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-

**package** mypackage;

**import** java.util.Scanner;

**import** java.util.StringTokenizer;

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

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

// Create a Scanner object for reading user input

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

// Prompt the user to enter a string

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

String input = scanner.nextLine();

// Create a StringTokenizer object to split the string

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();

}

}

Program-

A screenshot of a computer

Description automatically generated

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

Code-

**package** mypackage;

**import** java.util.Scanner;

**import** java.util.StringTokenizer;

**public** **class** WordCounter {

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

// Create a Scanner object for reading user input

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

// Prompt the user to enter a string

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

String input = scanner.nextLine();

// Create a StringTokenizer object to split the string

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

// Initialize a counter for the words

**int** wordCount = 0;

// Count the words

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

tokenizer.nextToken();

wordCount++;

}

// Print the word count

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

// Close the scanner

scanner.close();

}

}

Program-

A screenshot of a computer

Description automatically generated

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

Code-**package** mypackage;

**import** java.util.LinkedList;

**public** **class** LinkedListDemo {

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

// Create a LinkedList of strings

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

// Add elements to the LinkedList

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

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

list.add(1, "Middle"); // Add to the middle

// Print the LinkedList

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

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

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

}

}

}

Program-A screenshot of a computer

Description automatically generated

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

Code-

**package** mypackage;

**import** java.util.ArrayList;

**import** java.util.Collections;

**public** **class** ArrayListSort {

**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("Orange");

list.add("Mango");

list.add("Grapes");

// 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);

}

}

}

Program-

A screenshot of a computer

Description automatically generated

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

Code-

**package** mypackage;

**import** java.util.ArrayList;

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

**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("ArrayList before replacement:");

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

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

}

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

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

list.set(1, "NewElement");

} **else** {

System.***out***.println("The ArrayList does not have a second element to replace.");

}

// Print the ArrayList after replacement

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

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

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

}

}

}

Program-

A computer screen shot of a computer screen

Description automatically generated

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

Code-

**package** mypackage;

**import** java.util.LinkedList;

**import** java.util.ListIterator;

**public** **class** ReverseLinkedListIteration {

**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 LinkedList before iteration in reverse order

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

**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());

}

}

}

Program-

A screenshot of a computer

Description automatically generated

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

Code-

**package** mypackage;

**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("First");

list.add("Second");

list.add("Third");

list.add("Fourth");

// Print the LinkedList

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

**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 of the LinkedList: " + lastElement);

}

}

Program-

A computer screen shot of a computer screen

Description automatically generated

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

Code-

**package** mypackage;

**import** java.util.LinkedList;

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

**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);

}

}

}

Program-

A computer screen shot of a computer screen

Description automatically generated