**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.**

**Code:-**

import java.util.Scanner;

import java.util.StringTokenizer;

public class StringSplitExample {

public static void main(String[] args) {

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

Scanner scanner = new Scanner(System.in);

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

// Read the entire line of input

String input = scanner.nextLine();

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

StringTokenizer tokenizer = new StringTokenizer(input);

// Print each word on a new line

while (tokenizer.hasMoreTokens()) {

System.out.println(tokenizer.nextToken());

}

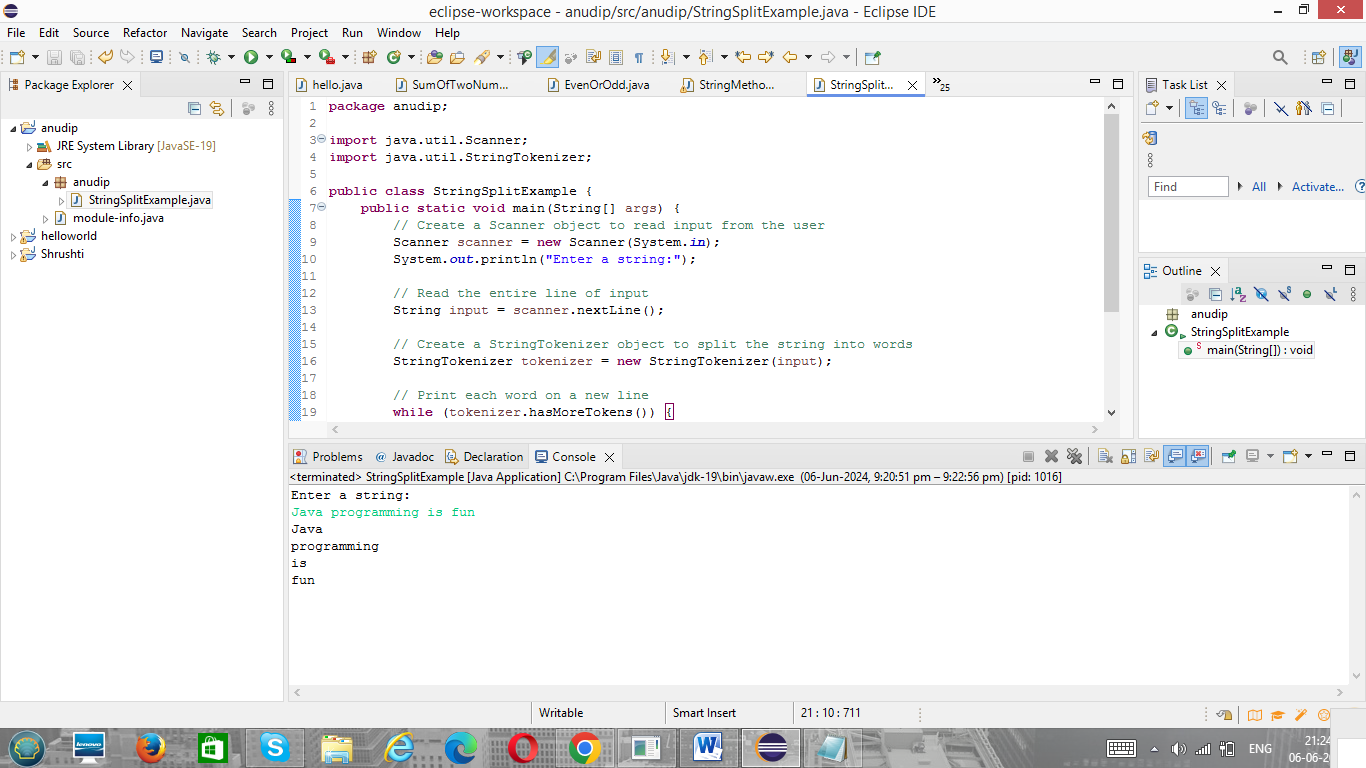
// Close the scanner

scanner.close();

}

}

**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.**

**Code:-**

import java.util.Scanner;

import java.util.StringTokenizer;

public class WordCounter {

public static void main(String[] args) {

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

Scanner scanner = new Scanner(System.in);

// Prompt the user to enter a string

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

String inputString = scanner.nextLine();

// Use StringTokenizer to split the string into words

StringTokenizer tokenizer = new StringTokenizer(inputString);

// Initialize a counter for the words

int wordCount = 0;

// Count each word

while (tokenizer.hasMoreTokens()) {

tokenizer.nextToken();

wordCount++;

}

// Print the number of words

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

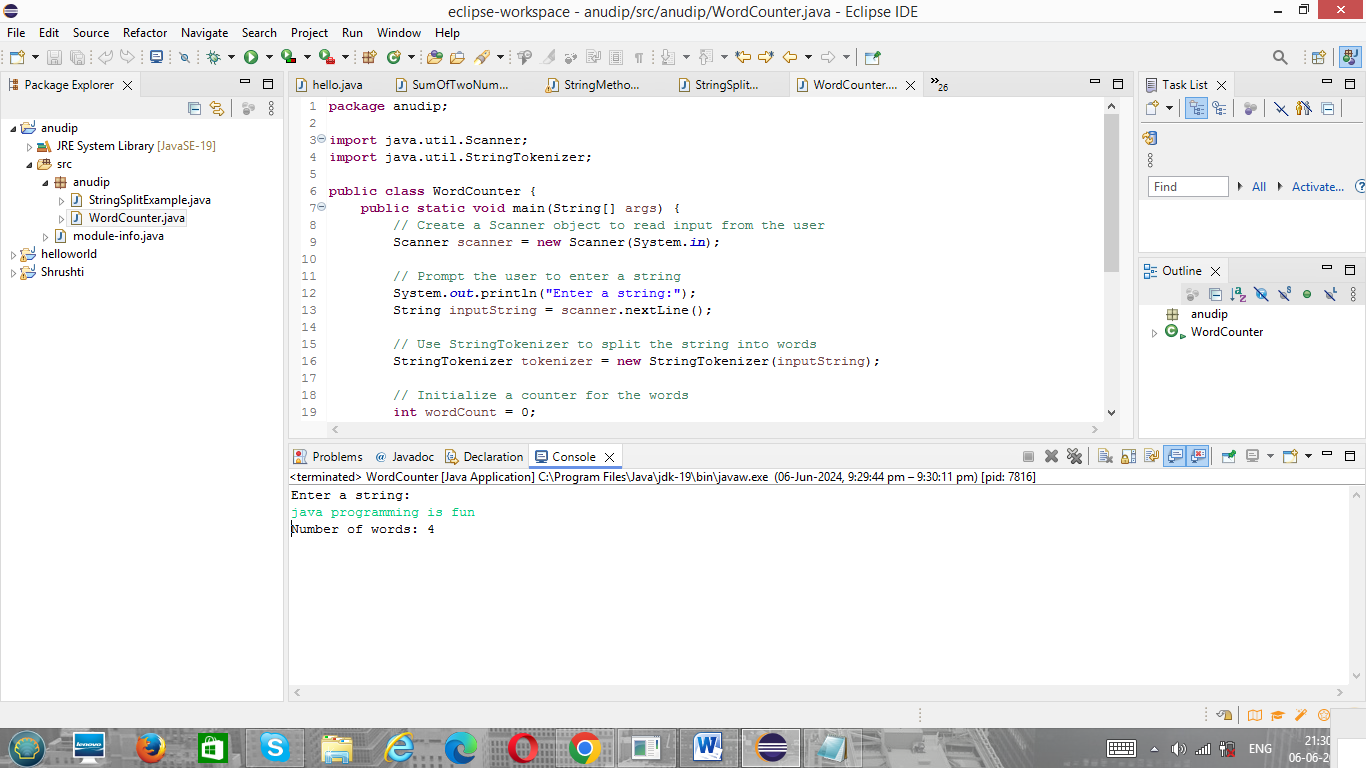
// Close the scanner

scanner.close();

}

}

**Output:**



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

**Code:-**

import java.util.LinkedList;

public class LinkedListExample {

public static void main(String[] args) {

// Create a LinkedList of strings

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

// Add elements at the beginning

list.addFirst("Apple");

// Add elements at the end

list.addLast("Orange");

// Add elements in the middle

list.add(1, "Mango");

// Print the list

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

for (String element : list) {

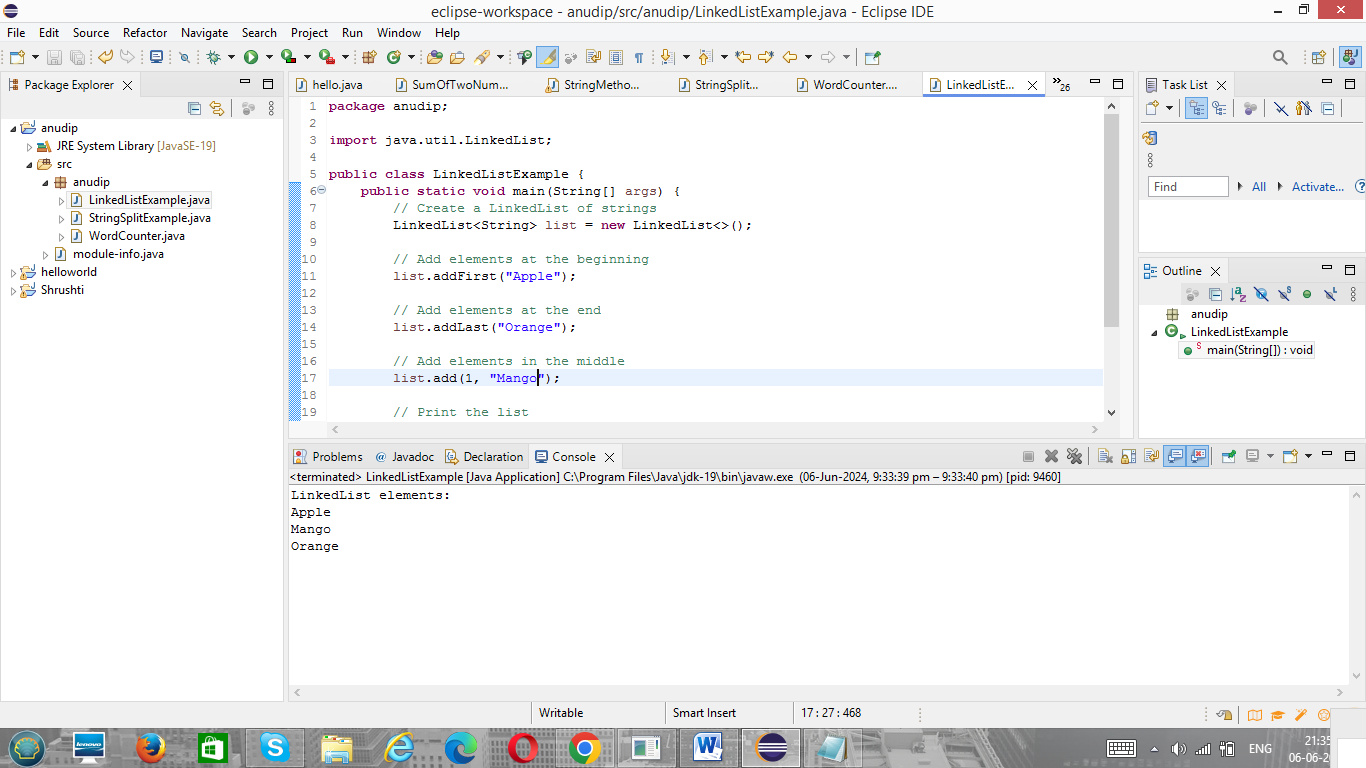
System.out.println(element);

}

}

}

**Output:-**



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

**Code:-**

import java.util.ArrayList;

import java.util.Collections;

public class ArrayListSortExample {

public static void main(String[] args) {

// Create an ArrayList of strings

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

// Add elements to the ArrayList

list.add("Banana");

list.add("Apple");

list.add("Cherry");

list.add("Mango");

list.add("Pineapple");

// Print the unsorted list

System.out.println("Unsorted ArrayList:");

for (String element : list) {

System.out.println(element);

}

// Sort the ArrayList

Collections.sort(list);

// Print the sorted list

System.out.println("\nSorted ArrayList:");

for (String element : list) {

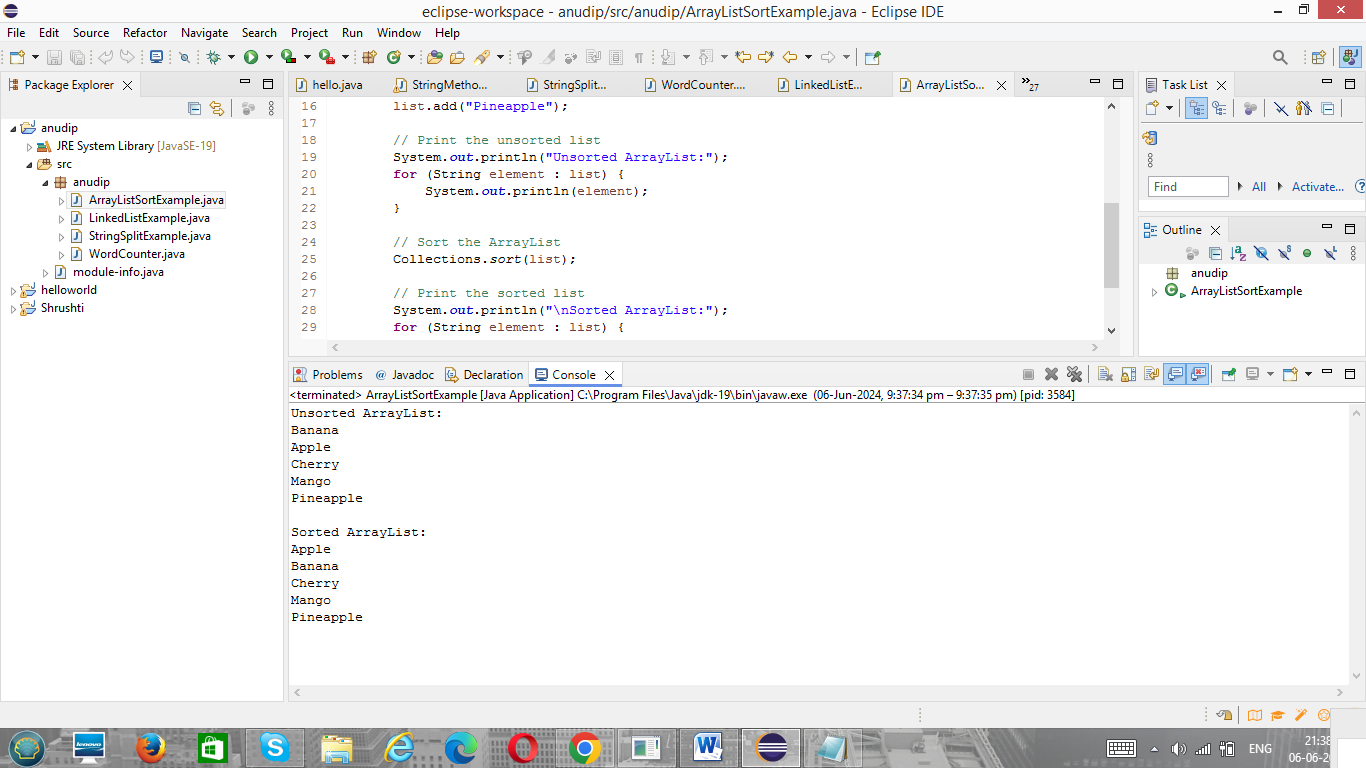
System.out.println(element);

}

}

}

**Output:-**



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

**Code:-**

import java.util.ArrayList;

public class ReplaceElementExample {

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 original list

System.out.println("Original ArrayList:");

for (String element : list) {

System.out.println(element);

}

// Replace the second element (index 1) with the specified element

String newElement = "NewSecond";

if (list.size() > 1) {

list.set(1, newElement);

} else {

System.out.println("The list does not have enough elements to replace the second one.");

}

// Print the modified list

System.out.println("\nModified ArrayList:");

for (String element : list) {

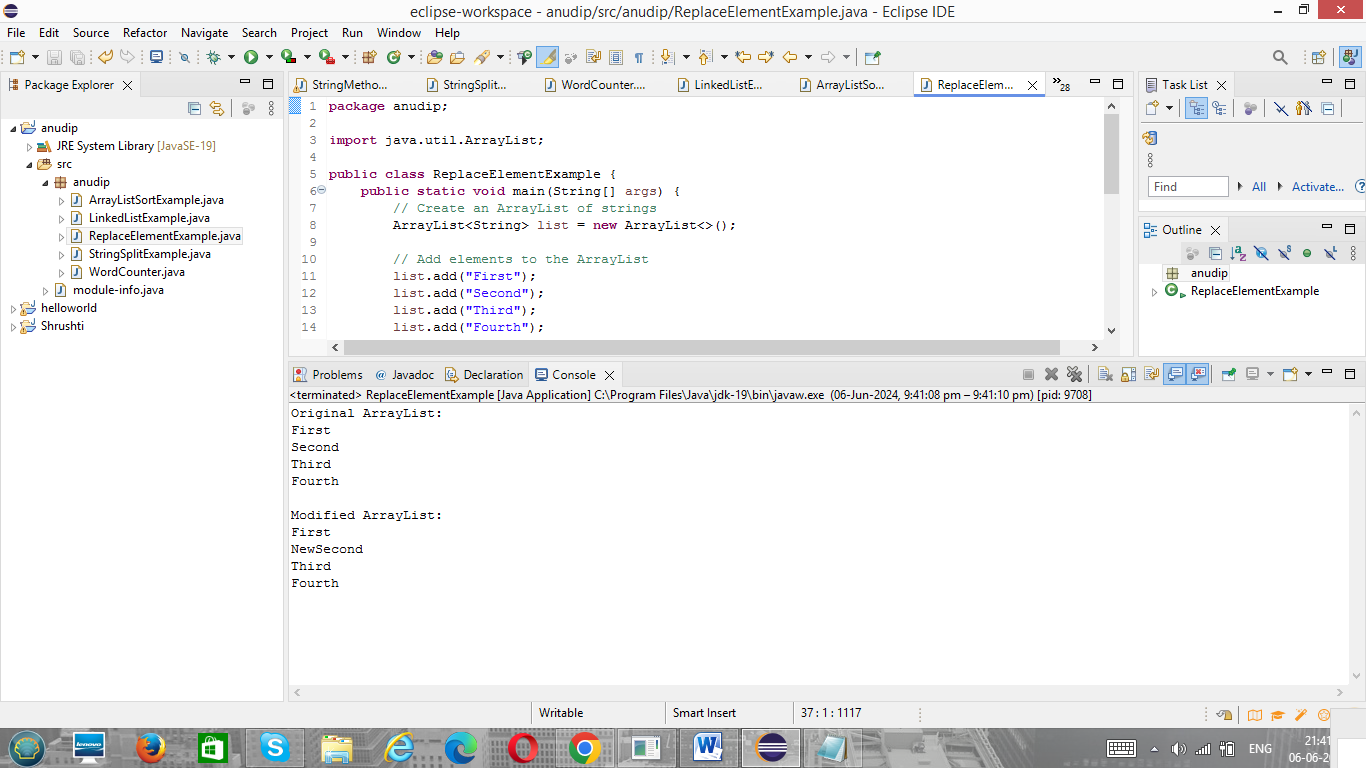
System.out.println(element);

}

}

}

**Output:-**



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

**Code:-**

import java.util.LinkedList;

import java.util.ListIterator;

public class ReverseLinkedListExample {

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");

// Print the original list

System.out.println("Original LinkedList:");

for (String element : list) {

System.out.println(element);

}

// Iterate the LinkedList in reverse order using a ListIterator

System.out.println("\nLinkedList in reverse order:");

ListIterator<String> iterator = list.listIterator(list.size());

while (iterator.hasPrevious()) {

String element = iterator.previous();

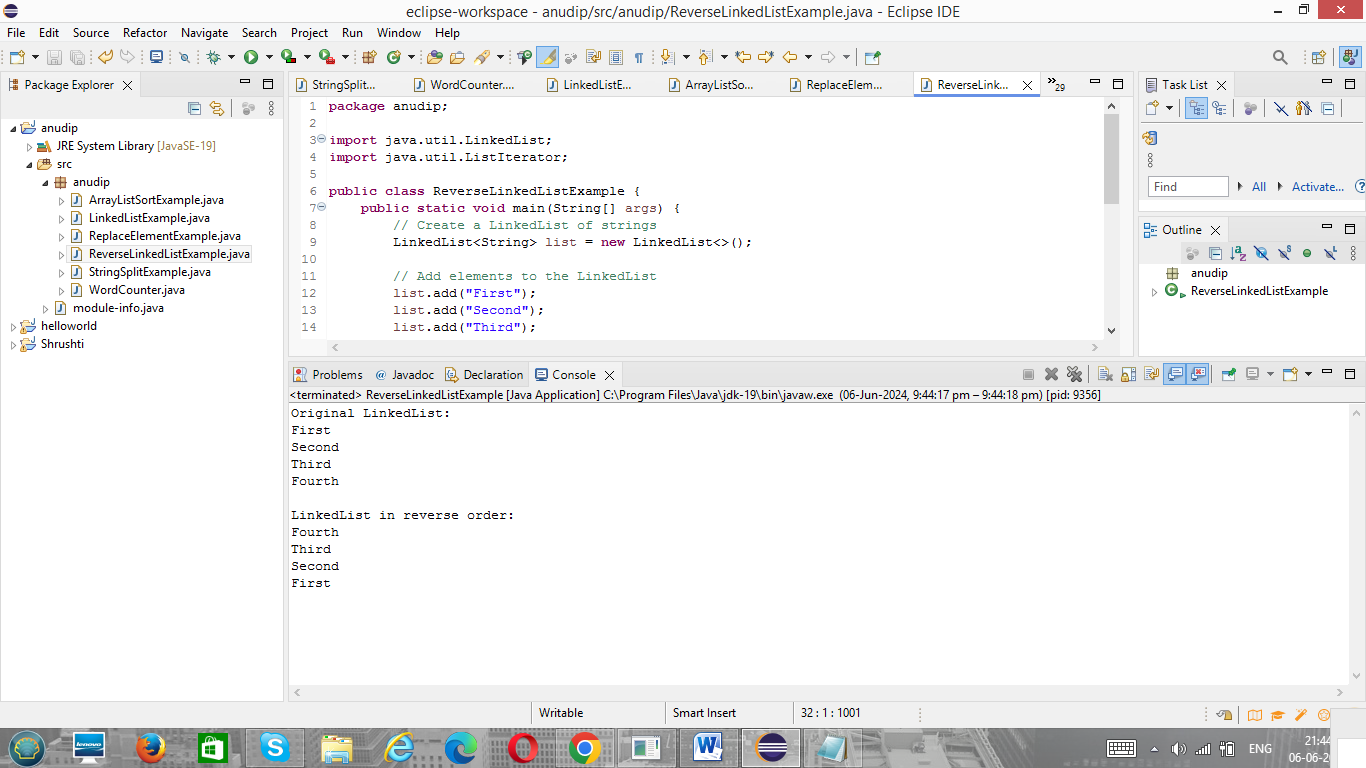
System.out.println(element);

}

}

}

**Output:-**



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

**Code:-**

import java.util.LinkedList;

public class RetrieveLastElementExample {

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");

// Print the original list

System.out.println("Original LinkedList:");

for (String element : list) {

System.out.println(element);

}

// Retrieve, but not remove, the last element

String lastElement = list.getLast();

// Print the last element

System.out.println("\nLast element (retrieved but not removed): " + lastElement);

// Print the list again to show it remains unchanged

System.out.println("\nLinkedList after retrieval:");

for (String element : list) {

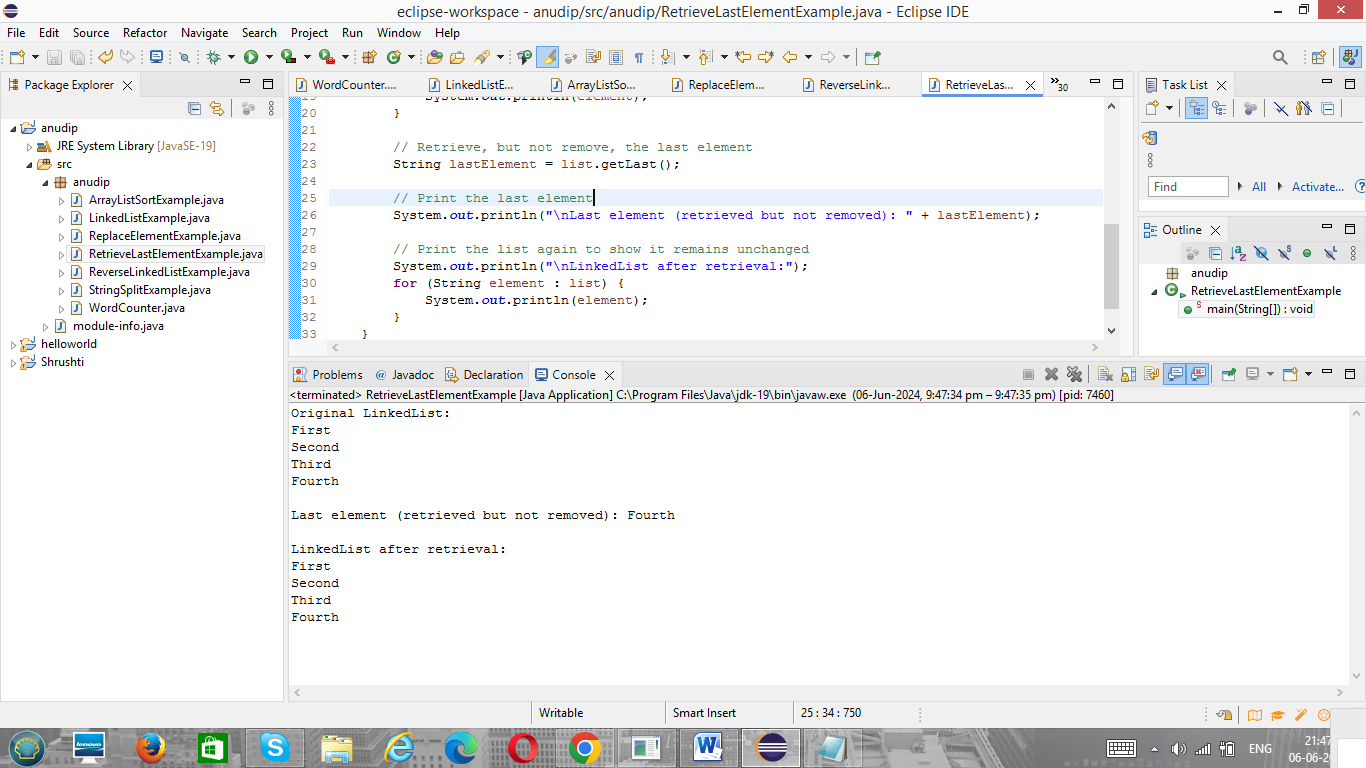
System.out.println(element);

}

}

}

**Output:-**



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

**Code:-**

import java.util.LinkedList;

public class LinkedListIntegerExample {

public static void main(String[] args) {

// Create a LinkedList of integers

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

// Add elements to the LinkedList

list.add(10);

list.add(20);

list.add(30);

list.add(40);

list.add(50);

// Print all the elements of the LinkedList

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

for (Integer element : list) {

System.out.println(element);

}

}

}

**Output:-**

