1. Declare a single-dimensional array of 5 integers inside the main method. Traverse the array to print the default values. Then accept records from the user and print the updated values of the array.

package org.example.ques1;

import java.util.Scanner;

public class program {

private static Scanner Sc = new Scanner(System.in);

private static void acceptRecord(int[] arr) {

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

System.out.println("Enter arr[" + index + "]: ");

arr[index] = Sc.nextInt();

}

}

private static void printRecord(int[] arr) {

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

System.out.println(arr[index]);

}

}

public static void main(String[] args) {

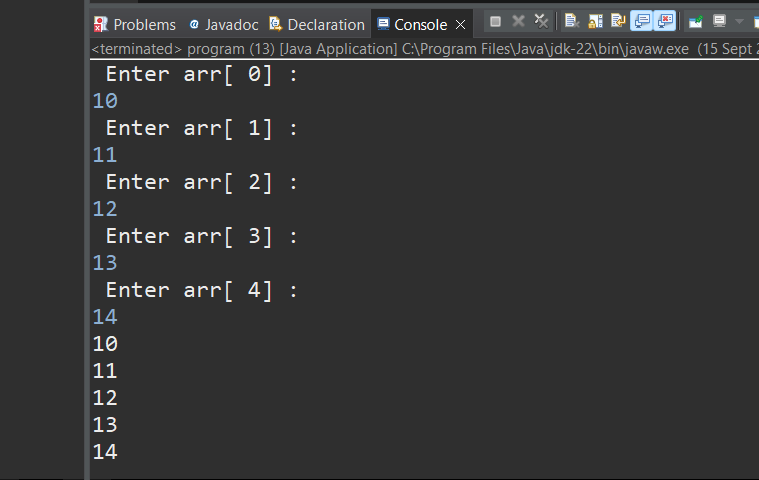
int[] arr = new int[5];

program.acceptRecord(arr);

program.printRecord(arr);

}

}



1. Declare a single-dimensional array of 5 integers inside the main method. Define a method named acceptRecord to get input from the terminal into the array and another method named printRecord to print the state of the array to the terminal.

package org.example.ques2;

import java.util.Scanner;

public class program {

// Method to accept values from the user

private static void acceptRecord(int[] arr) {

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

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

System.***out***.print("Enter value for arr[" + index + "]: ");

arr[index] = sc.nextInt();

}

}

// Method to print the values of the array

private static void printRecord(int[] arr) {

System.***out***.println("Array values:");

for (int value : arr) {

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

}

}

public static void main(String[] args) {

// Declare a single-dimensional array of 5 integers

int[] arr = new int[5];

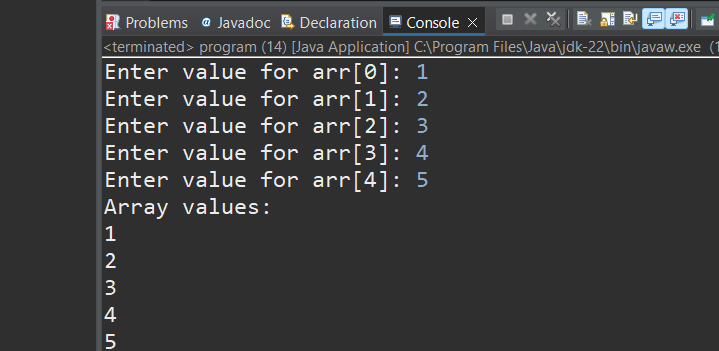
// Call methods to accept and print array values

*acceptRecord*(arr);

*printRecord*(arr);

}

}



1. Write a program to find the maximum and minimum values in a single-dimensional array of integers.

package org.example.ques3;

import java.util.Scanner;

public class program {

// Method to find the maximum value in the array

private static int findMax(int[]arr) {

int max = arr[0]; //suppose 1st element is max

for(int value: arr){

if(value>max) {

max= value; //updatemax if maximum value found

}

}

return max;

}

// Method to find the minimum value in the array

private static int findMin(int[]arr) {

int min = arr[0]; //suppose the first element is the minimum

for(int value: arr) {

if(value<min) {

min = value; //update min if a smaller value is found

}

}

return min;

}

public static void main(String[]args) {

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

int[]arr = new int [5];

//accepting input for the array

System.***out***.println("enter 5 integer values:");

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

arr[i] = sc.nextInt();

}

//finding and displaying max and min values

int max = *findMax*(arr);

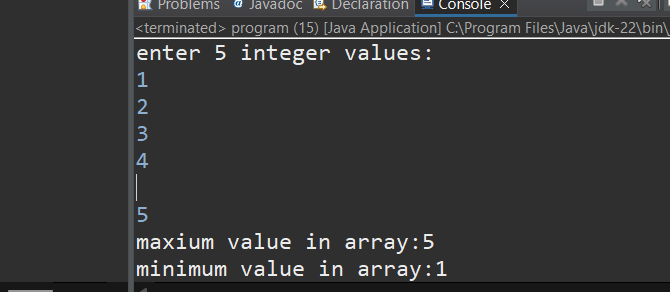
int min = *findMin*(arr);

System.***out***.println("maxium value in array:"+max);

System.***out***.println("minimum value in array:"+min);

}

}



1. Write a program to remove duplicate elements from a single-dimensional array of integers.

package org.example.ques4;

public class program {

public static int removeduplicates(int a[], int n) {

if (n == 0 || n == 1) {

return n; // No duplicates if array has 0 or 1 element

}

int[] temp = new int[n];

int j = 0;

for (int i = 0; i < n - 1; i++) {

if (a[i] != a[i + 1]) {

temp[j++] = a[i]; // Store the unique element

}

}

boolean check = true;

for (int i = 0; i < j; i++) {

if (temp[i] == a[n - 1]) { // Check if the last element is already in the array

check = false;

break;

}

}

if (check) {

temp[j++] = a[n - 1]; // Add the last element if it's unique

}

for (int i = 0; i < j; i++) {

a[i] = temp[i];

}

return j;

}

public static void main(String[] args) {

int a[] = {1, 1, 2, 2, 2, 2, 3, 4, 2}; // Array with duplicates

int n = a.length;

n = removeduplicates(a, n); // Remove duplicates and update the size

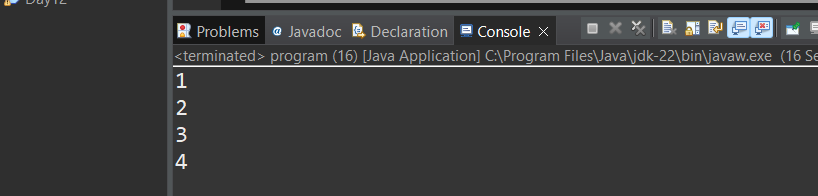
for (int i = 0; i < n; i++) {

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

}

}

}



1. Write a program to find the intersection of two single-dimensional arrays.

package org.example.ques5;

import java.util.Arrays;

import java.util.HashSet;

import java.util.Set;

public class program {

public static void main(String[] args) {

int[] arr1 = { 1, 2, 3, 4, 5 };

int[] arr2 = { 3, 4, 5, 6, 7 };

// Convert arrays to sets for easy intersection calculation

Set<Integer> set1 = new HashSet<>();

for (int num : arr1) {

set1.add(num);

}

Set<Integer> set2 = new HashSet<>();

for (int num : arr2) {

set2.add(num);

}

// Calculate the intersection

set1.retainAll(set2);

// Convert the result back to an array

int[] intersection = new int[set1.size()];

int index = 0;

for (int num : set1) {

intersection[index++] = num;

}

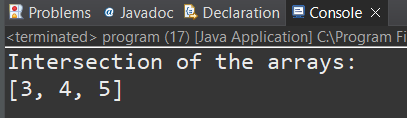
// Print the intersection

System.***out***.println("Intersection of the arrays:");

System.***out***.println(Arrays.*toString*(intersection));

}

}



1. Write a program to find the missing number in an array of integers ranging from 1 to N.

package org.example.ques6;

import java.util.Scanner;

public class program {

public static void main(String[] args) {

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

// Input the size of the array (N)

System.***out***.print("Enter the value of N: ");

int N = sc.nextInt();

// Input the elements of the array

int[] arr = new int[N - 1];

System.***out***.println("Enter the array elements (1 to N with one missing):");

int sum = 0;

for (int i = 0; i < N - 1; i++) {

arr[i] = sc.nextInt();

sum += arr[i];

}

// Calculate the expected sum from 1 to N

int totalSum = N \* (N + 1) / 2;

// The missing number is the difference between totalSum and sum

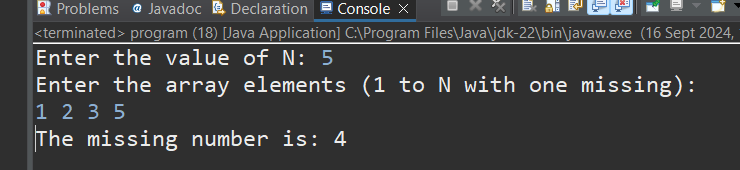
int missingNumber = totalSum - sum;

// Output the missing number

System.***out***.println("The missing number is: " + missingNumber);

}

}



1. Declare a single-dimensional array as a field inside a class and instantiate it inside the class constructor. Define methods named acceptRecord and printRecord within the class and test their functionality.

package org.example.ques7;

import java.util.Scanner;

class Record {

private int[] arr;

public Record(int size) {

arr = new int[size];

}

public void acceptRecord() {

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

System.***out***.println("Enter " + arr.length + " elements:");

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

arr[i] = sc.nextInt();

}

}

public void printRecord() {

System.***out***.println("Array elements are:");

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

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

}

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

}

}

public class program {

public static void main(String[] args) {

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

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

int size = sc.nextInt();

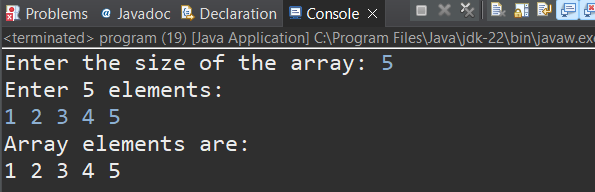
Record record = new Record(size);

record.acceptRecord(); // Accept the array elements

record.printRecord(); // Print the array elements

}

}



1. Modify the previous assignment to use getter and setter methods instead of acceptRecord and printRecord.

package org.example.ques8;

import java.util.Scanner;

public class program {

private int[] numbers; //single array

//constructor to instantiate

public program(int size) {

numbers = new int[size];

}

//setter method to set values

public void setNumbers(int[] numbers) {

if(numbers.length == this.numbers.length) {

this.numbers = numbers;

}else {

System.***out***.println("Array size is missmatch");

}

}

public int[]getNumbers(){

return numbers;

}

public static void main(String[]args) {

program example = new program(5); //institantiate the class

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

int[]inputArray = new int[5];

System.***out***.println("enter 5 numbers");

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

inputArray[i]= sc.nextInt();

}

example.setNumbers(inputArray);

int[] retrivedArray = example.getNumbers();

System.***out***.println("array elements are ");

for(int num: retrivedArray) {

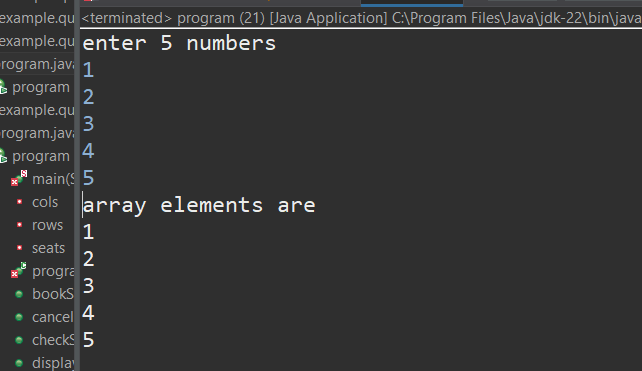
System.***out***.println(num+ " ");

}

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

}

}



1. You need to implement a system to manage airplane seat assignments. The airplane has seats arranged in rows and columns. Implement functionalities to:

* Initialize the seating arrangement with a given number of rows and columns.
* Book a seat to mark it as occupied.
* Cancel a booking to mark a seat as available.
* Check seat availability to determine if a specific seat is available.
* Display the current seating chart.

package org.example.ques9;

import java.util.Scanner;

public class program {

private char[][] seats;

private int rows;

private int cols;

public program(int rows, int cols) {

this.rows = rows;

this.cols = cols;

seats = new char[rows][cols];

for (int i = 0; i < rows; i++) {

for (int j = 0; j < cols; j++) {

seats[i][j] = 'A';

}

}

}

// Method to book a seat

public void bookSeat(int row, int col) {

if (row < 0 || row >= rows || col < 0 || col >= cols) {

System.***out***.println("Invalid seat number!");

} else if (seats[row][col] == 'B') {

System.***out***.println("Seat already booked.");

} else {

seats[row][col] = 'B';

System.***out***.println("Seat booked successfully.");

}

}

public void cancelBooking(int row, int col) {

if (row < 0 || row >= rows || col < 0 || col >= cols) {

System.***out***.println("Invalid seat number!");

} else if (seats[row][col] == 'A') {

System.***out***.println("Seat is not booked.");

} else {

seats[row][col] = 'A';

System.***out***.println("Booking cancelled.");

}

}

public void checkSeatAvailability(int row, int col) {

if (row < 0 || row >= rows || col < 0 || col >= cols) {

System.***out***.println("Invalid seat number!");

} else if (seats[row][col] == 'A') {

System.***out***.println("Seat is available.");

} else {

System.***out***.println("Seat is booked.");

}

}

public void displaySeatingChart() {

System.***out***.println("Current seating arrangement:");

for (int i = 0; i < rows; i++) {

for (int j = 0; j < cols; j++) {

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

}

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

}

}

public static void main(String[] args) {

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

System.***out***.print("Enter the number of rows: ");

int rows = sc.nextInt();

System.***out***.print("Enter the number of columns: ");

int cols = sc.nextInt();

program airplane = new program(rows, cols);

int choice;

do {

System.***out***.println("\n1. Book a Seat");

System.***out***.println("2. Cancel Booking");

System.***out***.println("3. Check Seat Availability");

System.***out***.println("4. Display Seating Chart");

System.***out***.println("5. Exit");

System.***out***.print("Enter your choice: ");

choice = sc.nextInt();

switch (choice) {

case 1:

System.***out***.print("Enter row number to book: ");

int bookRow = sc.nextInt();

System.***out***.print("Enter column number to book: ");

int bookCol = sc.nextInt();

airplane.bookSeat(bookRow - 1, bookCol - 1);

break;

case 2:

System.***out***.print("Enter row number to cancel: ");

int cancelRow = sc.nextInt();

System.***out***.print("Enter column number to cancel: ");

int cancelCol = sc.nextInt();

airplane.cancelBooking(cancelRow - 1, cancelCol - 1);

break;

case 3:

System.***out***.print("Enter row number to check: ");

int checkRow = sc.nextInt();

System.***out***.print("Enter column number to check: ");

int checkCol = sc.nextInt();

airplane.checkSeatAvailability(checkRow - 1, checkCol - 1);

break;

case 4:

airplane.displaySeatingChart();

break;

case 5:

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

break;

default:

System.***out***.println("Invalid choice. Please try again.");

}

} while (choice != 5);

sc.close();

}

}

