**Lab5**

**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** lab5;

**import** java.util.Scanner;

**import** java.util.StringTokenizer;

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

**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 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

System.***out***.println("The words in the string are:");

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

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

}

// Close the scanner

scanner.close();

}

}

**Output**

A screen shot 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**

**Program**

**package** lab5;

**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 input = scanner.nextLine();

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

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

// Count the number of words

**int** wordCount = 0;

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

tokenizer.nextToken();

wordCount++;

}

// Print the number of words

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

// Close the scanner

scanner.close();

}

}

**Output**

**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.**

**Program**

**package** lab5;

**import** java.util.LinkedList;

**import** java.util.Scanner;

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

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

// Create a LinkedList of strings

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

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

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

// Prompt the user to enter an element to add at the beginning

System.***out***.println("Enter the element to add at the beginning: ");

String beginningElement = scanner.nextLine();

linkedList.addFirst(beginningElement);

// Prompt the user to enter an element to add at the end

System.***out***.println("Enter the element to add at the end: ");

String endElement = scanner.nextLine();

linkedList.addLast(endElement);

// Prompt the user to enter an element to add in the middle

System.***out***.println("Enter the element to add in the middle: ");

String middleElement = scanner.nextLine();

// Calculate the middle index

**int** middleIndex = linkedList.size() / 2;

linkedList.add(middleIndex, middleElement);

// Print the LinkedList

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

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

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

}

// Close the scanner

scanner.close();

}

}

**Output**

**A screenshot of a computer

Description automatically generated**

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

**Program**

**package** lab5;

**import** java.util.ArrayList;

**import** java.util.Collections;

**import** java.util.Scanner;

**public** **class** ArrayListSorter {

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

// Create an ArrayList of strings

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

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

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

// Prompt the user to enter the number of elements

System.***out***.println("Enter the number of elements: ");

**int** numberOfElements = scanner.nextInt();

scanner.nextLine(); // Consume the newline character

// Read the elements from the user

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

**for** (**int** i = 0; i < numberOfElements; i++) {

String element = scanner.nextLine();

arrayList.add(element);

}

// Sort the ArrayList

Collections.*sort*(arrayList);

// Print the sorted ArrayList

System.***out***.println("Sorted ArrayList elements:");

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

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

}

// Close the scanner

scanner.close();

}

}

**Output**

**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.**

**Program**

**package** lab5;

**import** java.util.ArrayList;

**import** java.util.Scanner;

**public** **class** ReplaceElementInArrayList {

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

// Create an ArrayList of strings

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

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

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

// Prompt the user to enter the number of elements

System.***out***.println("Enter the number of elements: ");

**int** numberOfElements = scanner.nextInt();

scanner.nextLine(); // Consume the newline character

// Read the elements from the user

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

**for** (**int** i = 0; i < numberOfElements; i++) {

String element = scanner.nextLine();

arrayList.add(element);

}

// Check if the list has at least 2 elements

**if** (arrayList.size() < 2) {

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

} **else** {

// Prompt the user to enter the new element

System.***out***.println("Enter the new element to replace the second element: ");

String newElement = scanner.nextLine();

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

arrayList.set(1, newElement);

// Print the modified ArrayList

System.***out***.println("Modified ArrayList elements:");

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

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

}

}

// Close the scanner

scanner.close();

}

}

**Output**

**A screenshot of a computer

Description automatically generated**

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

**Program**

**package** lab5;

**import** java.util.LinkedList;

**import** java.util.Iterator;

**import** java.util.Scanner;

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

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

// Create a LinkedList of strings

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

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

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

// Prompt the user to enter the number of elements

System.***out***.println("Enter the number of elements: ");

**int** numberOfElements = scanner.nextInt();

scanner.nextLine(); // Consume the newline character

// Read the elements from the user

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

**for** (**int** i = 0; i < numberOfElements; i++) {

String element = scanner.nextLine();

linkedList.add(element);

}

// Iterate the LinkedList in reverse order

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

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

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

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

}

// Close the scanner

scanner.close();

}

}

**Output**

**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**

**Program**

**package** lab5;

**import** java.util.LinkedList;

**import** java.util.Scanner;

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

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

// Create a LinkedList of strings

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

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

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

// Prompt the user to enter the number of elements

System.***out***.println("Enter the number of elements: ");

**int** numberOfElements = scanner.nextInt();

scanner.nextLine(); // Consume the newline character

// Read the elements from the user

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

**for** (**int** i = 0; i < numberOfElements; i++) {

String element = scanner.nextLine();

linkedList.add(element);

}

// Retrieve but do not remove the last element

**if** (!linkedList.isEmpty()) {

String lastElement = linkedList.getLast();

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

} **else** {

System.***out***.println("The linked list is empty.");

}

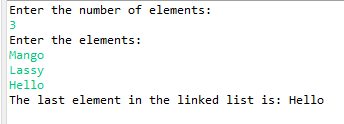
// Close the scanner

scanner.close();

}

}

**Output**

****

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

**Program**

**package** lab5;

**import** java.util.LinkedList;

**import** java.util.Scanner;

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

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

// Create a LinkedList of integers

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

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

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

// Prompt the user to enter the number of elements

System.***out***.println("Enter the number of elements: ");

**int** numberOfElements = scanner.nextInt();

// Read the elements from the user

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

**for** (**int** i = 0; i < numberOfElements; i++) {

**int** element = scanner.nextInt();

linkedList.add(element);

}

// Print all elements in the LinkedList

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

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

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

}

// Close the scanner

scanner.close();

}

}

**Output**

**A screenshot of a computer

Description automatically generated**