**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.**

package Session;

import java.util.Scanner;

import java.util.StringTokenizer;

public class JavaTokenizer {

public static void main(String[] args) {

// Scanner object to read input from the user

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

System.*out*.println("Enter the sentence:-");

String sentence= s.nextLine();

// StringTokenizer object to split the sentence

StringTokenizer tokenizer=new StringTokenizer(sentence);

System.*out*.println("Tokens are:-");

// Loop through all the tokens

while(tokenizer.hasMoreTokens())

{

String token= tokenizer.nextToken();

// Print the Token

System.*out*.println(token);

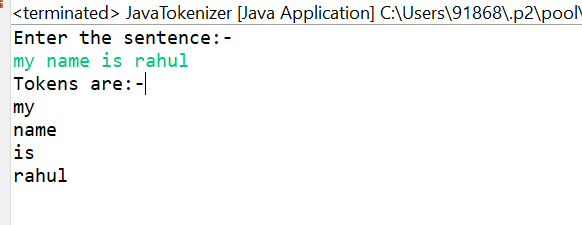
}

s.close();

}

}

**Output:-**

****

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

package Session;

import java.util.Scanner;

import java.util.StringTokenizer;

public class JavaTokenizer {

public static void main(String[] args) {

// Scanner object to read input from the user

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

System.*out*.println("Enter the sentence:-");

String sentence= s.nextLine();

// StringTokenizer object to split the sentence

StringTokenizer tokenizer=new StringTokenizer(sentence);

// Initial count=0

int count=0;

// Loop through all the tokens

while(tokenizer.hasMoreTokens())

{

String token= tokenizer.nextToken();

// Counting token through loop

count++;

}

// Print the count

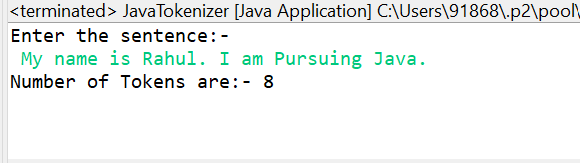
System.*out*.println("Number of Tokens are:- " + count);

s.close();

}

}

**Output:-**

****

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

package Session;

import java.util.LinkedList;

public class Linkedlistaddele {

public static void main(String[] args) {

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

// default given list

linkedList.add("Element 1");

linkedList.add("Element 3");

// Add an element to the beginning of the list

linkedList.addFirst("Element 0");

// Add an element to the middle of the list

linkedList.add(2, "Element 2");

// Add an element to the end of the list

linkedList.addLast("Element 4");

// Print the elements in the LinkedList

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

for (String element : linkedList) {

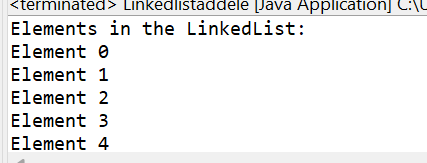
System.*out*.println(element);

}

}

}

**Output:-**

****

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

package Session;

import java.util.Collections;

import java.util.ArrayList;

public class ArrayListSorting {

public static void main(String[] args) {

// Create an ArrayList and add some integers

ArrayList<Integer> Sorting = new ArrayList<Integer>();

Sorting.add(3);

Sorting.add(5);

Sorting.add(6);

Sorting.add(8);

Sorting.add(2);

Sorting.add(4);

// Print the original list

System.*out*.println("Before Sorting "+Sorting);

// Sort the list

Collections.*sort*(Sorting);

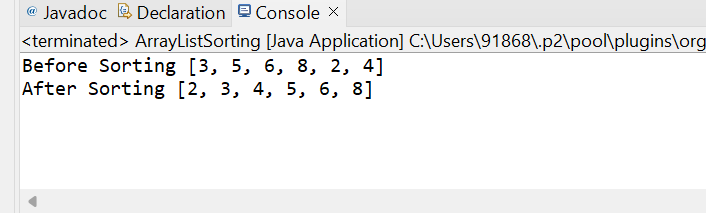
// Print the sorted list

System.*out*.println("After Sorting "+Sorting);

}

}

**Output:-**

****

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

**package** Session;

**import** java.util.Collections;

**import** java.util.ArrayList;

**public** **class** ArrayListSorting {

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

// Create an ArrayList and add some integers

ArrayList<Integer> Sorting = **new** ArrayList<Integer>();

Sorting.add(3);

Sorting.add(5);

Sorting.add(6);

Sorting.add(8);

Sorting.add(2);

Sorting.add(4);

// Print the original list

System.***out***.println("Before "+Sorting);

// Replacing the Second Element with 50

Sorting.set(1, 50);

// Sort the list

//Collections.sort(Sorting);

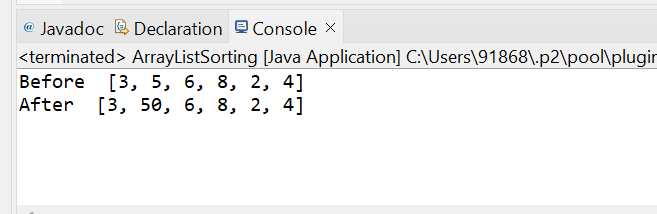
// Print the sorted list

System.***out***.println("After "+Sorting);

}

}

**Output:-**

****

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

package Session;

import java.util.LinkedList;

import java.util.Stack;

public class LinkedListReverse {

public static void main(String[] args) {

// Create a LinkedList

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

list.add("number 1");

list.add("number 2");

list.add("number 3");

list.add("number 4");

list.add("number 5");

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

// Use a Stack to reverse the LinkedList

Stack<String> stack = new Stack<String>();

// Push all elements

for(String e : list)

{

stack.push(e);

}

// Print the elements in reverse order by popping from the stack

System.*out*.println("After Reverse: ");

while (!stack.isEmpty())

{

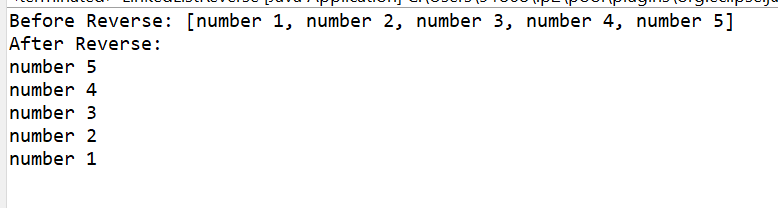
System.*out*.println(stack.pop());

}

}

}

**Output:-**

****

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

**package** Session;

**import** java.util.LinkedList;

**public** **class** LinkedListRetrive {

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

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

list.add("number 1");

list.add("number 2");

list.add("number 3");

list.add("number 4");

list.add("number 5");

System.***out***.println("Element in list Before Retrive " + list);

String lastelement = list.getLast();

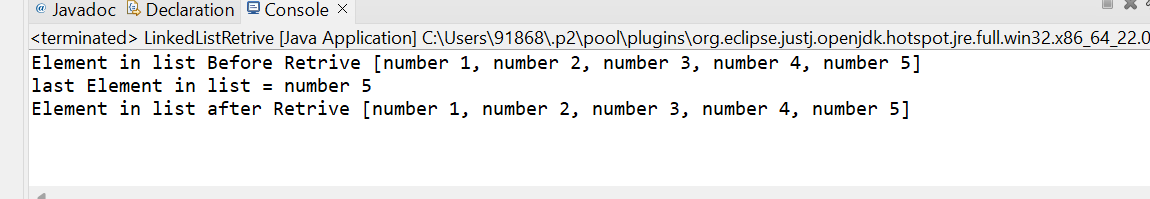
System.***out***.println("last Element in list = " + lastelement);

System.***out***.println("Element in list after Retrive " + list);

}

}

**Output:-**

****

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

package Session;

import java.util.LinkedList;

public class LinkedListRetrive {

public static void main(String[] args) {

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

list.add(1);

list.add(2);

list.add(3);

list.add(4);

list.add(5);

System.*out*.println("Element in list = " + list);

// Integer lastelement = list.getLast();

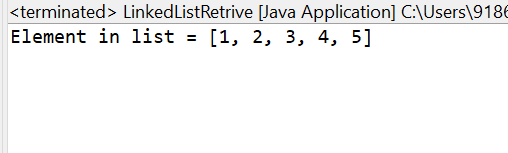
// System.out.println("last Element in list = " + lastelement);

//System.out.println("Element in list after Retrive " + list);

}

}

**Output:-**

****