Arrays

-----------

**What is an array**

**—-------------------------**

**Introduction**

**An array is an indexed collection of fixed number of homogeneous data elements.**

**The main advantage of arrays is we can represent multiple values with the same name so that readability of the code will be improved.  
  
 But the main disadvantage of arrays is:  
 Fixed in size that is once we created an array there is no chance of increasing or decreasing the size based on our requirement that is to use arrays concept compulsory we should know the size in advance which may not possible always.**

**We can resolve this problem by using collections.**

**Array declarations:**

**Single dimensional array declaration:**

**Example:**

**int[] a;//recommended to use because name is clearly separated from the type**

**int []a;**

**int a[];**

**At the time of declaration we can't specify the size otherwise we will get compile time error.**

**Example:**

**int[] a;//valid**

**int[5] a;//invalid**

**Two dimensional array declaration:**

**Example:**

**int[][] a;**

**int [][]a;**

**int a[][]; All are valid.(6 ways)**

**int[] []a;**

**int[] a[];**

**int []a[];**

**Three dimensional array declaration:**

**Example:**

**int[][][] a;**

**int [][][]a;**

**int a[][][];**

**int[] [][]a;**

**int[] a[][]; All are valid.(10 ways)**

**int[] []a[];**

**int[][] []a;**

**int[][] a[];**

**int []a[][];**

**int [][]a[];**

**Which of the following declarations are valid?**

**1) int[] a1,b1; //a-1,b-1 (valid)**

**2) int[] a2[],b2; //a-2,b-1 (valid)**

**3) int[] []a3,b3; //a-2,b-2 (valid)**

**4) int[] a,[]b; //C.E:<identifier> expected (invalid)**

**Note :**

**If we want to specify the dimension before the variable that rule is applicable only for the 1st variable.**

**Second variable onwards we can't apply in the same declaration.**

**Example:**

**Array construction:**

**Every array in java is an object hence we can create by using new operator.**

**Example:**

**int[] a=new int[3];**

**Diagram:**

**For every array type corresponding classes are available but these classes are part of java language and not available to the programmer level.**

| **Array Type** | **corresponding class name** |
| --- | --- |
| **int[]** | **[I** |
| **int[][]** | **[[I** |
| **double[]** | **[D** |

**Rule 1:**

**At the time of array creation compulsory we should specify the size otherwise we will get compile time error.**

**Example:**

**int[] a=new int[3];**

**int[] a=new int[];//C.E:array dimension missing**

**Rule 2:**

**It is legal to have an array with size zero in java.**

**Example:**

**int[] a=new int[0];**

**System.out.println(a.length);//0**

**Rule 3:**

**If we are taking array size with -ve int value then we will get runtime exception saying NegativeArraySizeException.**

**Example:**

**int[] a=new int[-3];//R.E:NegativeArraySizeException**

**Rule 4:**

**The allowed data types to specify array size are byte, short, char, int.**

**By mistake if we are using any other type we will get compile time error.**

**Example:**

**int[] a=new int['a'];//(valid)**

**byte b=10;**

**int[] a=new int[b];//(valid)**

**short s=20;**

**int[] a=new int[s];//(valid)**

**int[] a=new int[10l];//C.E:possible loss of precision//(invalid)**

**int[] a=new int[10.5];//C.E:possible loss of precision//(invalid)**

**Rule 5:**

**The maximum allowed array size in java is maximum value of int size [2147483647].**

**Example:**

**int[] a1=new int[2147483647];(valid)**

**int[] a2=new int[2147483648];**

**//C.E:integer number too large: 2147483648(invalid)**

**In the first case we may get RE : OutOfMemoryError.**

**Multi dimensional array creation:**

**In java multidimensional arrays are implemented as array of arrays approach but not matrix form.**

**The main advantage of this approach is to improve memory utilization.**

**Example 1:**

**int[][] a=new int[2][];**

**a[0]=new int[3];**

**a[1]=new int[2];**

**Diagram:**

**Example 2:**

**int[][][] a=new int[2][][];**

**a[0]=new int[3][];**

**a[0][0]=new int[1];**

**a[0][1]=new int[2];**

**a[0][2]=new int[3];**

**a[1]=new int[2][2];**

**Diagram:**

**Which of the following declarations are valid?**

**1) int[] a=new int[]//C.E: array dimension missing(invalid)**

**2) int[][] a=new int[3][4];(valid)**

**3) int[][] a=new int[3][];(valid)**

**4) int[][] a=new int[][4];//C.E:']' expected(invalid)**

**5) int[][][] a=new int[3][4][5];(valid)**

**6) int[][][] a=new int[3][4][];(valid)**

**7) int[][][] a=new int[3][][5];//C.E:']' expected(invalid)**

**Array Initialization:**

**Whenever we are creating an array every element is initialized with default value automatically.**

**Example 1:**

**int[] a=new int[3];**

**System.out.println(a);//[I@3e25a5**

**System.out.println(a[0]);//0**

**Diagram:**

**Note: Whenever we are trying to print any object reference internally toString() method will be executed which is implemented by default to return the following. classname@hexadecimalstringrepresentationofhashcode.**

**Example 2:**

**System.out.println(a);//[[I@3e25a5**

**System.out.println(a[0]);//[I@19821f**

**System.out.println(a[0][0]);//0**

**Diagram:**

**Example 3:**

**int[][] a=new int[2][];**

**System.out.println(a);//[[I@3e25a5**

**System.out.println(a[0]);//null**

**System.out.println(a[0][0]);//R.E:NullPointerException**

**Diagram:**

**Once we created an array all its elements by default initialized with default values.**

**If we are not satisfied with those default values then we can replays with our customized values.**

**Example:**

**int[] a=new int[4];**

**a[0]=10;**

**a[1]=20;**

**a[2]=30;**

**a[3]=40;**

**a[4]=50;//R.E:ArrayIndexOutOfBoundsException: 4**

**a[-4]=60;//R.E:ArrayIndexOutOfBoundsException: -4**

**Diagram:**

**Note: if we are trying to access array element with out of range index we will get Runtime Exception saying ArrayIndexOutOfBoundsException.**

**Declaration, construction and initialization of an array in a single line:**

**We can perform declaration, construction and initialization of an array in a single line.**

**Example:**

**char[] ch={'a','e','i','o','u'};(valid)**

**String[] s={"balayya","venki","nag","chiru"};(valid)**

**We can extend this short cut even for multi dimensional arrays also.**

**Example:**

**int[][] a={{10,20,30},{40,50}};**

**Diagram:**

**Example:**

**int[][][] a={{{10,20,30},{40,50}},{{60},{70,80},{90,100,110}}};**

**Diagram:**

**int[][][] a={{{10,20,30},{40,50}},{{60},{70,80},{90,100,110}}};**

**System.out.println(a[0][1][1]);//50(valid)**

**System.out.println(a[1][0][2]);//R.E:ArrayIndexOutOfBoundsException: 2(invalid)**

**System.out.println(a[1][2][1]);//100(valid)**

**System.out.println(a[1][2][2]);//110(valid)**

**System.out.println(a[2][1][0]);//R.E:ArrayIndexOutOfBoundsException: 2(invalid)**

**System.out.println(a[1][1][1]);//80(valid)**

* **If we want to use this short cut compulsory we should perform declaration, construction and initialization in a single line.**
* **If we are trying to divide into multiple lines then we will get compile time error.**

**Example:**

**length Vs length():**

**length:**

**1.** **It is the final variable applicable only for arrays.**

**2.** **It represents the size of the array.**

**Example:**

**int[] x=new int[3];**

**System.out.println(x.length());//C.E: cannot find symbol**

**System.out.println(x.length);//3**

**length() method:**

**1.** **It is a final method applicable for String objects.**

**2.** **It returns the no of characters present in the String.**

**Example:**

**String s="bhaskar";**

**System.out.println(s.length);//C.E:cannot find symbol**

**System.out.println(s.length());//7**

**In multidimensional arrays length variable represents only base size but not total size.**

**Example:**

**int[][] a=new int[6][3];**

**System.out.println(a.length);//6**

**System.out.println(a[0].length);//3**

**Print the array elements**

**—--------------------------------------**

**public class ArrayExample {**

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

**// Declare and initialize an array**

**int[] numbers = {10, 20, 30, 40, 50};**

**// Access elements using a loop**

**for (int i = 0; i < numbers.length; i++) {**

**System.out.println("Element at index " + i + ": " + numbers[i]);**

**}**

**}**

**}**

**Same program using scanner class**

**—-------------------------------------------------------**

**package com.codegnan.arrayprograms;**

**import java.util.Scanner;**

**public class ArrayExampleWithScanner {**

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

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

**// Prompt user for array size**

**System.*out*.print("Enter the size of the array: ");**

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

**// Declare the array**

**int[] numbers = new int[size];**

**// Prompt user to enter elements**

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

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

**System.*out*.print("Element at index " + i + ": ");**

**numbers[i] = scanner.nextInt();**

**}**

**// Access elements using a loop**

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

**for (int i = 0; i < numbers.length; i++) {**

**System.*out*.println("Element at index " + i + ": " + numbers[i]);**

**}**

**// Close the scanner**

**scanner.close();**

**}**

**}**

**Count the Number of elements in an array**

**—-----------------------------------------------**

**package com.codegnan.arrayprograms;**

**public class ArrayLength {**

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

**int[] numbers = {1, 2, 3, 4, 5, 6, 7, 8, 9};**

**// Print the number of elements in the array**

**System.*out*.println("Number of elements in the array: " + numbers.length);**

**}**

**}**

**Same program using scanner class**

**package com.codegnan.arrayprograms;**

**import java.util.Scanner;**

**public class ArrayLength {**

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

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

**// Prompt user for the number of elements in the array**

**System.out.print("Enter the number of elements in the array: ");**

**int n = scanner.nextInt();kll**

**// Create an array to store the elements**

**int[] numbers = new int[n];**

**// Input each element of the array**

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

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

**numbers[i] = scanner.nextInt();**

**}**

**// Print the number of elements in the array**

**System.out.println("Number of elements in the array: " + numbers.length);**

**// Close the scanner**

**scanner.close();**

**}**

**}**

**1. Sum of Array Elements**

**—-----------------------------------------------------**

**public class SumOfArray {**

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

**//Initialize array**

**int [] arr = new int [] {1, 2, 3, 4, 5};**

**int sum = 0;**

**//Loop through the array to calculate sum of elements**

**for (int i = 0; i < arr.length; i++) {**

**sum = sum + arr[i];**

**}**

**System.out.println("Sum of all the elements of an array: " + sum);**

**}**

**}**

**Same program by using scanner class**

**—--------------------------------------------------------**

**package com.codegnan.arrayprograms;**

**import java.util.Scanner;**

**public class SumOfArrayWithScanner {**

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

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

**// Prompt user for array size**

**System.*out*.print("Enter the size of the array: ");**

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

**// Declare the array**

**int[] numbers = new int[size];**

**// Prompt user to enter elements**

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

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

**System.*out*.print("Element at index " + i + ": ");**

**numbers[i] = scanner.nextInt();**

**}**

**// Calculate sum of array elements**

**int sum = 0;**

**for (int number : numbers) {**

**sum += number;**

**}**

**// Print the sum of array elements**

**System.*out*.println("Sum of array elements: " + sum);**

**// Close the scanner**

**scanner.close();**

**}**

**}**

**3. Copying Elements from One Array to Another**

**—-----------------------------------------------------------------**

**package com.codegnan.arrayprograms;**

**public class CopyArray {**

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

**int[] sourceArray = {1, 2, 3, 4, 5};**

**int[] destinationArray = new int[sourceArray.length];**

**// Copy elements from sourceArray to destinationArray**

**for (int i = 0; i < sourceArray.length; i++) {**

**destinationArray[i] = sourceArray[i];**

**}**

**// Print elements of destinationArray**

**System.*out*.print("Copied array elements: ");**

**for (int number : destinationArray) {**

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

**}**

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

**}**

**}**

**4. Reversing an Array**

**—-----------------------------------**

**package com.codegnan.arrayprograms;**

**public class ReverseArray {**

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

**//Initialize array**

**int [] arr = new int [] {1, 2, 3, 4, 5};**

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

**for (int i = 0; i < arr.length; i++) {**

**System.out.print(arr[i] + " ");**

**}**

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

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

**//Loop through the array in reverse order**

**for (int i = arr.length-1; i >= 0; i--) {**

**System.out.print(arr[i] + " ");**

**}**

**}**

**}**

**5. Finding Average of Array Elements**

**—-----------------------------------------------------**

**package com.codegnan.arrayprograms;**

**public class AverageOfArray {**

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

**int[] numbers = {1, 2, 3, 4, 5};**

**double average = 0.0;**

**// Calculate average of array elements**

**for (int number : numbers) {**

**average += number;**

**}**

**average /= numbers.length;**

**System.*out*.println("Average of array elements: " + average);**

**}**

**}**

**6. Count Occurrences of an Element in Array**

**—---------------------------------------------------------------------**

**package com.codegnan.arrayprograms;**

**public class CountOccurrences {**

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

**int[] numbers = {1, 2, 3, 4, 2, 5, 2};**

**int target = 2;**

**int count = 0;**

**// Count occurrences of target in numbers array**

**for (int number : numbers) {**

**if (number == target) {**

**count++;**

**}**

**}**

**System.*out*.println("Occurrences of " + target + " in array: " + count);**

**}**

**}**

**7. Sum of odd even numbers in an array**

**—-----------------------------------------------------------**

**package com.codegnan.arrayprograms;**

**public class SumEvenOddNumbers {**

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

**int[] numbers = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};**

**// Initialize sums**

**int sumEven = 0;**

**int sumOdd = 0;**

**// Iterate through the array**

**for (int number : numbers) {**

**if (number % 2 == 0) {**

**// Number is even**

**sumEven += number;**

**} else {**

**// Number is odd**

**sumOdd += number;**

**}**

**}**

**// Print the sums**

**System.*out*.println("Sum of even numbers: " + sumEven);**

**System.*out*.println("Sum of odd numbers: " + sumOdd);**

**}**

**}**

**Odd and even numbers in an array**

**—-----------------------------------------------**

**package com.codegnan.arrayprograms;**

**public class OddEvenNumbers {**

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

**int[] numbers = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};**

**// Arrays to store odd and even numbers**

**int[] oddNumbers = new int[numbers.length];**

**int[] evenNumbers = new int[numbers.length];**

**int oddCount = 0;**

**int evenCount = 0;**

**// Iterate through the numbers array**

**for (int number : numbers) {**

**if (number % 2 != 0) {**

**// Number is odd**

**oddNumbers[oddCount] = number;**

**oddCount++;**

**} else {**

**// Number is even**

**evenNumbers[evenCount] = number;**

**evenCount++;**

**}**

**}**

**// Print odd numbers**

**System.*out*.print("Odd numbers: ");**

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

**System.*out*.print(oddNumbers[i] + " ");**

**}**

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

**// Print even numbers**

**System.*out*.print("Even numbers: ");**

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

**System.*out*.print(evenNumbers[i] + " ");**

**}**

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

**}**

**}**

**Odd and even number index positions an array**

**—---------------------------------------------------------**

**package com.codegnan.arrayprograms;**

**public class OddEvenPosition {**

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

**int[] numbers = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};**

**System.*out*.print("Elements at odd indices: ");**

**for (int i = 1; i < numbers.length; i += 2) {**

**System.*out*.print(numbers[i] + " ");**

**}**

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

**System.*out*.print("Elements at even indices: ");**

**for (int i = 0; i < numbers.length; i += 2) {**

**System.*out*.print(numbers[i] + " ");**

**}**

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

**}**

**}**

**Frequency array elements in an array**

**—--------------------------------------------------**

**package com.codegnan.arrayprograms;**

**public class FrequencyCounter1 {**

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

**// Initialize the array with some sample elements**

**int[] array = { 1, 2, 3, 2, 1, 4, 3, 5, 1 };**

**// A boolean array to track which elements have been processed already**

**boolean[] processed = new boolean[array.length];**

**// Outer loop to iterate over each element in the array**

**for (int i = 0; i < array.length; i++) {**

**// If the element at index i has already been processed, skip it**

**if (processed[i])**

**continue;**

**// Initialize the count for the current element**

**int count = 1;**

**// Inner loop to compare the current element with the rest of the elements**

**for (int j = i + 1; j < array.length; j++) {**

**// If the element at index j is equal to the element at index i**

**if (array[j] == array[i]) {**

**count++; // Increment the count for that element**

**processed[j] = true; // Mark the element at index j as processed**

**}**

**}**

**// Output the element and its count**

**System.out.println(array[i] + " occurs " + count + " times");**

**}**

**}**

**}**

**Same program on strings array**

**----------------------------------------------**

**public class FrequencyCounter {**

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

**// Initialize the array with some sample strings**

**String[] array = { "apple", "banana", "apple", "cherry", "banana", "apple" };**

**// Create a boolean array to track which elements have been processed**

**boolean[] processed = new boolean[array.length];**

**// Traverse through each element of the array**

**for (int i = 0; i < array.length; i++) {**

**// Skip the current element if it has already been processed**

**if (processed[i])**

**continue; // Skip if the element at index i has been processed**

**// Initialize the element to be processed and set its initial count to 1**

**String element = array[i];**

**int count = 1;**

**// Traverse the remaining elements starting from i+1 to count occurrences of the current element**

**for (int j = i + 1; j < array.length; j++) {**

**// If an element matches the current element, increment the count and mark it as processed**

**if (array[j].equals(element)) {**

**count++; // Increment the count for the current element**

**processed[j] = true; // Mark the element at index j as processed**

**}**

**}**

**// Print the frequency of the current element**

**System.out.println(element + " occurs " + count + " times");**

**}**

**}**

**}**

**Write a java program to check each and every string elements of an array odd and even**

**---------------------------------------------------------------------------------------------------------------------**

**public class EvenOddStringArrays {**

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

**// Initialize an array of strings**

**String[] strings = { "apple", "banana", "cherry", "date", "fig", "grape" };**

**// Variables to count even and odd length strings**

**int evenCount = 0, oddCount = 0;**

**// First pass: Count even and odd length strings**

**for (String str : strings) {**

**if (str.length() % 2 == 0) { // If the string's length is even**

**evenCount++; // Increment evenCount**

**} else { // If the string's length is odd**

**oddCount++; // Increment oddCount**

**}**

**}**

**// Create arrays to store even and odd length strings**

**String[] evenLengthStrings = new String[evenCount]; // Array for even length strings**

**String[] oddLengthStrings = new String[oddCount]; // Array for odd length strings**

**// Second pass: Fill the arrays with even and odd length strings**

**int evenIndex = 0, oddIndex = 0; // Indices to track where to insert the strings**

**for (String str : strings) {**

**if (str.length() % 2 == 0) { // If the string has even length**

**evenLengthStrings[evenIndex++] = str; // Add to even array**

**} else { // If the string has odd length**

**oddLengthStrings[oddIndex++] = str; // Add to odd array**

**}**

**}**

**// Print even length strings**

**System.out.println("Even length strings:");**

**for (String str : evenLengthStrings) {**

**System.out.println(str); // Print each string in the evenLengthStrings array**

**}**

**// Print odd length strings**

**System.out.println("\nOdd length strings:");**

**for (String str : oddLengthStrings) {**

**System.out.println(str); // Print each string in the oddLengthStrings array**

**}**

**}**

**}**

**Write a java program to check each and every element inside an array is palindrome or not**

**-------------------------------------------------------------------------------------------------------------**

**public class PalindromeChecker {**

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

**// An array of strings to be checked for palindromes**

**String[] strings = { "madam", "racecar", "hello", "world", "level", "java" };**

**// Iterate through each string and check if it's a palindrome**

**for (String str : strings) {**

**// Print the result of palindrome check using ternary operator**

**System.out.println(str + " is " + (isPalindrome(str) ? "a palindrome." : "not a palindrome."));**

**}**

**}**

**// Method to check if a string is a palindrome**

**public static boolean isPalindrome(String str) {**

**// Left pointer at the start of the string, right pointer at the end**

**int left = 0, right = str.length() - 1;**

**// Loop until the two pointers meet in the middle**

**while (left < right) {**

**// If characters at the left and right pointers don't match, return false**

**if (str.charAt(left++) != str.charAt(right--)) {**

**return false;**

**}**

**}**

**// If all characters matched, return true (it's a palindrome)**

**return true;**

**}**

**}**

**Assending and Desending order of eleemnts in an array**

**-------------------------------------------------------------------------------**

**public class ArrayAscDscExample {**

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

**// Initialize an array with some integer values**

**int[] arr = new int[] { 5, 2, 8, 7, 1 };**

**int temp; // Temporary variable for swapping elements**

**// Display the elements of the original array**

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

**for (int i = 0; i < arr.length; i++) {**

**System.*out*.print(arr[i] + " "); // Print each element followed by a space**

**}**

**// Sort the array in ascending order using a simple sorting algorithm**

**for (int i = 0; i < arr.length - 1; i++) { // Outer loop to traverse the array**

**for (int j = i + 1; j < arr.length; j++) { // Inner loop to compare elements**

**if (arr[i] > arr[j]) { // Compare current element with the next element**

**// Swap elements if they are in the wrong order**

**temp = arr[i]; // Store the current element in a temporary variable**

**arr[i] = arr[j]; // Place the next element in the current position**

**arr[j] = temp; // Place the original element in the next position**

**}**

**}**

**}**

**// Print a new line for better readability in output**

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

**// Display the elements of the array after sorting**

**System.*out*.println("Elements of the array sorted in descending order:");**

**for (int i = 0; i < arr.length; i++) {**

**System.*out*.print(arr[i] + " "); // Print each element followed by a space**

**}**

**// Sort the array in ascending order using a simple sorting algorithm**

**for (int i = 0; i < arr.length - 1; i++) { // Outer loop to traverse the array**

**for (int j = i + 1; j < arr.length; j++) { // Inner loop to compare elements**

**if (arr[i] < arr[j]) { // Compare current element with the next element**

**// Swap elements if they are in the wrong order**

**temp = arr[i]; // Store the current element in a temporary variable**

**arr[i] = arr[j]; // Place the next element in the current position**

**arr[j] = temp; // Place the original element in the next position**

**}**

**}**

**}**

**// Print a new line for better readability in output**

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

**// Display the elements of the array after sorting**

**System.*out*.println("Elements of the array sorted in ascending order:");**

**for (int i = 0; i < arr.length; i++) {**

**System.*out*.print(arr[i] + " "); // Print each element followed by a space**

**}**

**}**

**}**

3) You&#39;re working on a voting system for an election. The system needs to calculate the voting

results and determine the winner based on the votes cast by the voters.

Design and implement the necessary Java classes and interface for this voting system.

Define an interface named VoteCounter with the following methods:

1. countVotes(String[] candidates, int[] votes): This method should calculate the total number of

votes received by each candidate based on the array of candidates and the corresponding array

of votes provided as input. It should return an array containing the vote counts for each candidate

in the same order as the candidate&#39;s array.

2. determineWinner(int[] voteCounts): This method should determine the winner of the election

based on the vote counts provided as input. It should return an integer indicating the index of the

winning candidate in the voteCounts array.

Implement a class named ElectionSystem that implements the VoteCounter interface. In this

class, provide the necessary logic for both the countVotes() and determineWinner() methods.

Note: The main class with the driver code has been provided to you in the editor.

Code:

package q28926;

import java.util.Scanner;

interface VoteCounter { // fill the missing code...

int[] countVotes(String[] candidates, int[] votes);

int determineWinner(int[] voteCounts);

// write your code here...

}

class ElectionSystem implements VoteCounter { // fill the missing code...

// write your code here...

public int[] countVotes(String[] candidates, int[] votes){

return votes;

}

public int determineWinner(int[] voteCounts){

int maxVotes=Integer.MIN\_VALUE;

int winnerIndex=-1;

boolean isTie=false;

for(int i=0;i&lt;voteCounts.length;i++){

if(voteCounts[i]&gt;maxVotes){

maxVotes=voteCounts[i];

winnerIndex=i;

isTie=false;

} else if(voteCounts[i]==maxVotes){

isTie=true;

}

}

if(isTie){

return -1;

}

return winnerIndex;

}

}

public class Main {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

ElectionSystem electionSystem = new ElectionSystem();

System.out.print(&quot;no of candidates: &quot;);

int numCandidates = scanner.nextInt();

scanner.nextLine();

String[] candidates = new String[numCandidates];

int[] votes = new int[numCandidates];

for (int i = 0; i &lt; numCandidates; i++) {

System.out.print(&quot;name of candidate &quot; + (i + 1) + &quot;: &quot;);

candidates[i] = scanner.nextLine();

}

for (int i = 0; i &lt; numCandidates; i++) {

System.out.print(&quot;votes for candidate &quot; + candidates[i] + &quot;: &quot;);

votes[i] = scanner.nextInt();

}

// Count votes

int[] voteCounts = electionSystem.countVotes(candidates, votes);

System.out.println(&quot;Vote Counts:&quot;);

for (int i = 0; i &lt; candidates.length; i++) {

System.out.println(candidates[i] + &quot;: &quot; + voteCounts[i]);

}

// Determine the winner

int winningCandidateIndex = electionSystem.determineWinner(voteCounts);

if (winningCandidateIndex == -1) {

System.out.println(&quot;It&#39;s a tie!&quot;);

} else {

System.out.println(&quot;Winner: &quot; + candidates[winningCandidateIndex]);

}

}

}