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.**

**package** lab5;

**import** java.util.Scanner;

**import** java.util.StringTokenizer;

**public** **class** StringSplitter {

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

// Create a Scanner object for reading input from the user

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

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

// Read the entire line of input

String input = s.nextLine();

// Create a StringTokenizer to split the string into words

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

**while** (tokenizer.hasMoreTokens()) {

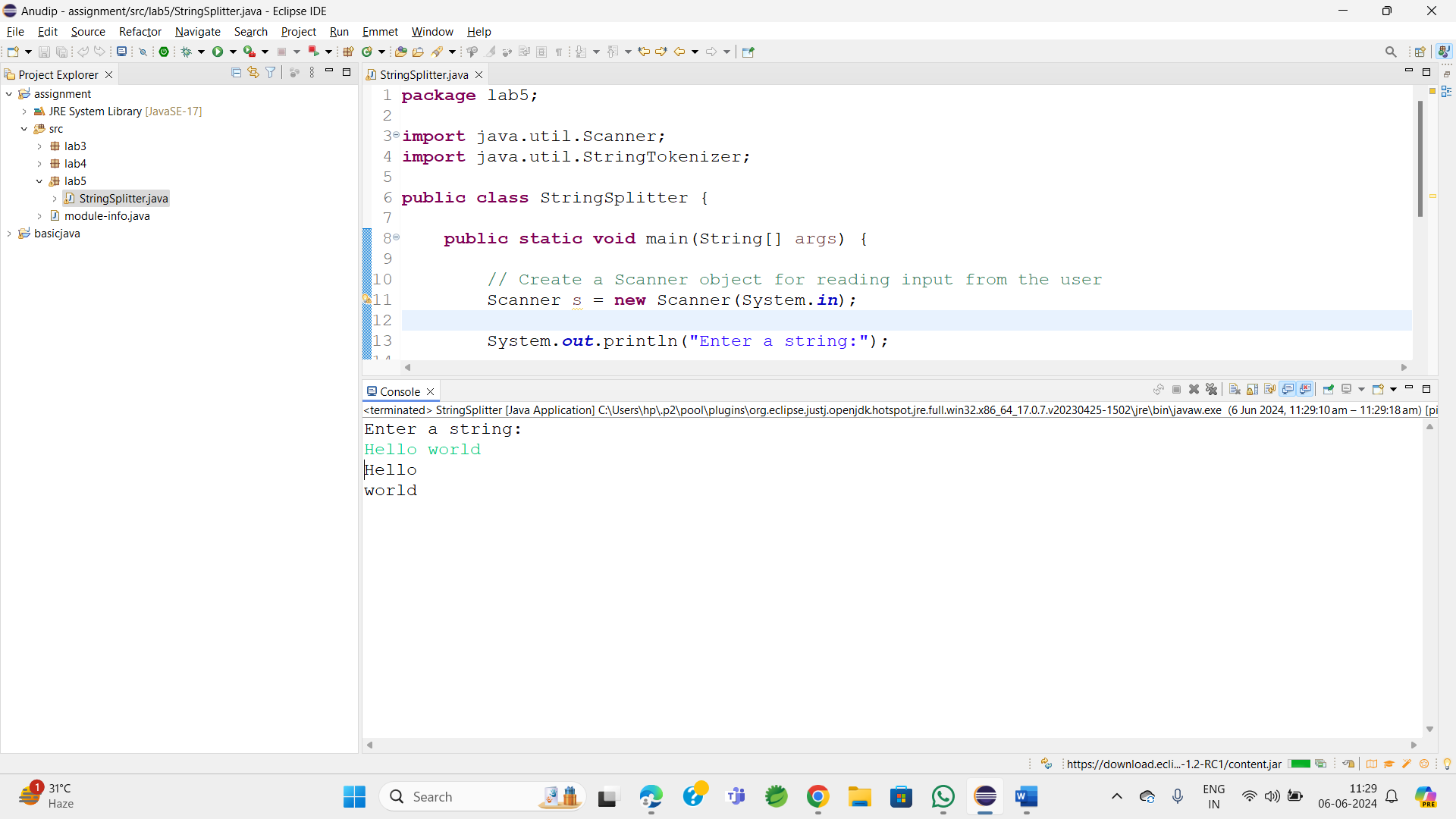
System.***out***.println(tokenizer.nextToken());

}

}

}

**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.**

package lab5;

import java.util.Scanner;

import java.util.StringTokenizer;

public class WordCounter {

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

Scanner s = new Scanner(System.in);

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

// Read the input string

String input = s.nextLine();

StringTokenizer t = new StringTokenizer(input);

// Count the number of tokens (words)

int wordCount = t.countTokens();

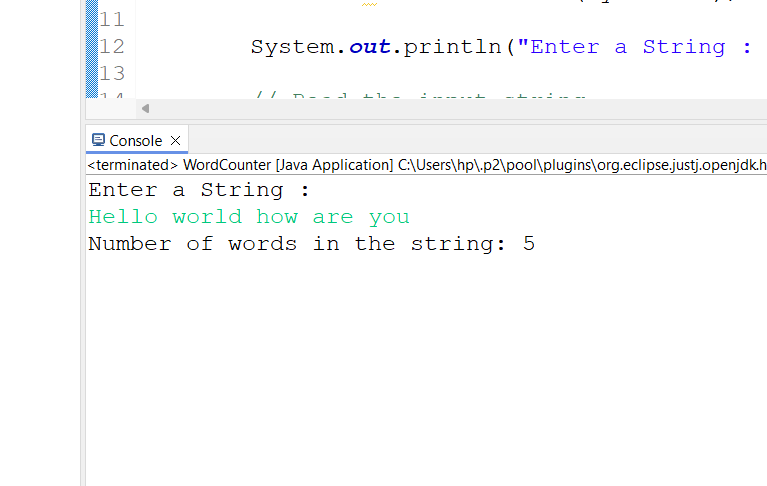
// Print the number of words

System.out.println("Number of words in the 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.**

**package** lab5;

**import** java.util.LinkedList;

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

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

// Create a LinkedList of strings

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

list.add("End");// add to the end

list.addFirst("Beginning");// add to the beginning

// Add an element to the middle

// Assuming "middle" is index 1 for a 3-element list

list.add(1, "Middle");

// Print the LinkedList

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

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

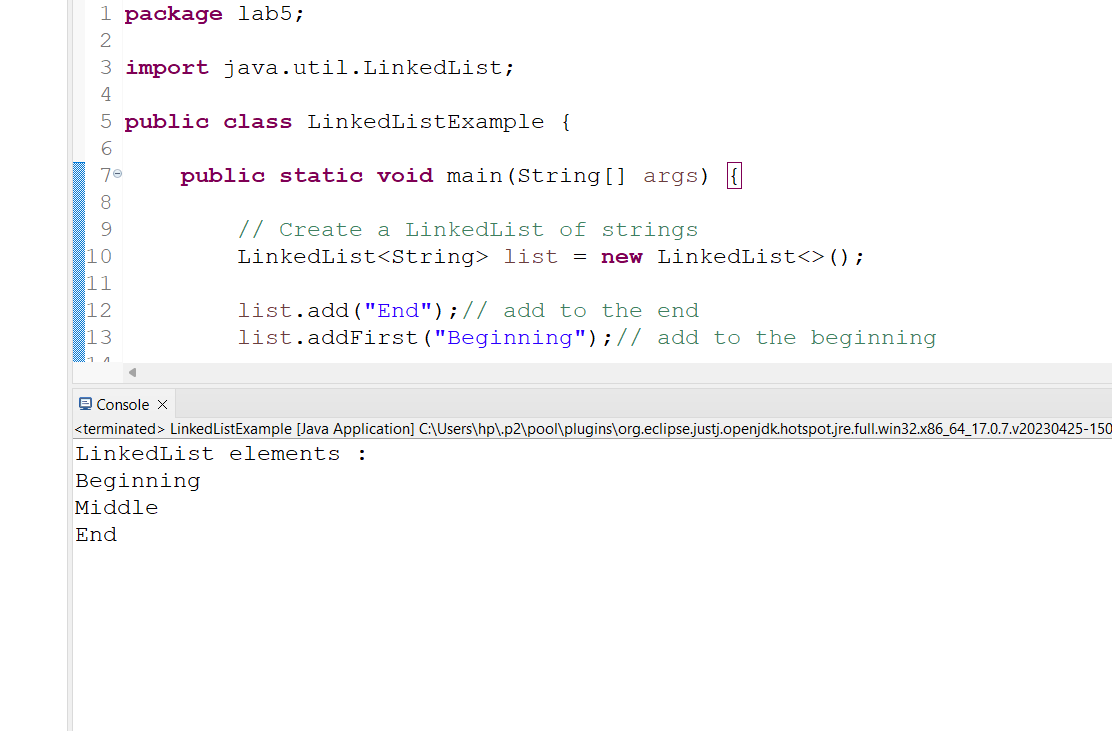
System.***out***.println(element);

}

}

}

**OutPut:**



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

**package** lab5;

**import** java.util.ArrayList;

**import** java.util.Collections;

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

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

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

// Add elements to the ArrayList

list.add("Banana");

list.add("Apple");

list.add("Mango");

list.add("Cherry");

list.add("Date");

// Print the ArrayList before sorting

System.***out***.println("ArrayList before Sorting : ");

**for**(String element :list )

{

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

}

// Sort the ArrayList

Collections.*sort*(list);

// Print the ArrayList after sorting

System.***out***.println("\nArrayList after sorting:");

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

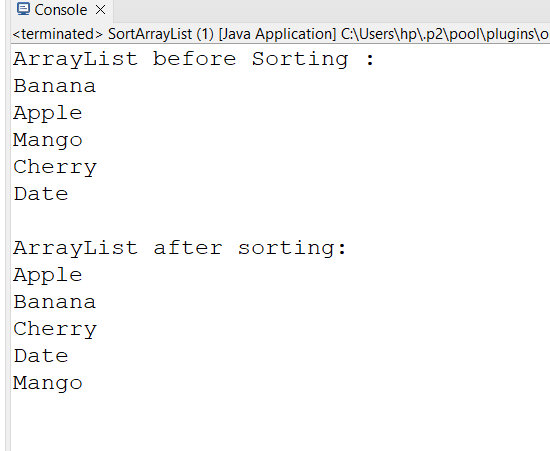
System.***out***.println(element);

}

}

}

**Output:**



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

**package** lab5;

**import** java.util.ArrayList;

**import** java.util.Iterator;

**public** **class** ReplaceElement {

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

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

// Add elements to the ArrayList

list.add("ONE");

list.add("TWO");

list.add("THREE");

list.add("FOUR");

list.add("FIVE");

System.***out***.println("ArrayList before replacement : ");

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

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

}

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

String newElement = "New Second";

list.set(1, newElement);

// Print the ArrayList after replacement

System.***out***.println("\nArrayList after replacement:");

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

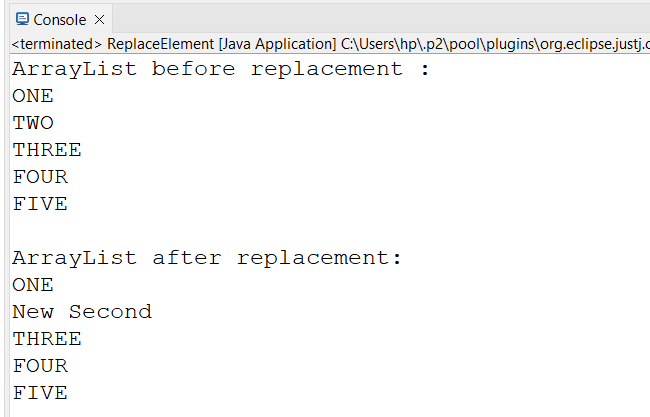
System.***out***.println(element);

}

}

}

**OutPut:**



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

**package** lab5;

**import** java.util.LinkedList;

**import** java.util.ListIterator;

**public** **class** ReverseLinkedList {

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

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

// Add elements to the LinkedList

list.add("ONE");

list.add("TWO");

list.add("THREE");

list.add("FOUR");

list.add("FIVE");

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

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

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

}

// Iterate the LinkedList in reverse order

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

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

**while** (iterator.hasPrevious()) {

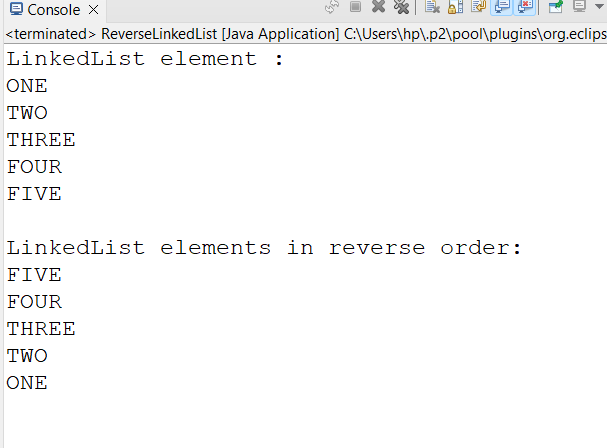
System.***out***.println(iterator.previous());

}

}

}

OutPut:



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

**package** lab5;

**import** java.util.LinkedList;

**public** **class** RetrieveLastElement {

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

// Create a LinkedList of strings

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

// Add elements to the LinkedList

list.add("Dog");

list.add("Rat");

list.add("Cat");

list.add("Bat");

// Retrieve, but do not remove, the last element

String lastElement = list.peekLast();

// Print the last element

System.***out***.println("The last element is: " + lastElement);

// Print the LinkedList to show it has not been modified

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

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

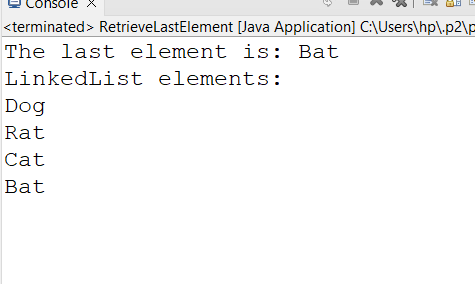
System.***out***.println(element);

}

}

}

**OutPut:**



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

**package** lab5;

**import** java.util.LinkedList;

**public** **class** LinkedListOfIntegers {

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

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 the LinkedList elements

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

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

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

}

}

}

**OutPut:**

