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** Demo;

**import** java.util.Scanner;

**import** java.util.StringTokenizer;

**public** **class** StringTokenizerExample {

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

// **TODO** Auto-generated method stub

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 to split the input string into words

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

// Print each word on a new line

System.***out***.println("Words in the input string:");

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

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

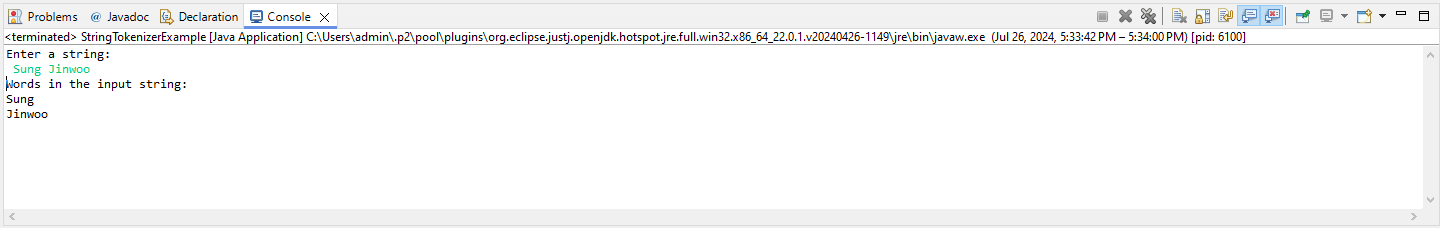
}

// Close the scanner

scanner.close();

}

}

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.

Program:-

**package** Demo;

**import** java.util.Scanner;

**import** java.util.StringTokenizer;

**public** **class** WordCountExample {

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

// **TODO** Auto-generated method stub

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 to split the input string into words

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

// Count the number of words

**int** wordCount = tokenizer.countTokens();

// Print the number of words

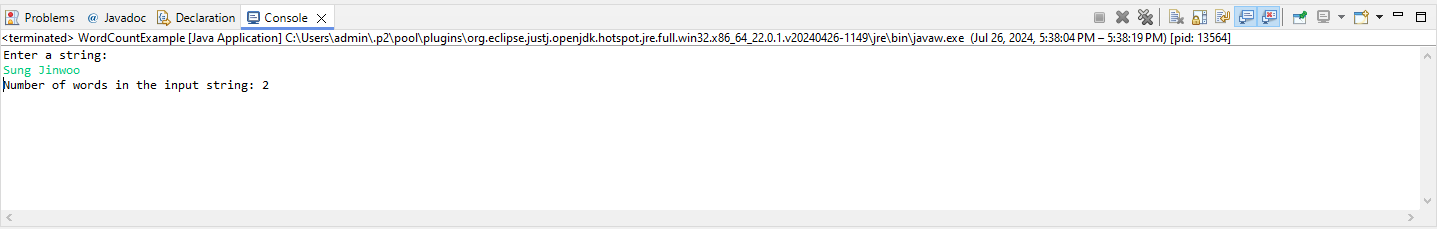
System.***out***.println("Number of words in the input string: " + wordCount);

// Close the scanner

scanner.close();

}

}

Output:-

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

Program:-

**package** Demo;

**import** java.util.LinkedList;

**public** **class** LinkListExample {

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

// **TODO** Auto-generated method stub

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

// Add elements to the list

list.add("Element at end"); // Adding to the end initially

// Add an element at the beginning

list.addFirst("Element at beginning");

// Add an element at the middle

list.add(1, "Element at middle"); // The list size is 2 now, so index 1 is the middle

// Add another element to the end

list.add("Another element at end");

// Print the LinkedList

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

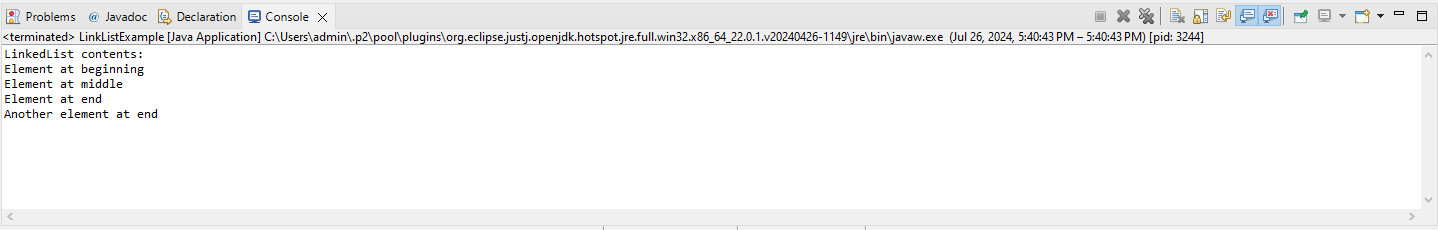
**for** (String element : list) {

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

}

}

}

Output:-

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

Program:-

**package** Demo;

**import** java.util.ArrayList;

**import** java.util.Collections;

**public** **class** ArrayListSortExample {

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

// **TODO** Auto-generated method stub

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

// Add elements to the ArrayList

list.add(5);

list.add(2);

list.add(9);

list.add(1);

list.add(6);

// Print the original list

System.***out***.println("Original ArrayList:");

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

System.***out***.print(num + " ");

}

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

// Sort the ArrayList

Collections.*sort*(list);

// Print the sorted list

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

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

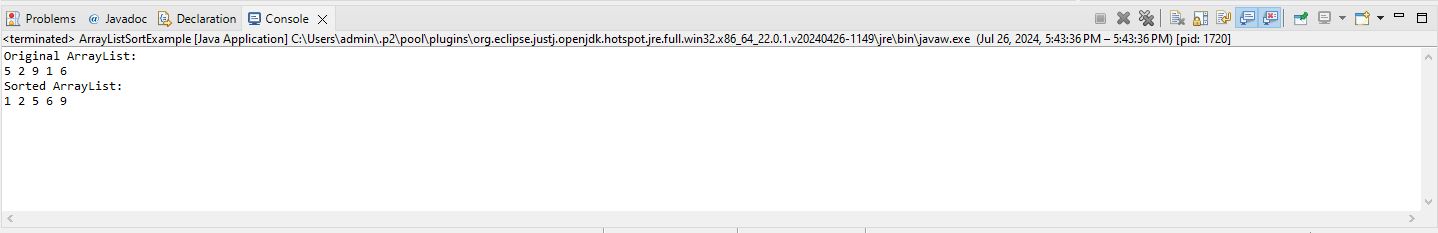
System.***out***.print(num + " ");

}

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

}

}

Output:-

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

Program:-

**package** Demo;

**import** java.util.ArrayList;

**public** **class** ReplaceElementExample {

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

// **TODO** Auto-generated method stub

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

list.add("Apple");

list.add("Banana");

list.add("Cherry");

list.add("Date");

// Print the original ArrayList

System.***out***.println("Original ArrayList:");

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

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

}

// Define the new element to replace the second element

String newElement = "Blueberry";

// Check if the ArrayList has at least two elements

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

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

list.set(1, newElement);

} **else** {

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

}

// Print the ArrayList after replacement

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

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

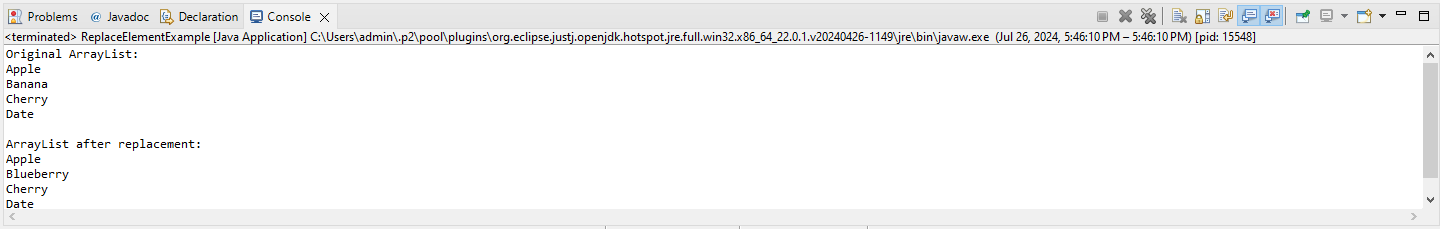
System.***out***.println(element);

}

}

}

Output:-



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

Program:-

**package** Demo;

**import** java.util.LinkedList;

**import** java.util.ListIterator;

**public** **class** LinkedListReverseIterationExample {

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

// **TODO** Auto-generated method stub

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

list.add("Continental");

list.add("BMW");

list.add("Dugati");

list.add("kawasaki");

// Print the original LinkedList

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

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

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

}

// Iterate the LinkedList in reverse order using ListIterator

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

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

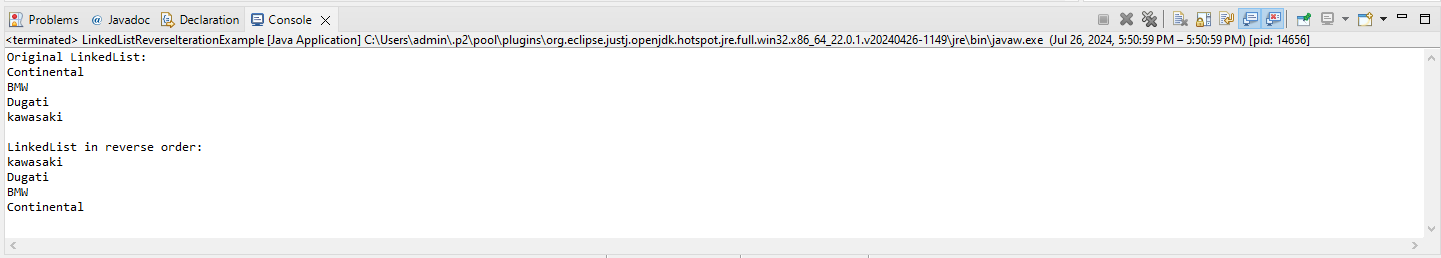
**while** (iterator.hasPrevious()) {

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

}

}

}

Output:-

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

Program:-

**package** Demo;

**import** java.util.LinkedList;

**public** **class** ReverseLastElementExample {

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

// **TODO** Auto-generated method stub

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

list.add("Dodge Charger");

list.add("Mustang");

list.add("GTR");

list.add("Supra");

// Print the original LinkedList

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

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

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

}

// Retrieve but do not remove the last element

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

String lastElement = list.getLast();

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

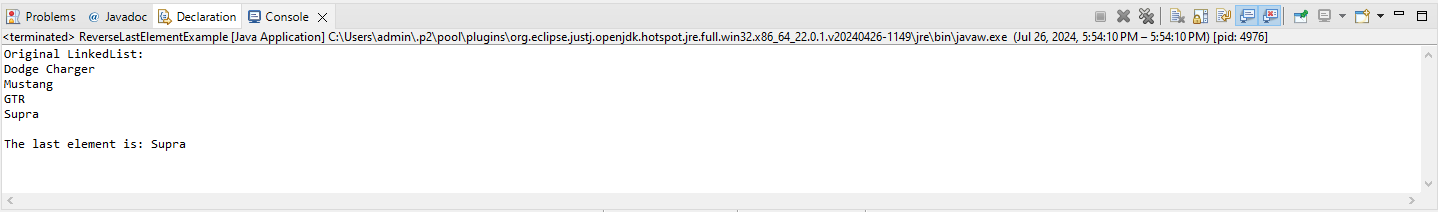
} **else** {

System.***out***.println("\nThe LinkedList is empty.");

}

}

}

Output:-

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

Program:-

**package** Demo;

**import** java.util.LinkedList;

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

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

// **TODO** Auto-generated method stub

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

// Add some integers to the LinkedList

list.add(10);

list.add(20);

list.add(30);

list.add(40);

list.add(50);

// Print all elements of the LinkedList

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

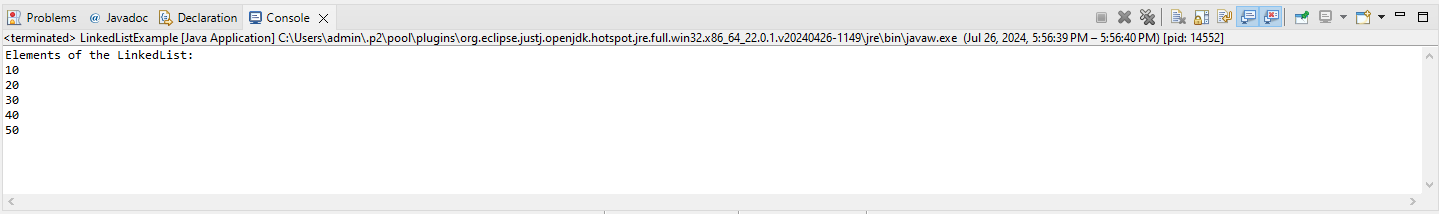
**for** (Integer element : list) {

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

}

}

}

Output:-