**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** hellow;

**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 inputString = scanner.nextLine();

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

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

// Loop through the tokens and print each word on a new line

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

String word = tokenizer.nextToken();

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

}

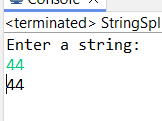
// Close the scanner

scanner.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.

Code:

**package** hellow;

**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();

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

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

// Initialize a counter to keep track of the number of words

**int** wordCount = 0;

// Loop through the tokens and count each word

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

tokenizer.nextToken();

wordCount++;

}

// Print the total number of words

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

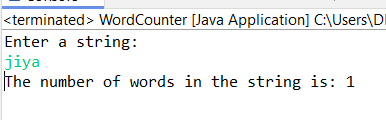
// Close the scanner

scanner.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.

Code:

**package** hellow;

**import** java.util.LinkedList;

**public** **class** LinkList {

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

// Create a LinkedList of strings

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

// Add elements to the LinkedList

linkedList.add("End"); // Adding element at the end

linkedList.addFirst("Starting"); // Adding element at the beginning

// Adding an element in the middle

// Assuming we want to add at index 1 (between "Beginning" and "End")

linkedList.add(1, "Middle");

// Print the LinkedList

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

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

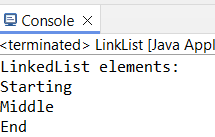
System.***out***.println(element);

}

}

}

Output:



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

Code:

**package** hellow;

**import** java.util.ArrayList;

**import** java.util.Collections;

**public** **class** Arraylist {

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

// Create an ArrayList of strings

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

// Add elements to the ArrayList

arrayList.add("vk");

arrayList.add("virat");

arrayList.add("karan");

arrayList.add("chahat");

arrayList.add("angel");

// Print the ArrayList before sorting

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

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

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

}

// Sort the ArrayList

Collections.*sort*(arrayList);

// Print the ArrayList after sorting

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

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

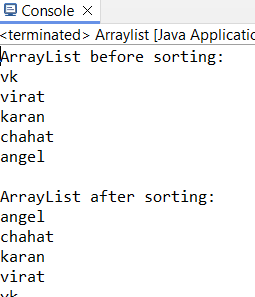
System.***out***.println(element);

}

}

}

Output:



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

Code:

**package** hellow;

**import** java.util.ArrayList;

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

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

// Create an ArrayList of strings

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

// Add elements to the ArrayList

arrayList.add("First");

arrayList.add("Second");

arrayList.add("Third");

arrayList.add("Fourth");

// Print the ArrayList before replacement

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

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

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

}

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

String newElement = "Replaced";

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

arrayList.set(1, newElement);

}

// Print the ArrayList after replacement

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

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

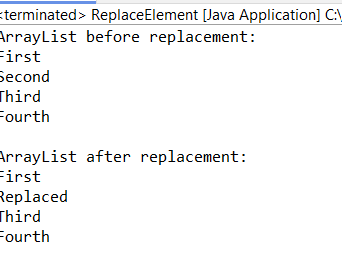
System.***out***.println(element);

}

}

}

Output:



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

Code:

**package** hellow;

**import** java.util.LinkedList;

**import** java.util.ListIterator;

**public** **class** Reverse {

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

// Create a LinkedList of strings

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

// Add elements to the LinkedList

linkedList.add("First");

linkedList.add("Second");

linkedList.add("Third");

linkedList.add("Fourth");

// Print the LinkedList before reverse iteration

System.***out***.println("LinkedList before reverse iteration:");

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

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

}

// Iterate over the LinkedList in reverse order

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

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

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

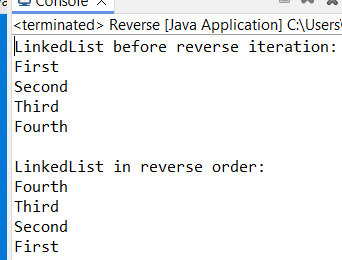
System.***out***.println(iterator.previous());

}

}

}

Output:



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

Code:

**package** hellow;

**import** java.util.LinkedList;

**public** **class** RetriveLastElement {

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

// Create a LinkedList of strings

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

// Add elements to the LinkedList

linkedList.add("First");

linkedList.add("Second");

linkedList.add("Third");

linkedList.add("Fourth");

// Print the LinkedList

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

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

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

}

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

String lastElement = linkedList.peekLast();

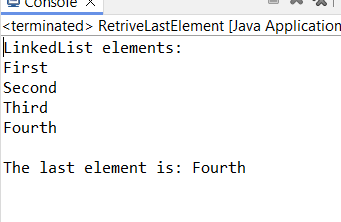
// Print the last element

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

}

}

Output:



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

Code:

**package** hellow;

**import** java.util.LinkedList;

**public** **class** Link {

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

// Create a LinkedList of integers

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

// Add elements to the LinkedList

linkedList.add(1);

linkedList.add(2);

linkedList.add(3);

linkedList.add(4);

linkedList.add(5);

// Print all the elements of the LinkedList

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

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

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

}

}

}

Output:

