**LAB-5**

1. Write a Java program that reads a string from the user and uses StringTokenizer to split the string into individual words. Print each word on a new line.

Program-

**package** demo;

**import** java.util.StringTokenizer;

**public** **class** StringTokenizerDemo {

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

//create a StringTokenizer object to break the string into words

StringTokenizer st=**new** StringTokenizer("I am Karishma");

//Loop through and print each word on a new line

**while**(st.hasMoreTokens())

{

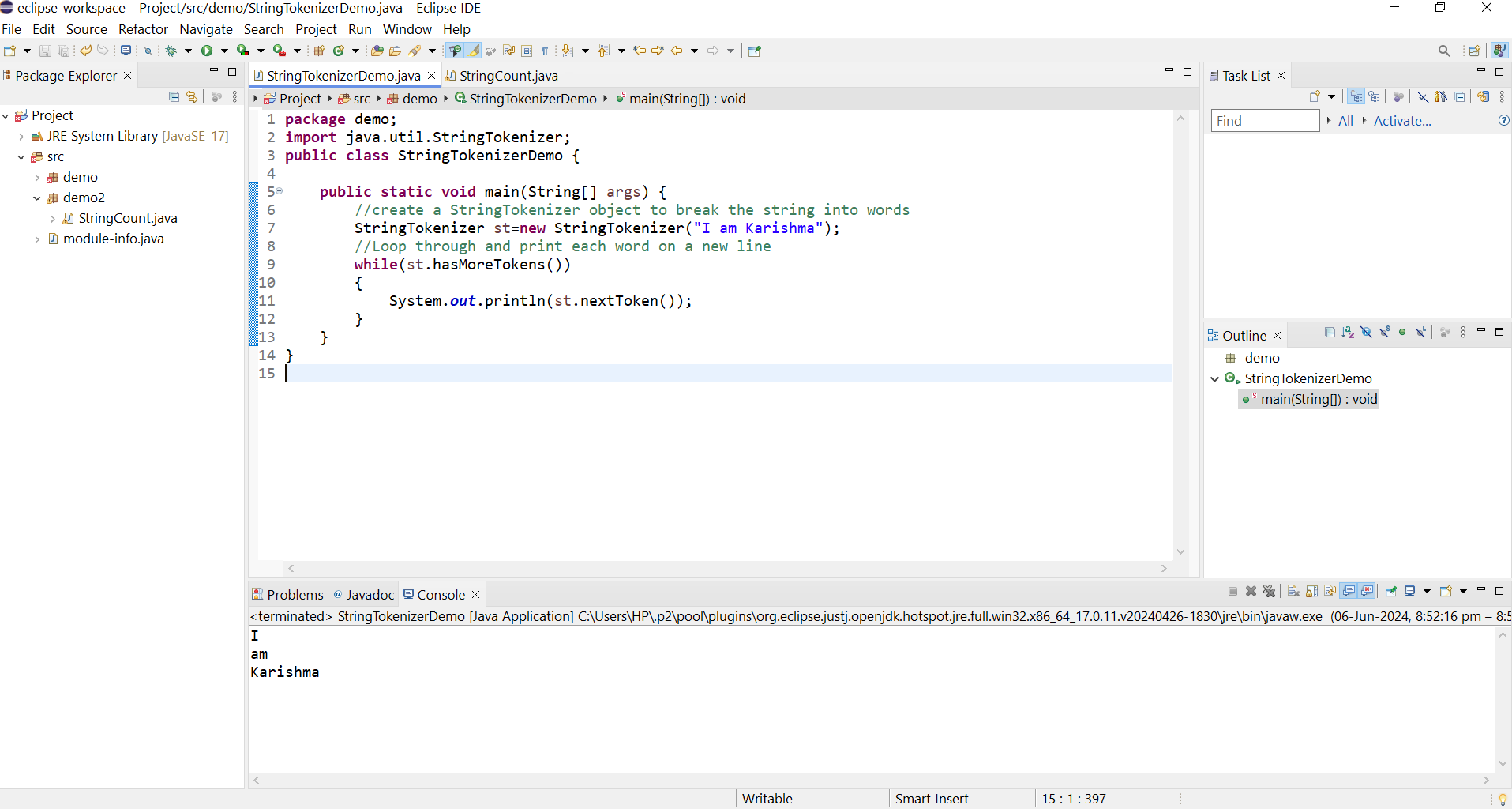
System.***out***.println(st.nextToken());

}

}

}

Output-



1. Write a Java program that reads a string from the user and uses StringTokenizer to count the number of words in the string.

Program-

**package** demo2;

**import** java.util.Scanner;

**import** java.util.StringTokenizer;

**public** **class** StringCount {

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

// Create a Scanner object to read input from the user

Scanner sc=**new** Scanner(System.***in***);

System.***out***.println("Enter a string : ");

//Read the entire line of input

String input=sc.nextLine();

//Create a StringTokenizer object to break the string into words

StringTokenizer tokenizer=**new** StringTokenizer(input);

//Count the number of words

**int** wordCount=tokenizer.countTokens();

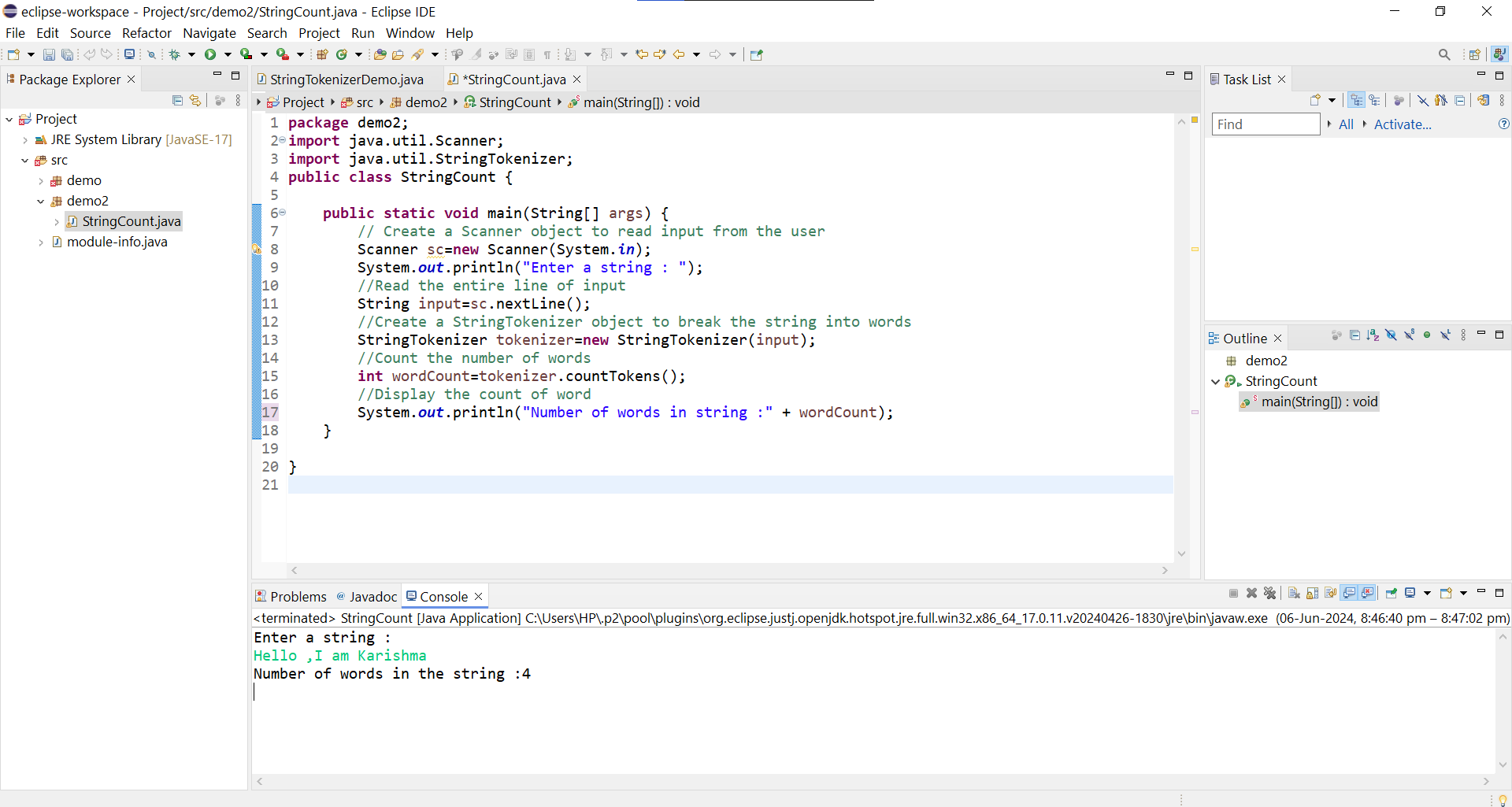
//Display the count of word

System.***out***.println("Number of words in string :" + wordCount);

}

}

Output-



1. Write a Java program to create a LinkedList of strings, add elements at specific positions (beginning, middle, end), and print the list.

Program-

**package** demo2;

**import** java.util.LinkedList;

**public** **class** LinkedListElements {

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

//Create a LinkedList of strings

LinkedList<String> list =**new** LinkedList<>();

//Add elements to LinkedList

//Add to the end of the list

list.add("End");

//Adds to the beginning of the list

list.addFirst("Beginning");

//Adds to the middle

list.add(1,"Middle");

//Prints the LinkedList

System.***out***.println("LinkedList elements:");

**for** (String element : list) {

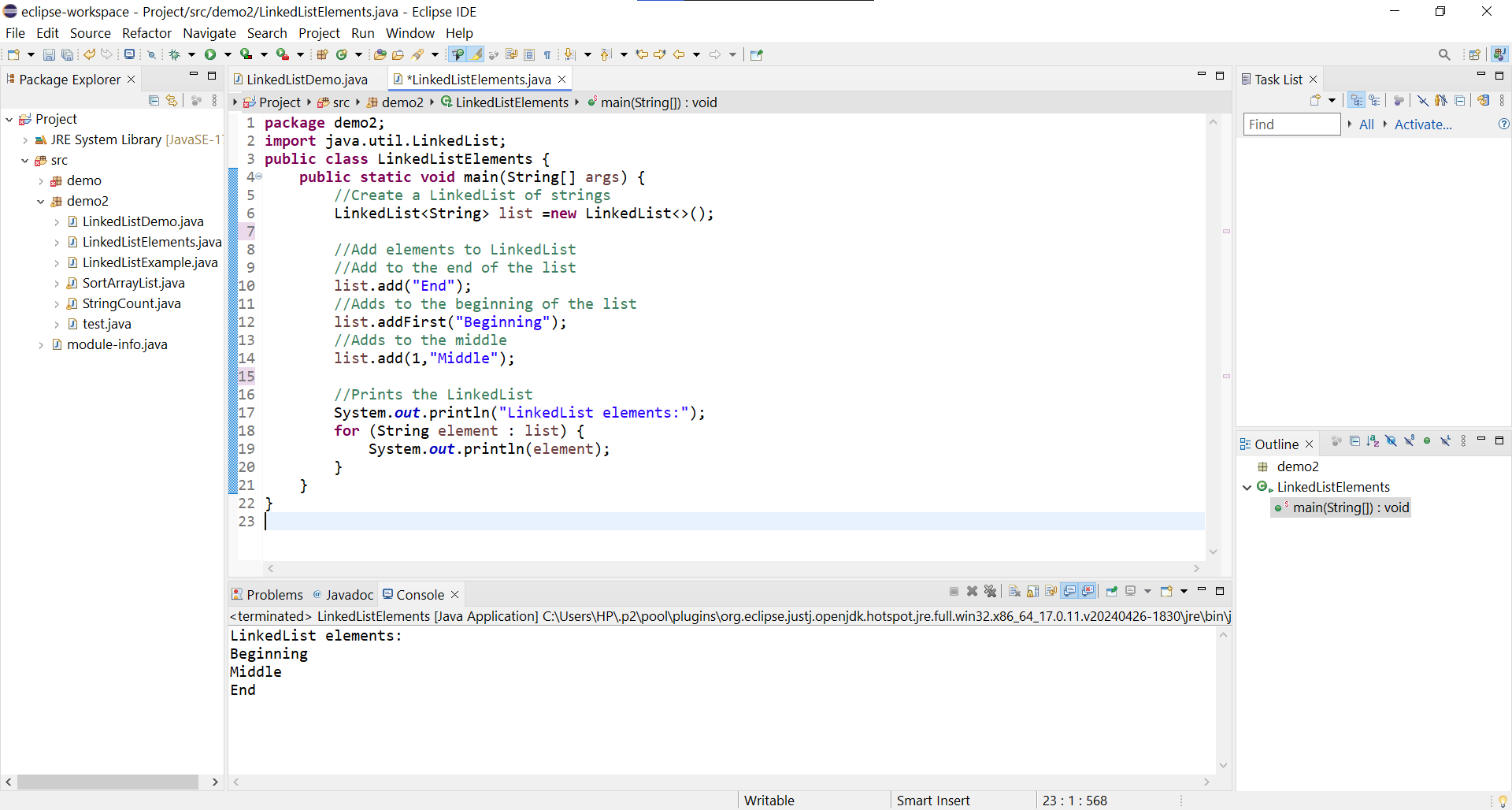
System.***out***.println(element);

}

}

}

Output-



1. Write a Java program to sort a given array list.

Program -

**package** demo2;

**import** java.util.Collections;

**import** java.util.ArrayList;

**public** **class** SortArrayList {

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

// Create an ArrayList of strings

ArrayList<String> list = **new** ArrayList<>();

// Add elements to the ArrayList

list.add("Rose");

list.add("Alex");

list.add("Jules");

list.add("Kate");

// Print the ArrayList before sorting

System.***out***.println("Before sorting: " + list);

// Sort the ArrayList

Collections.*sort*(list);

// Print the ArrayList after sorting

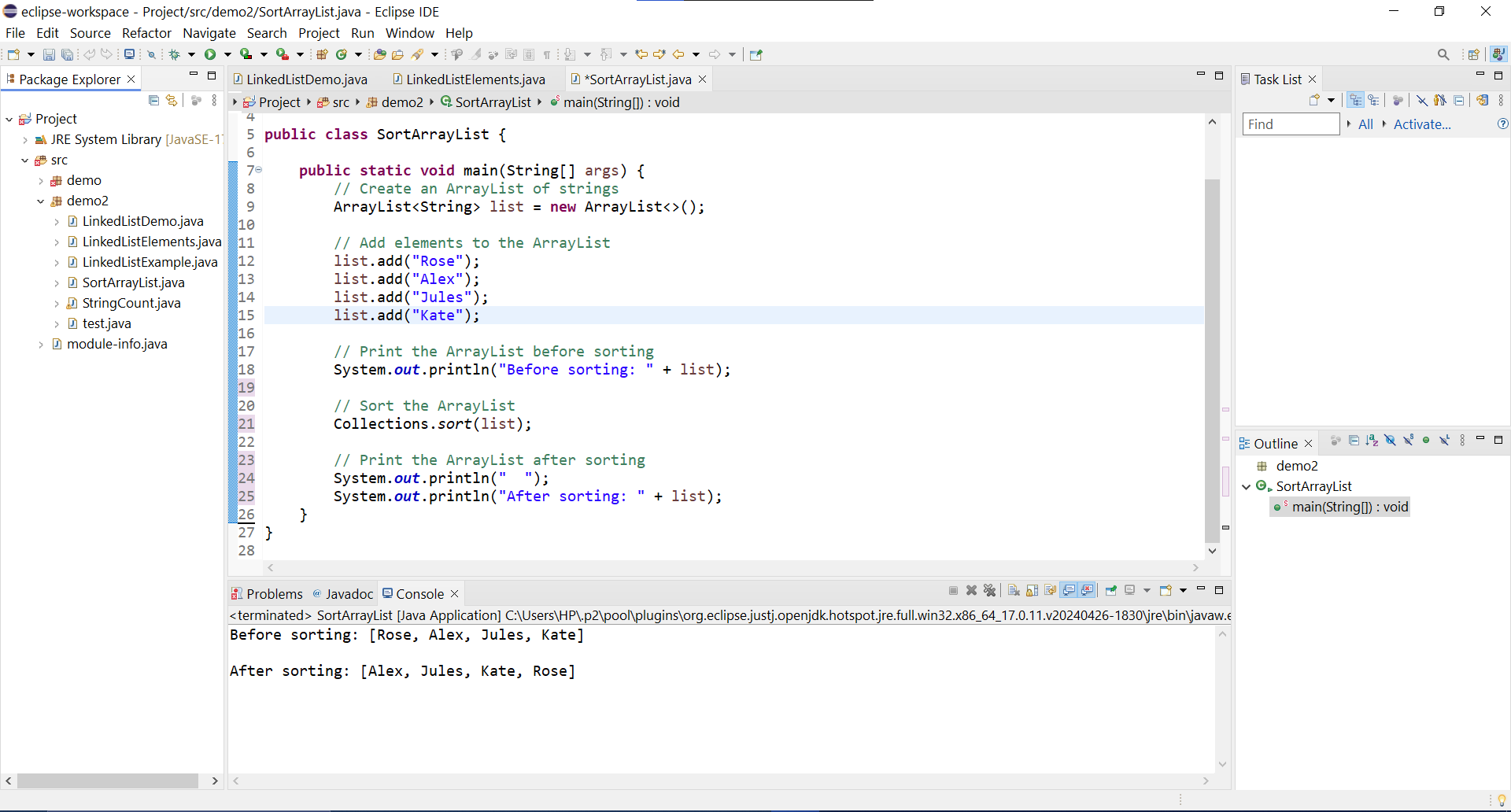
System.***out***.println(" ");

System.***out***.println("After sorting: " + list);

}

}

Output-



1. Write a Java program to replace the second element of an ArrayList with the specified element.

Program-

**package** demo2;

**import** java.util.ArrayList;

**public** **class** Replace {

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

// Create an ArrayList of strings

ArrayList<String> list = **new** ArrayList<>();

// Add elements to the ArrayList

list.add("First");

list.add("Second");

list.add("Third");

list.add("Fourth");

// Print the ArrayList before replacement

System.***out***.println("Before replacing the elements: " + list);

// Replace the second element with a specified element

String element= "Tenth";

**if** (list.size() > 1) {

list.set(1, element);

}

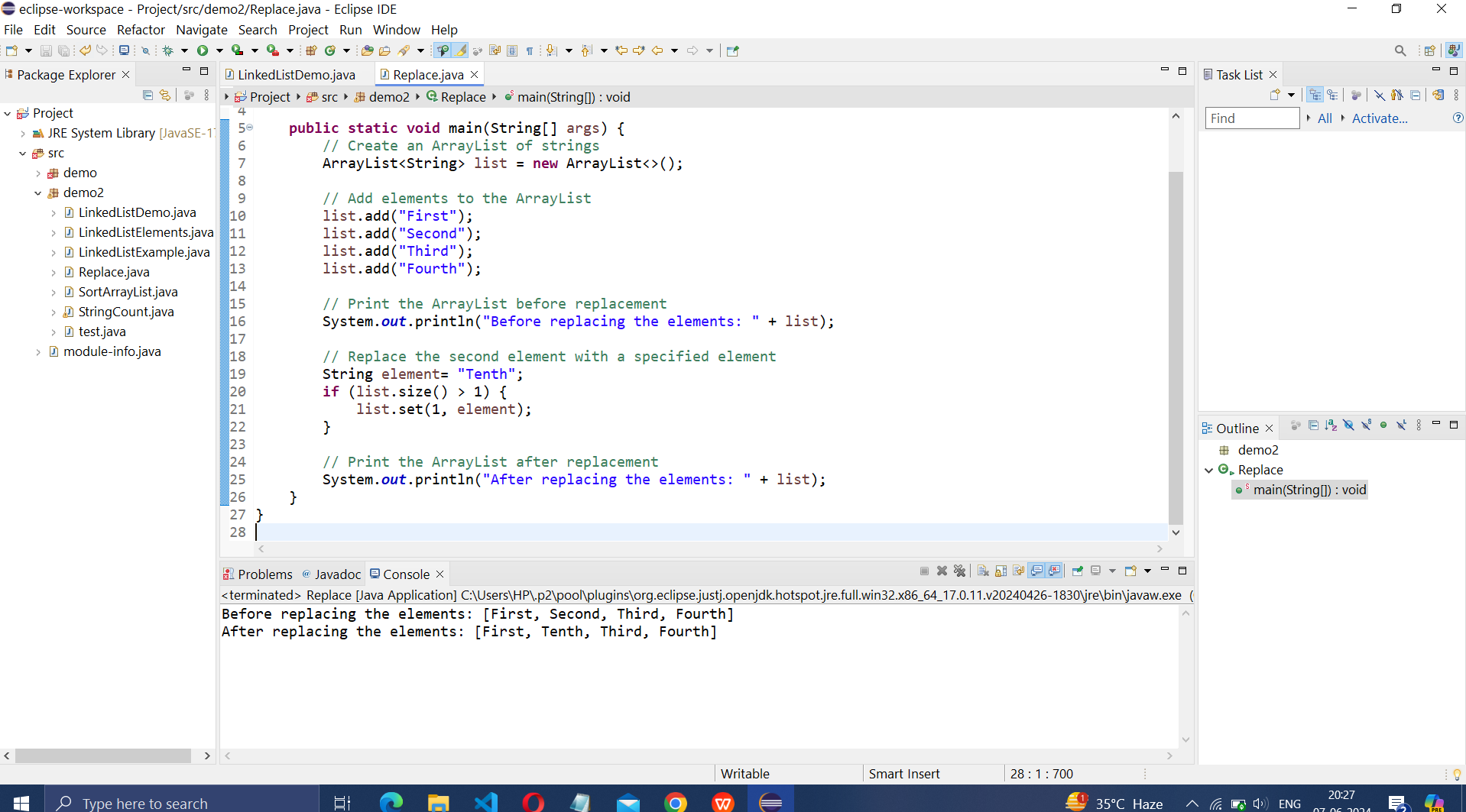
// Print the ArrayList after replacement

System.***out***.println("After replacing the elements: " + list);

}

}

Output-



1. Write a Java program to iterate a linked list in reverse order.

Program-

**package** demo2;

**import** java.util.LinkedList;

**import** java.util.Iterator;

**public** **class** Reverse {

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

// Create a LinkedList of strings

LinkedList<String> list = **new** LinkedList<>();

// Add elements to the LinkedList

list.add("One");

list.add("Two");

list.add("Three");

list.add("Four");

// Use the descendingIterator to iterate in reverse order

Iterator<String> iterator = list.descendingIterator();

// Iterate the LinkedList in reverse order

System.***out***.println("Linked List in reverse order:");

**while** (iterator.hasNext()) {

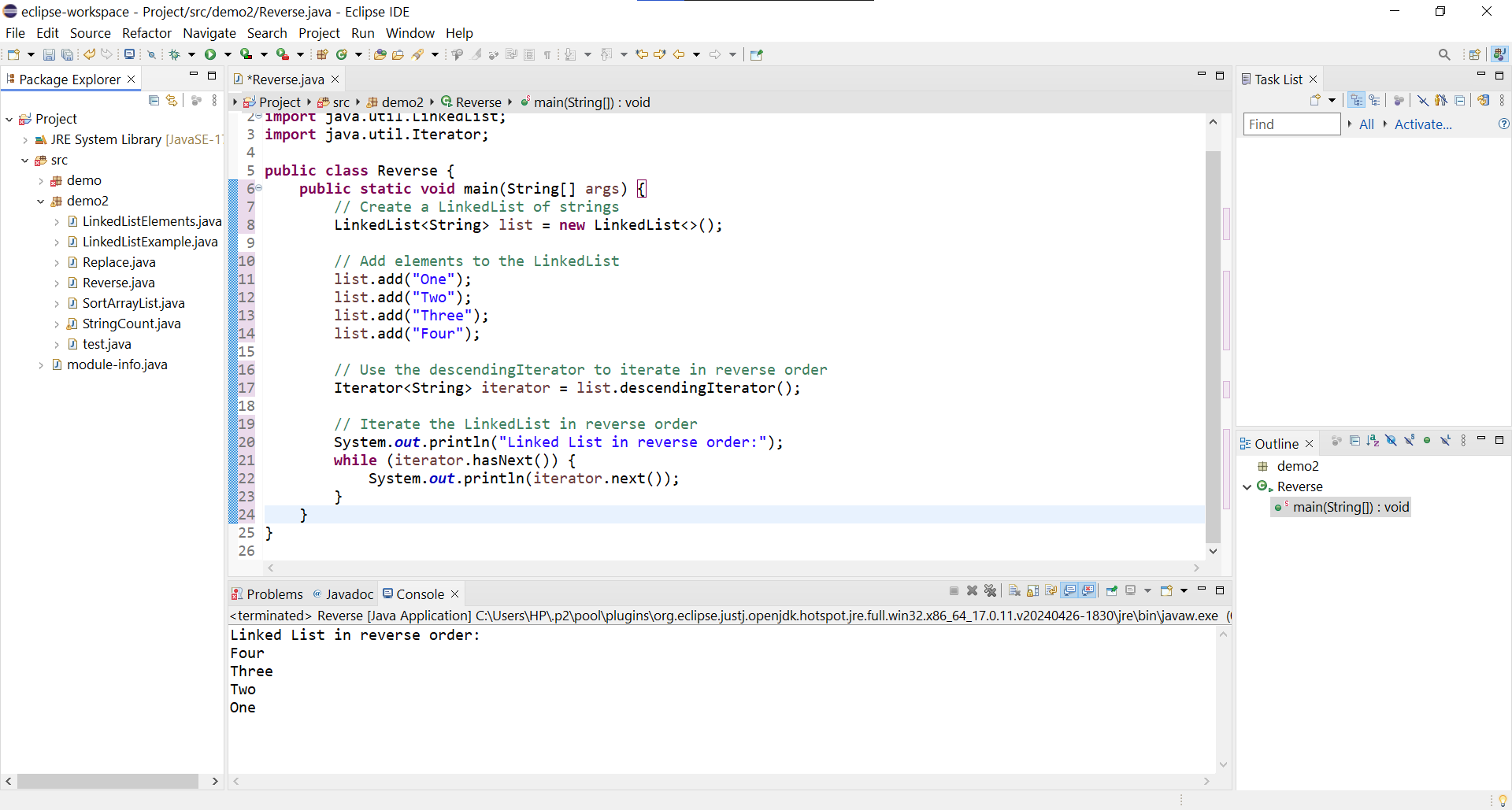
System.***out***.println(iterator.next());

}

}

}

Output-



1. Write a Java program to retrieve, but not remove, the last element of a linked list.

Program-

**package** demo2;

**import** java.util.LinkedList;

**public** **class** Retrieve {

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

//Create a LinkedList of strings

LinkedList<String> list = **new** LinkedList<>();

// Add elements to the LinkedList

list.add("First");

list.add("Second");

list.add("Third");

list.add("Fourth");

// Retrieve but do not remove the last element

String lastElement = list.getLast();

// Print the last element

System.***out***.println("Last element in list: " + lastElement);

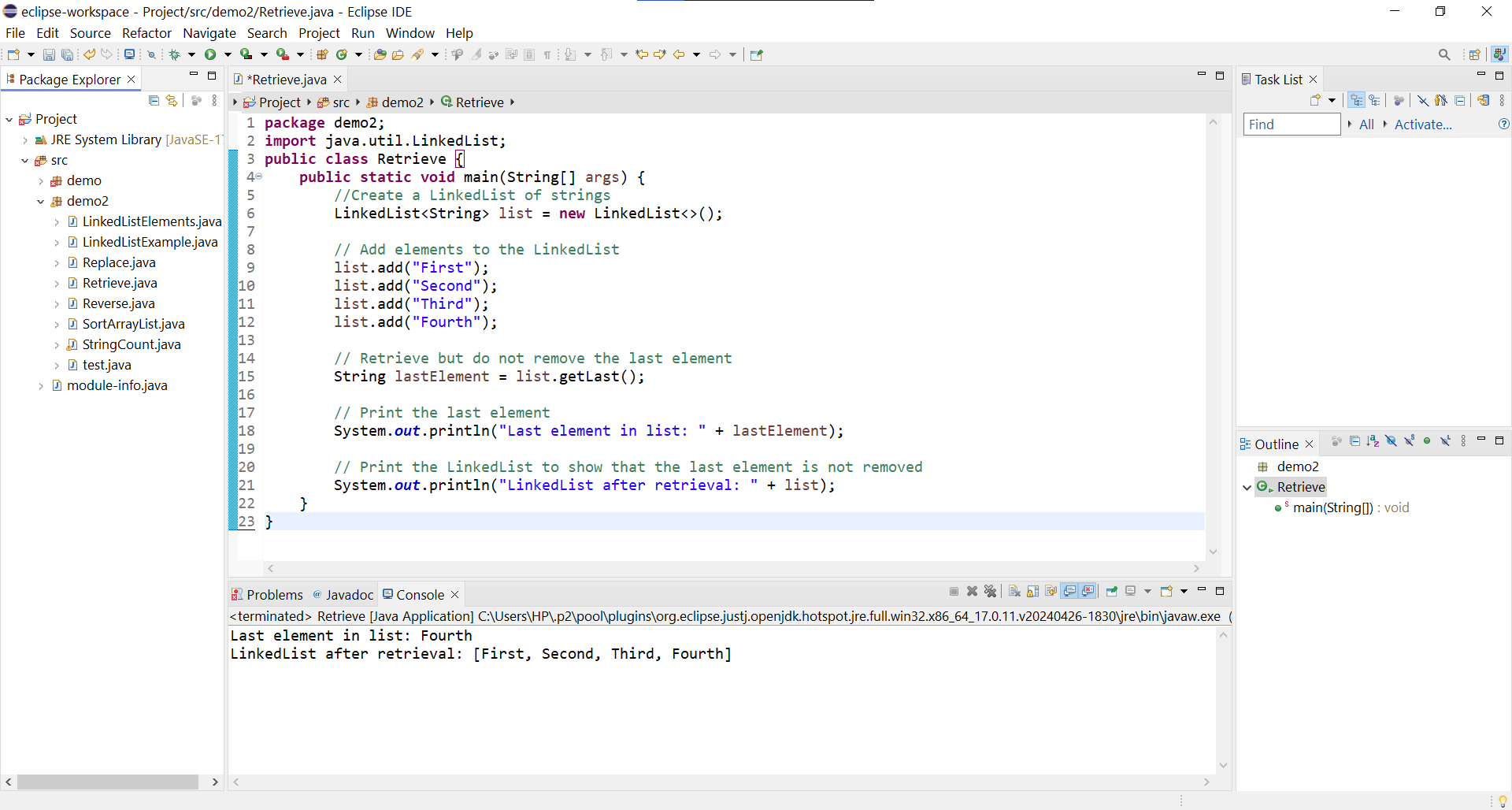
// Print the LinkedList to show that the last element is not removed

System.***out***.println("LinkedList after retrieval: " + list);

}

}

Output-



8. Write a Java program to create a LinkedList of integers and print all the elements.

Program-

**package** demo2;

**import** java.util.LinkedList;

**public** **class** LinkedListExample{

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

// Step 1: Create a LinkedList of integers

LinkedList<Integer> list = **new** LinkedList<>();

//Add elements to the LinkedList

list.add(10);

list.add(15);

list.add(20);

list.add(25);

list.add(30);

//Print all the elements of LinkedList

System.***out***.println("Elements of the LinkedList: ");

**for**(Integer element : list)

{

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

}

}

}

Output-

