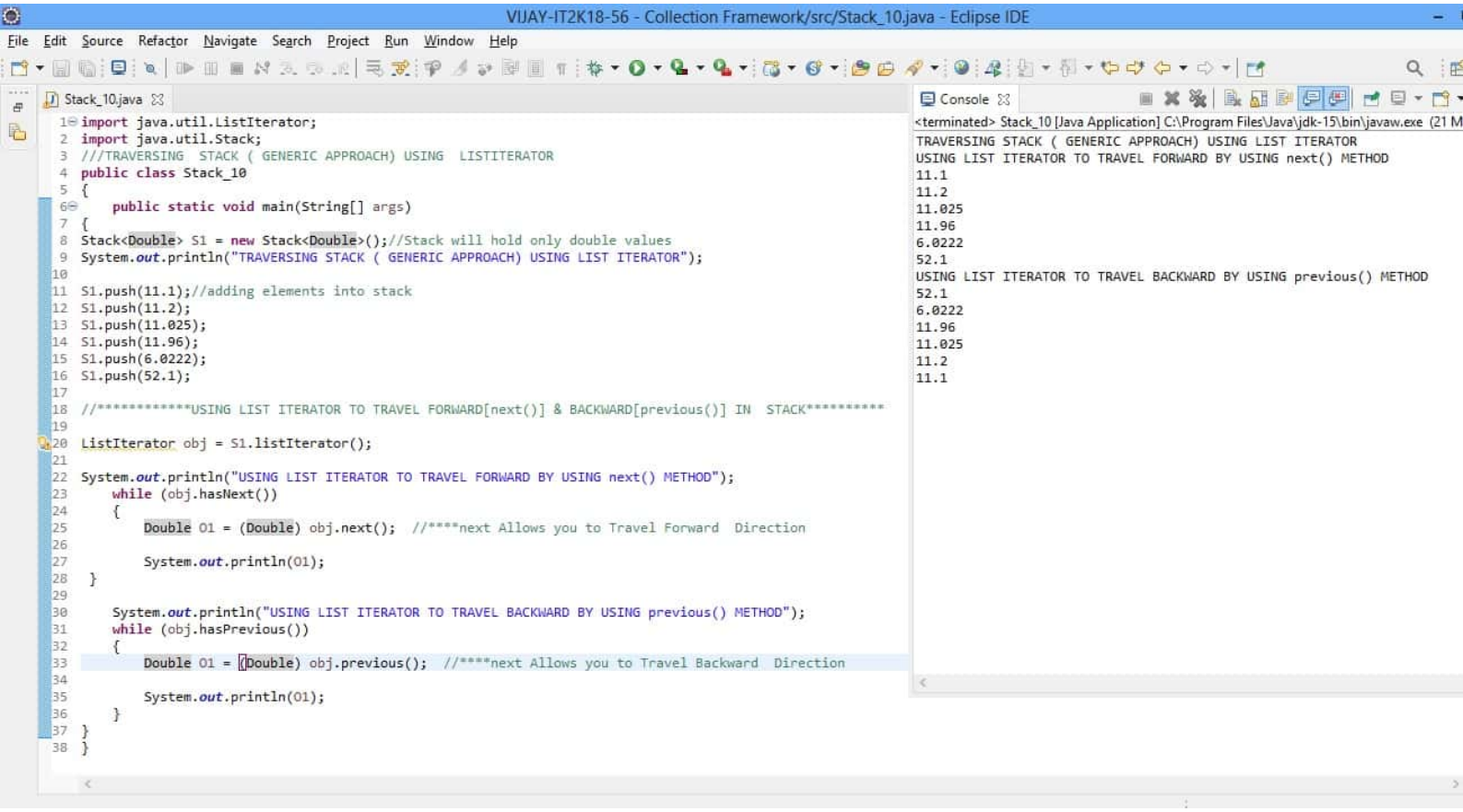


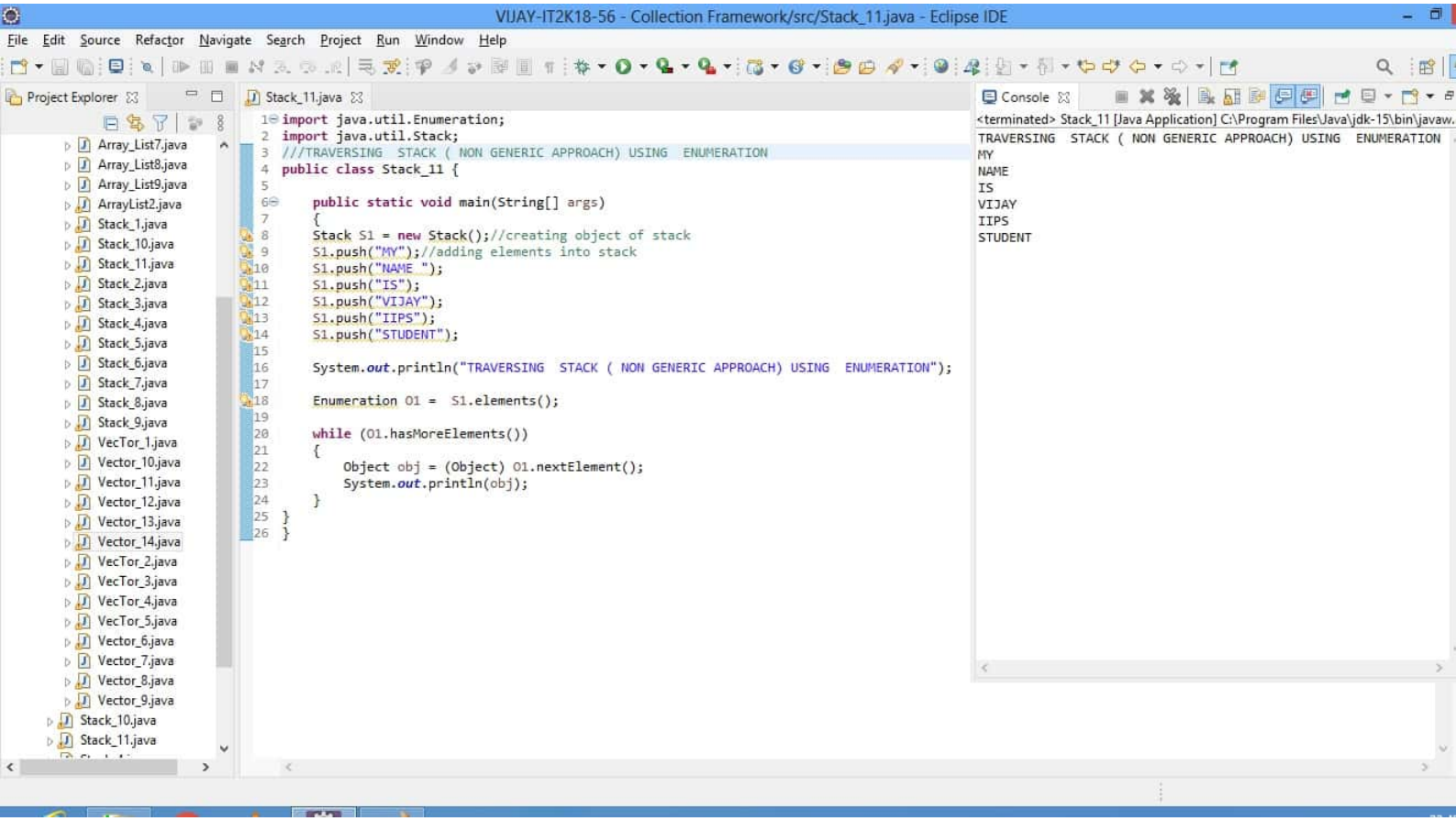


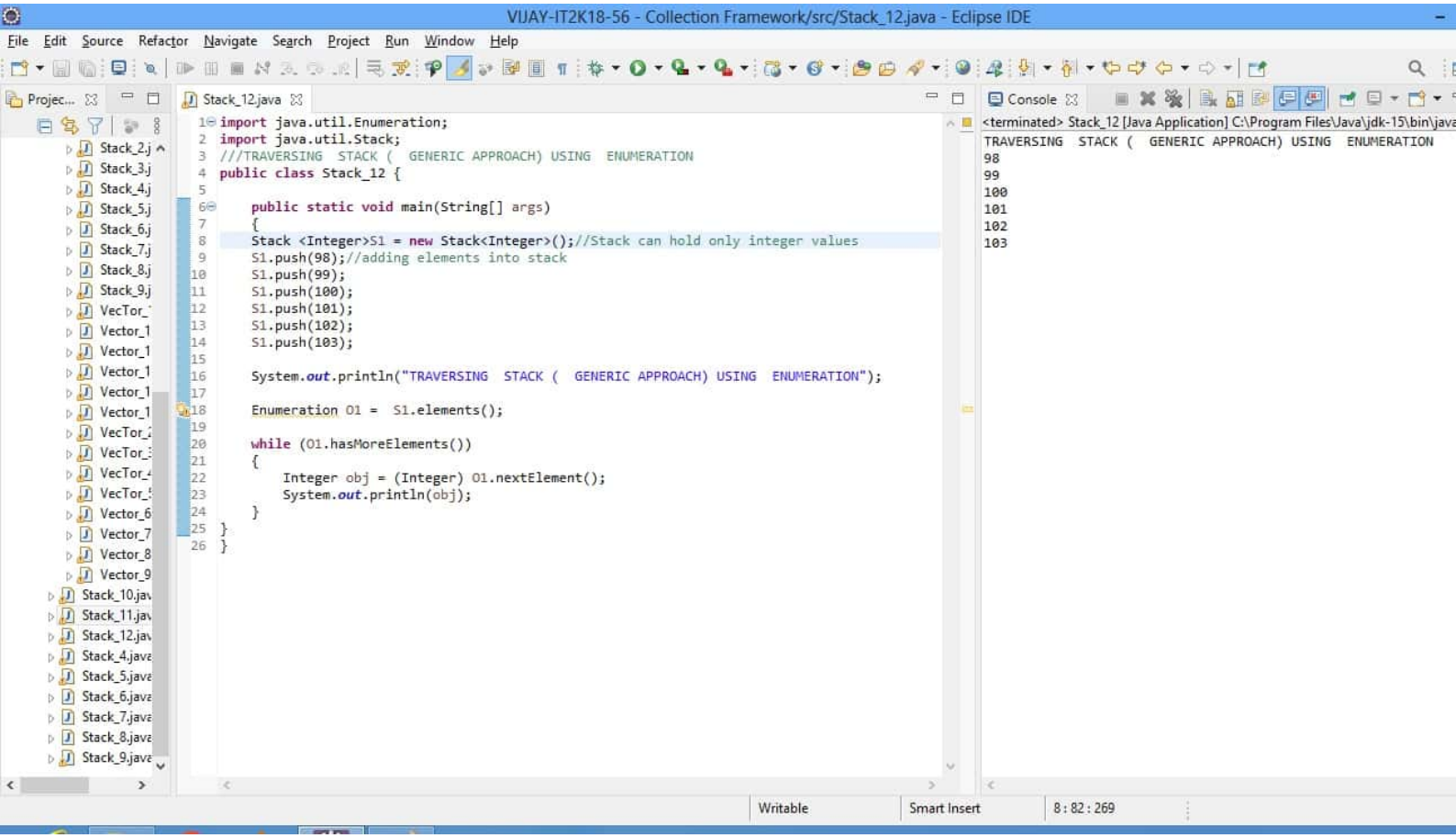


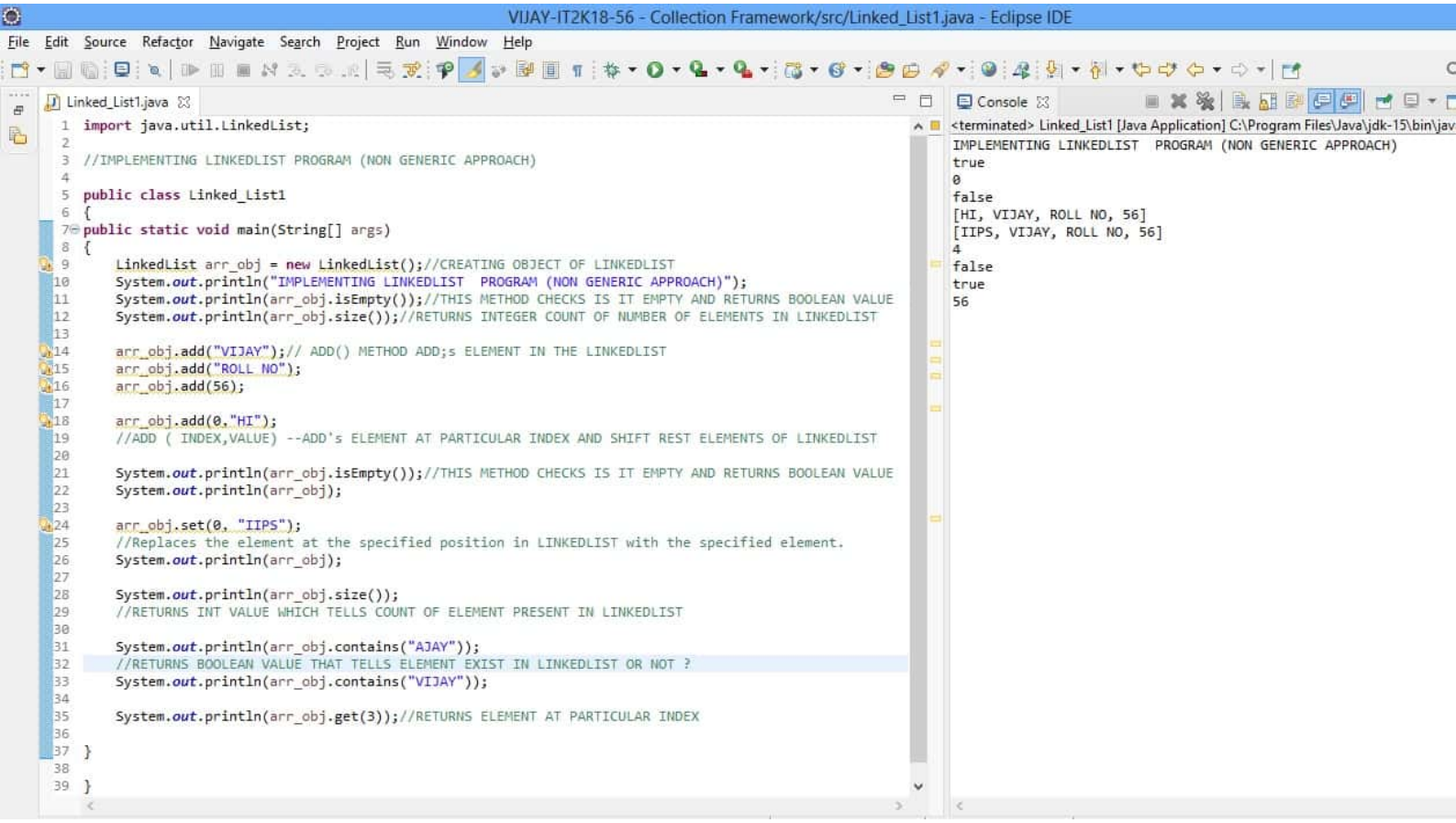


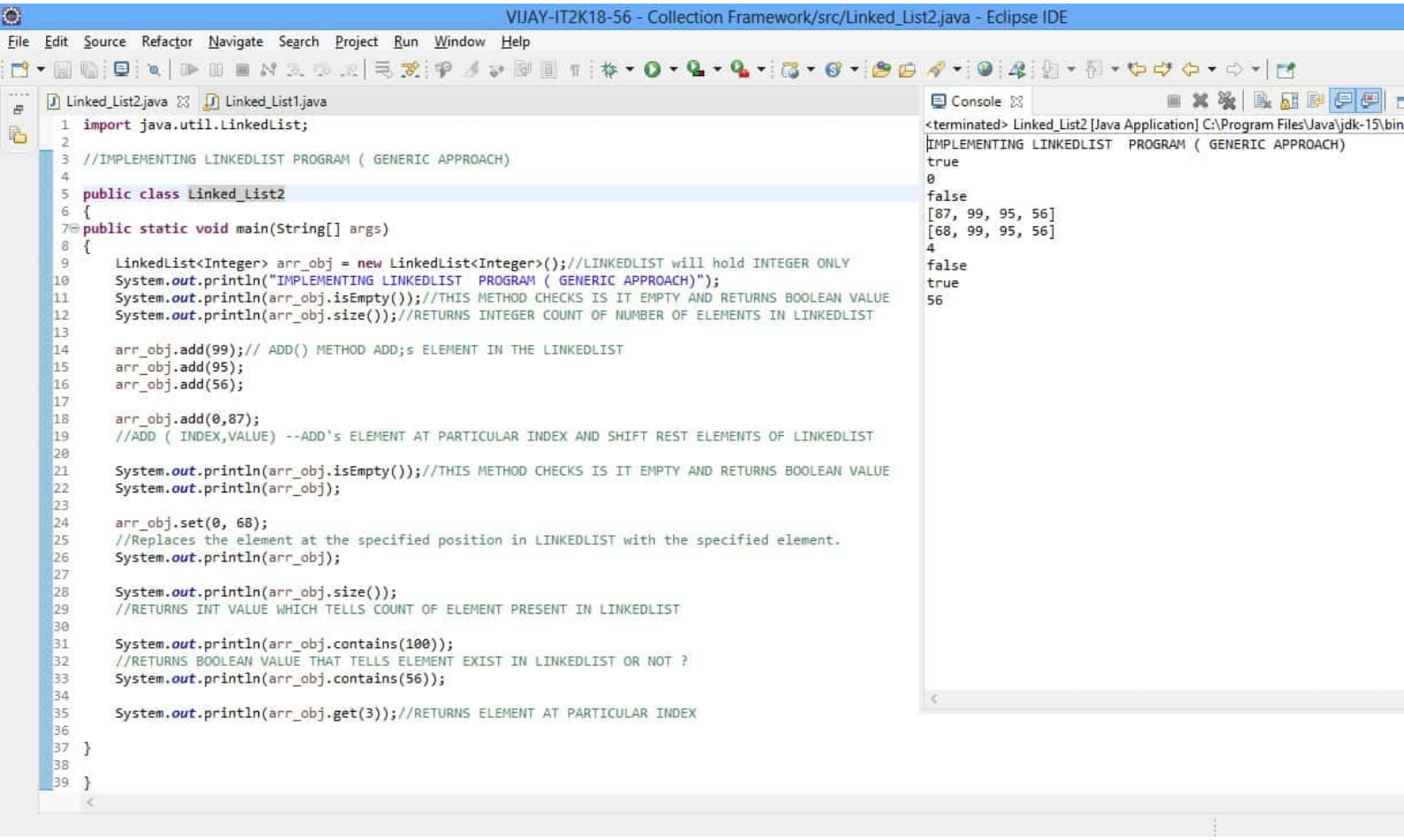




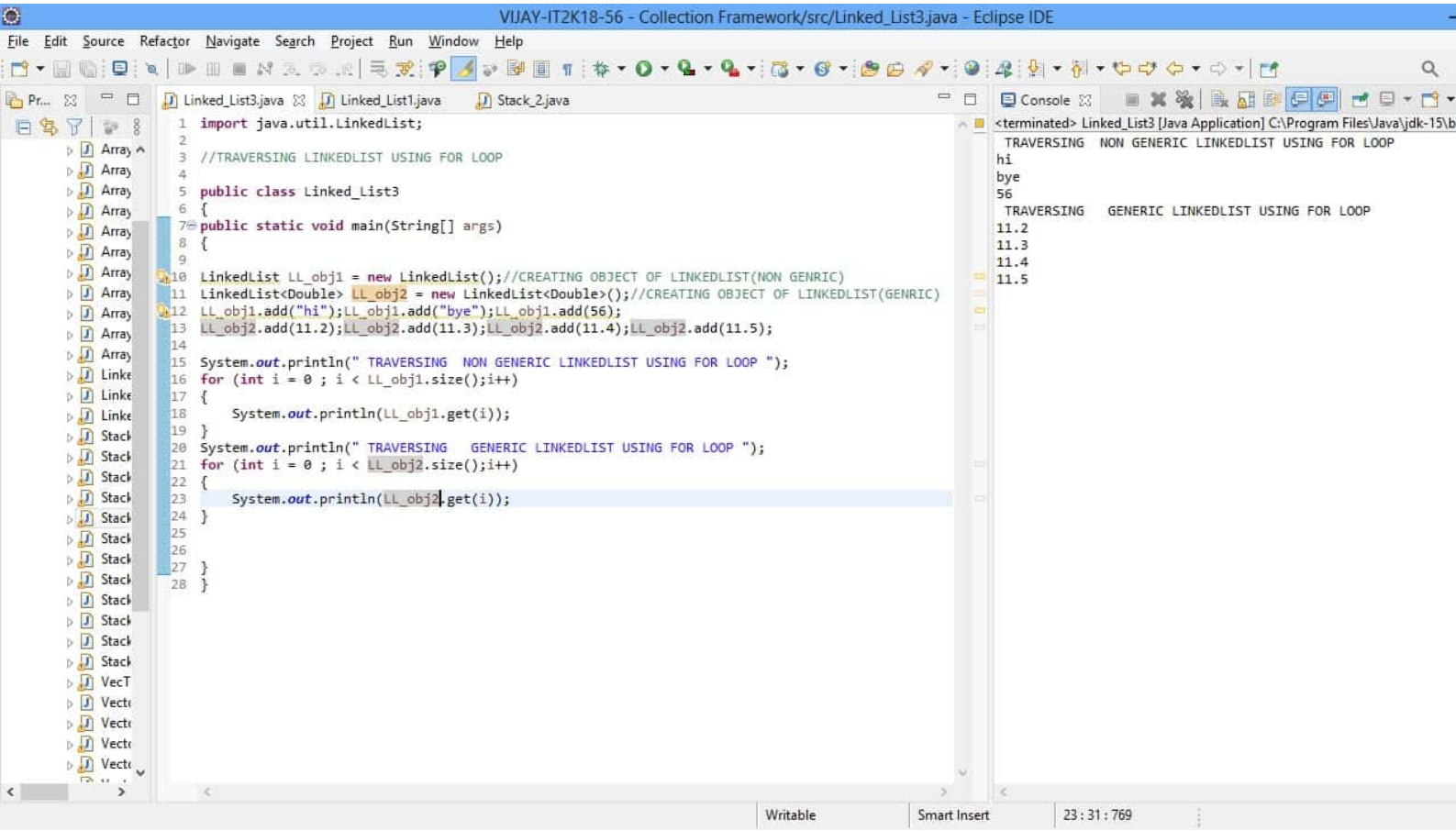


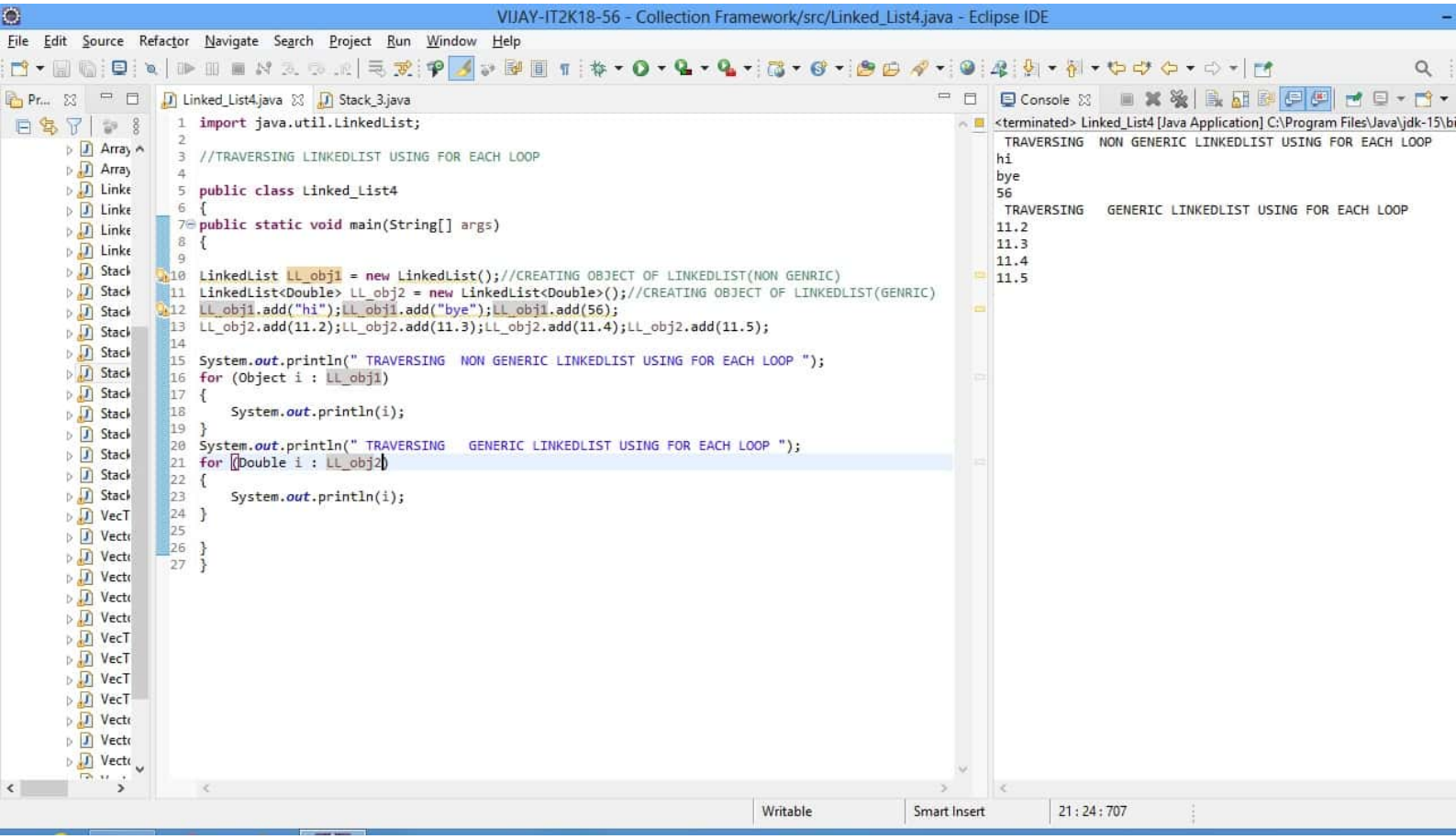


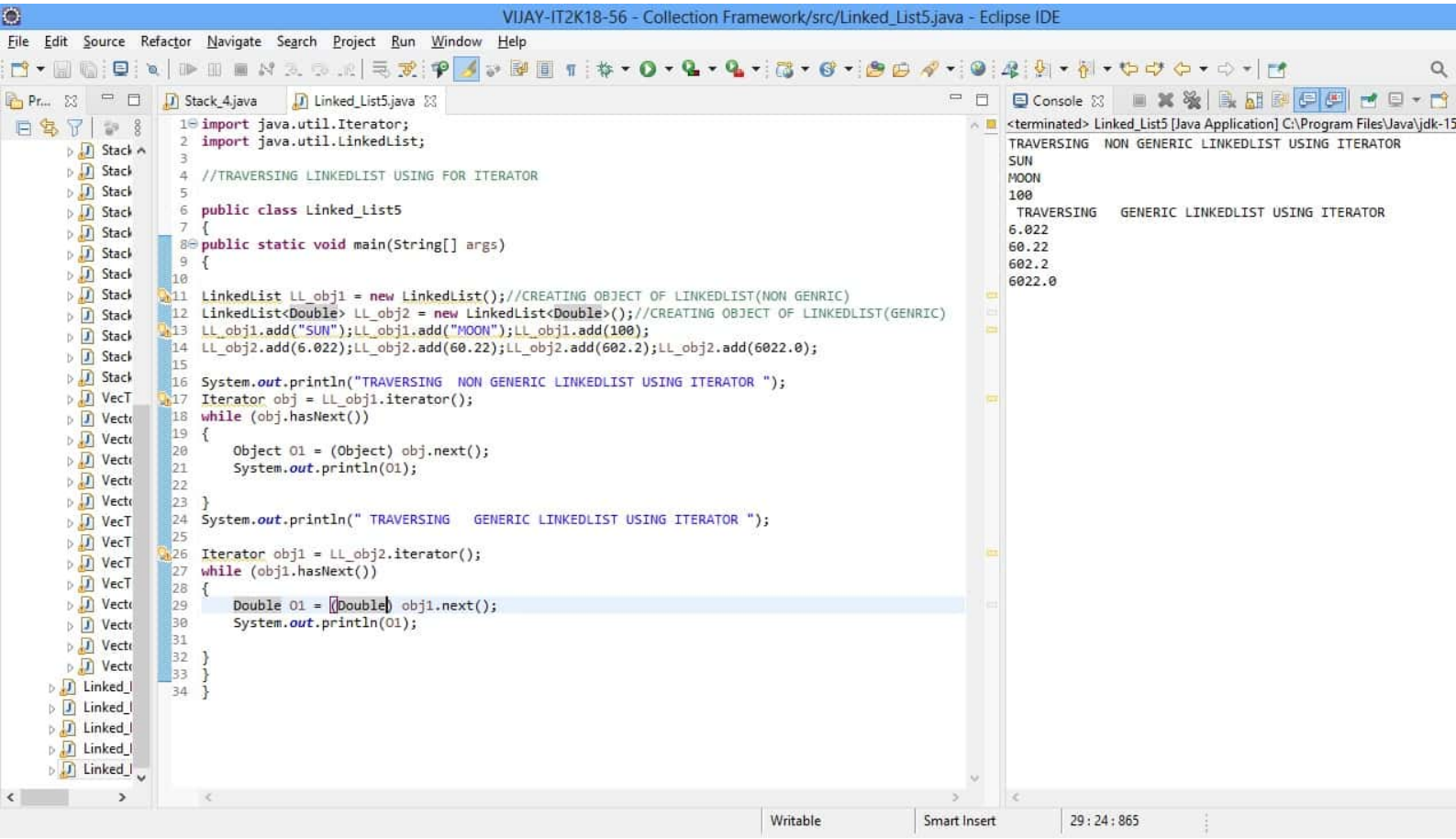


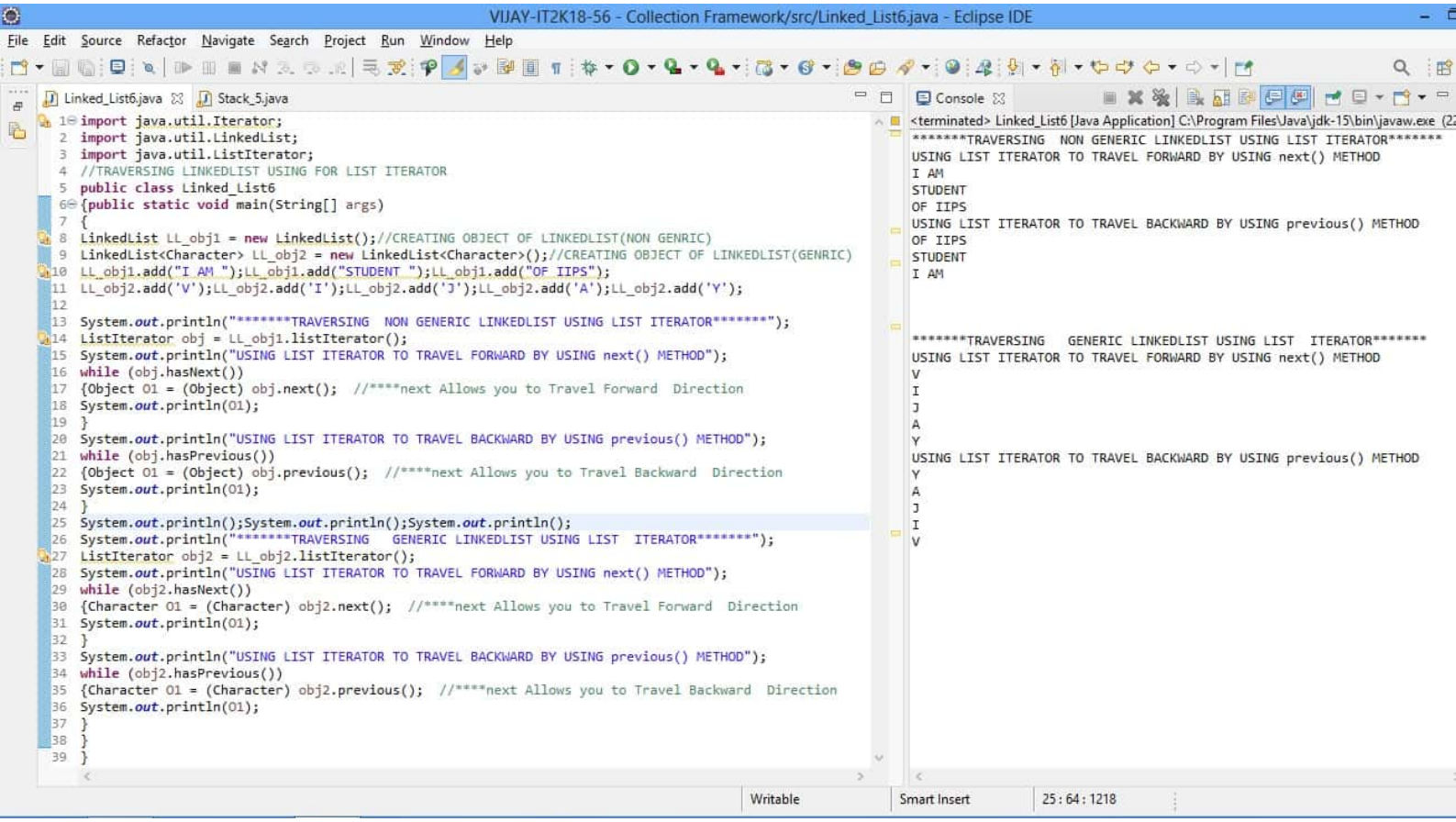


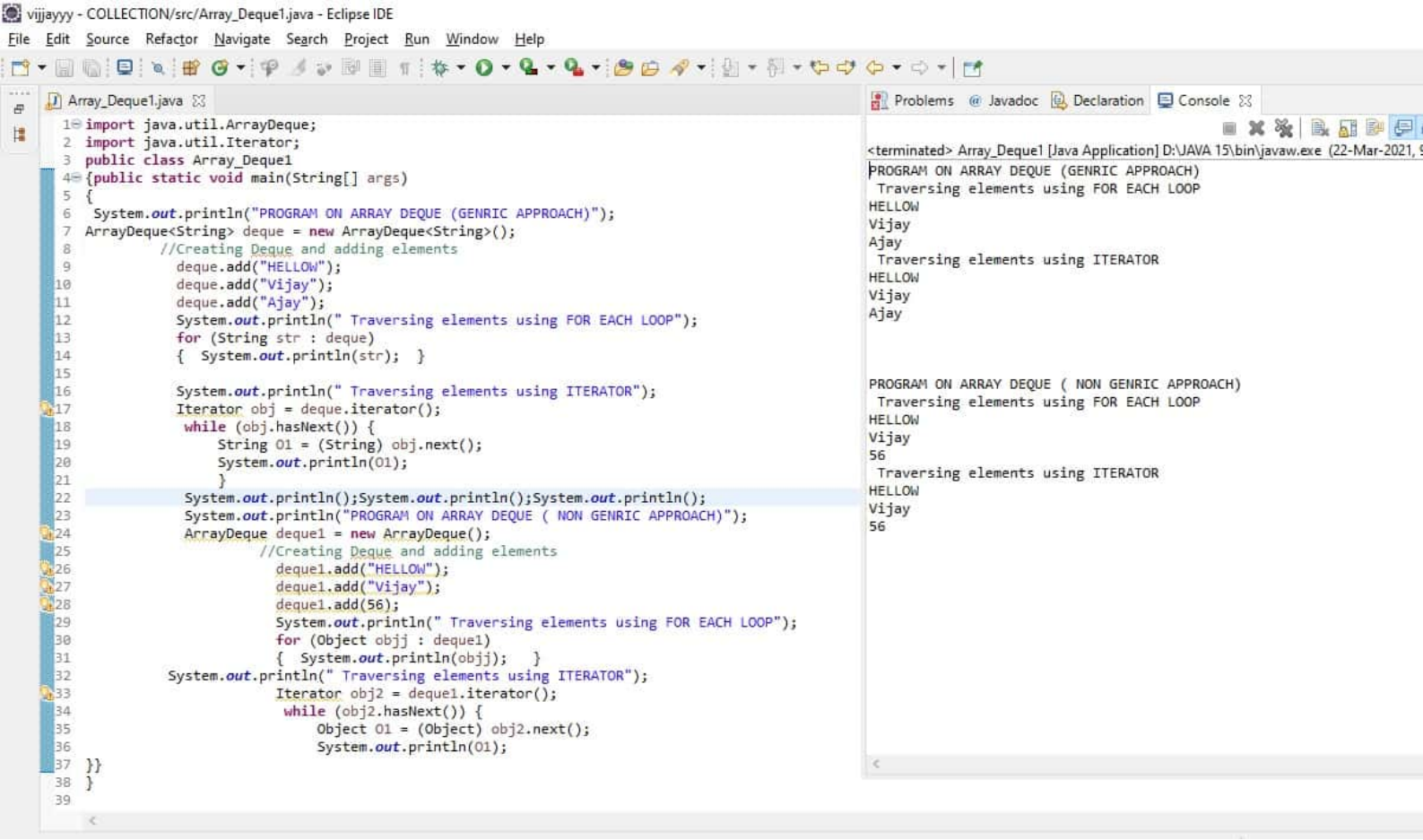

```java
import java.util.LinkedList;

//IMPLEMENTING LINKEDLIST PROGRAM ( GENERIC APPROACH)

public class Linked_List2
{
public static void main(String[] args)
{
    LinkedList<Integer> arr_obj = new LinkedList<Integer>();//LINKEDLIST will hold INTEGER ONLY
    System.out.println("IMPLEMENTING LINKEDLIST  PROGRAM ( GENERIC APPROACH)");
    System.out.println(arr_obj.isEmpty());//THIS METHOD CHECKS IS IT EMPTY AND RETURNS BOOLEAN VALUE
    System.out.println(arr_obj.size());//RETURNS INTEGER COUNT OF NUMBER OF ELEMENTS IN LINKEDLIST

    arr_obj.add(99);// ADD() METHOD ADD;s ELEMENT IN THE LINKEDLIST
    arr_obj.add(95);
    arr_obj.add(56);

    arr_obj.add(0,87);
    //ADD ( INDEX,VALUE) --ADD's ELEMENT AT PARTICULAR INDEX AND SHIFT REST ELEMENTS OF LINKEDLIST

    System.out.println(arr_obj.isEmpty());//THIS METHOD CHECKS IS IT EMPTY AND RETURNS BOOLEAN VALUE
    System.out.println(arr_obj);

    arr_obj.set(0, 68);
    //Replaces the element at the specified position in LINKEDLIST with the specified element.
    System.out.println(arr_obj);

    System.out.println(arr_obj.size());
    //RETURNS INT VALUE WHICH TELLS COUNT OF ELEMENT PRESENT IN LINKEDLIST

    System.out.println(arr_obj.contains(100));
    //RETURNS BOOLEAN VALUE THAT TELLS ELEMENT EXIST IN LINKEDLIST OR NOT ?
    System.out.println(arr_obj.contains(56));

    System.out.println(arr_obj.get(3));//RETURNS ELEMENT AT PARTICULAR INDEX

}

}
```



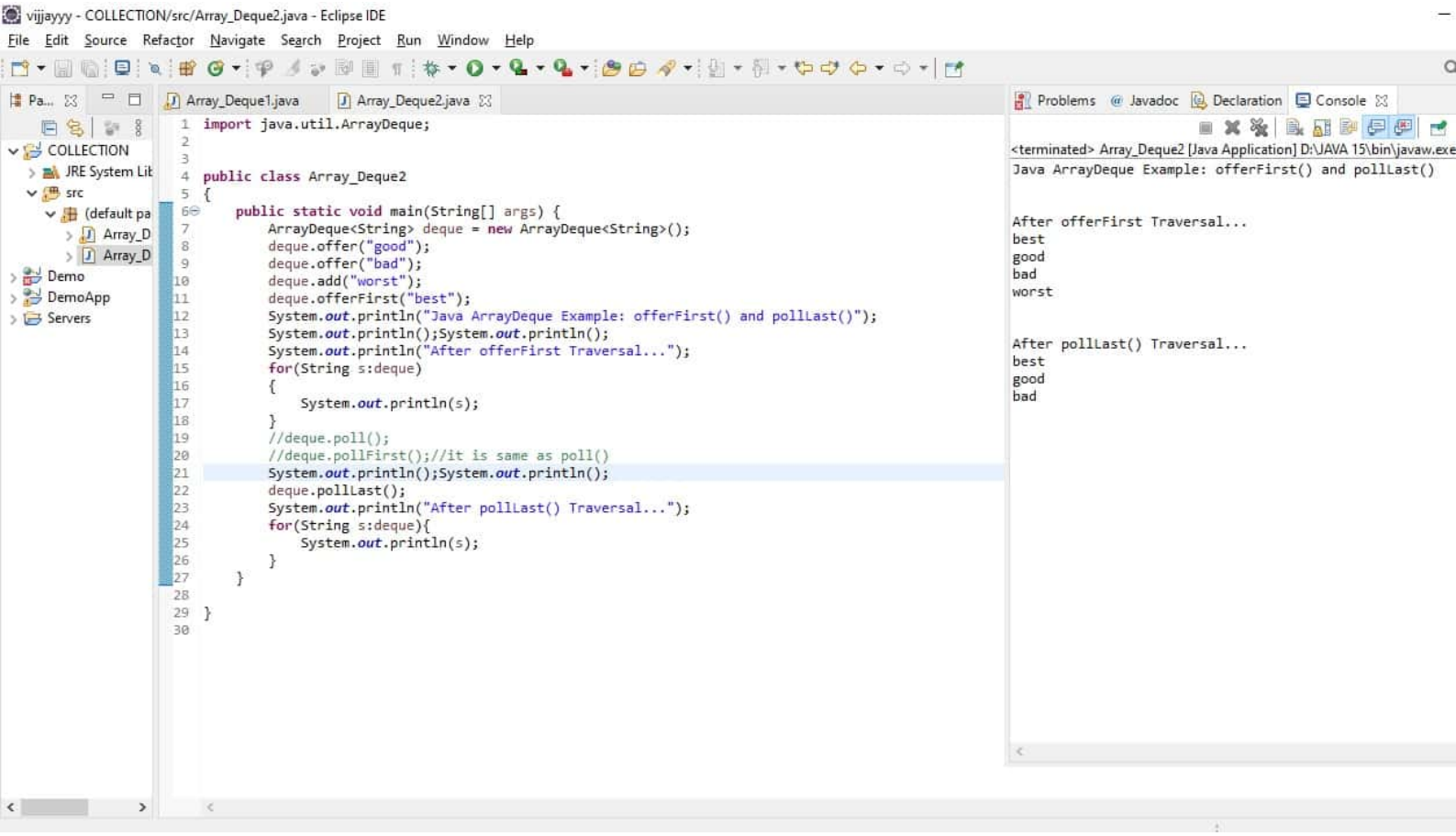


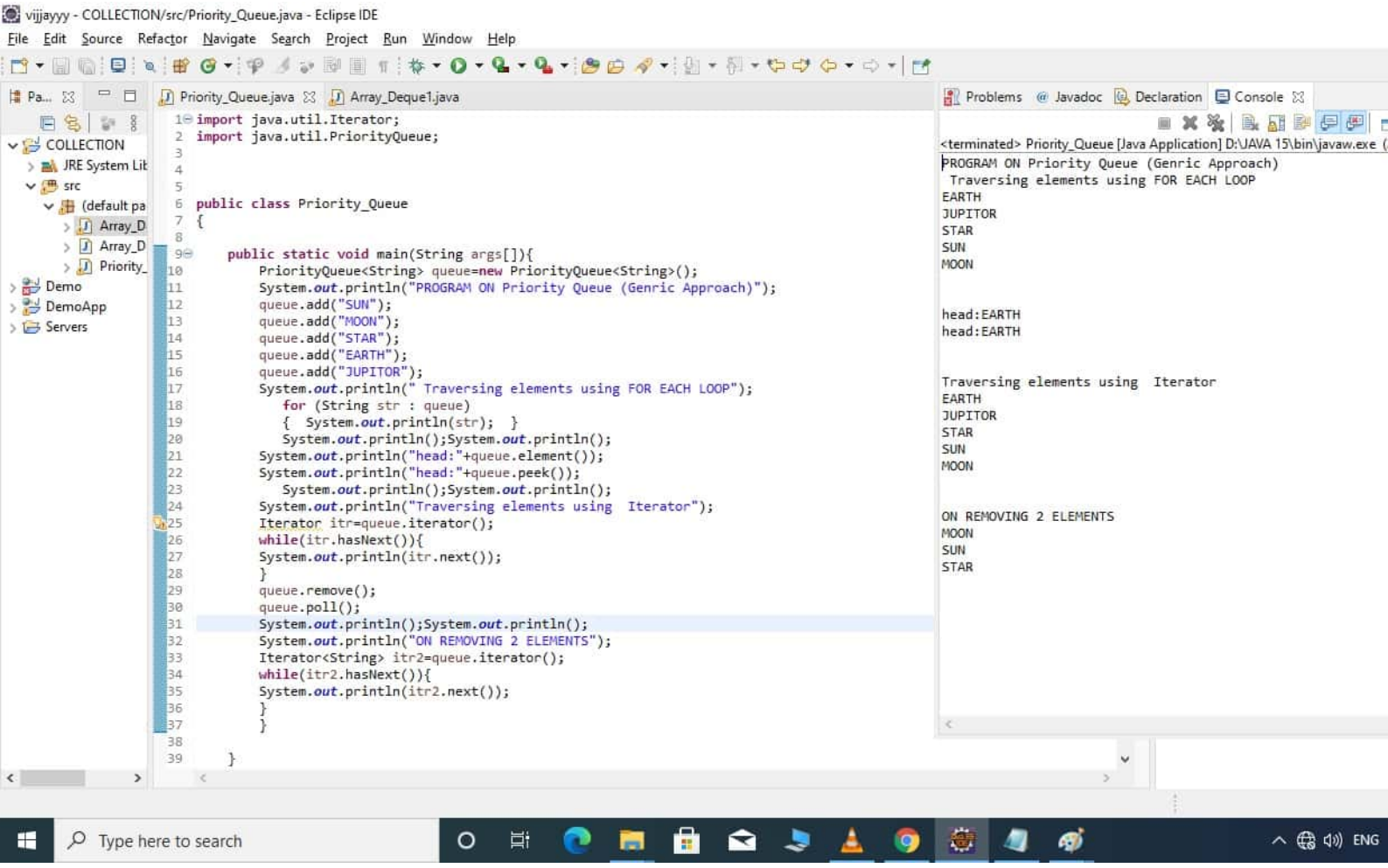







```java
import java.util.ArrayDeque;
import java.util.Iterator;
public class Array_Deque1
{public static void main(String[] args)
{
 System.out.println("PROGRAM ON ARRAY DEQUE (GENRIC APPROACH)");
ArrayDeque<String> deque = new ArrayDeque<String>();
         //Creating Deque and adding elements
           deque.add("HELLOW");
           deque.add("Vijay");
           deque.add("Ajay");
           System.out.println(" Traversing elements using FOR EACH LOOP");
           for (String str : deque)
           {  System.out.println(str);  }

           System.out.println(" Traversing elements using ITERATOR");
           Iterator obj = deque.iterator();
            while (obj.hasNext()) {
                String O1 = (String) obj.next();
                System.out.println(O1);
                }
            System.out.println();System.out.println();System.out.println();
            System.out.println("PROGRAM ON ARRAY DEQUE ( NON GENRIC APPROACH)");
            ArrayDeque deque1 = new ArrayDeque();
                     //Creating Deque and adding elements
                       deque1.add("HELLOW");
                       deque1.add("Vijay");
                       deque1.add(56);
                       System.out.println(" Traversing elements using FOR EACH LOOP");
                       for (Object objj : deque1)
                       {  System.out.println(objj);   }
          System.out.println(" Traversing elements using ITERATOR");
                       Iterator obj2 = deque1.iterator();
                        while (obj2.hasNext()) {
                            Object O1 = (Object) obj2.next();
                            System.out.println(O1);
}}
}
```






```
import java.util.Iterator;
import java.util.PriorityQueue;


public class Priority_Queue
{

    public static void main(String args[]){
        PriorityQueue<String> queue=new PriorityQueue<String>();
        System.out.println("PROGRAM ON Priority Queue (Genric Approach)");
        queue.add("SUN");
        queue.add("MOON");
        queue.add("STAR");
        queue.add("EARTH");
        queue.add("JUPITOR");
        System.out.println(" Traversing elements using FOR EACH LOOP");
           for (String str : queue)
           {  System.out.println(str);  }
           System.out.println();System.out.println();
        System.out.println("head:"+queue.element());
        System.out.println("head:"+queue.peek());
           System.out.println();System.out.println();
        System.out.println("Traversing elements using  Iterator");
        Iterator itr=queue.iterator();
        while(itr.hasNext()){
        System.out.println(itr.next());
        }
        queue.remove();
        queue.poll();
        System.out.println();System.out.println();
        System.out.println("ON REMOVING 2 ELEMENTS");
        Iterator<String> itr2=queue.iterator();
        while(itr2.hasNext()){
        System.out.println(itr2.next());
        }
        }

    }
```