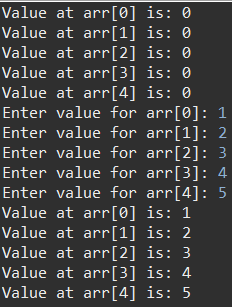
1. Declare a single-dimensional array of 5 integers inside the main method. Traverse the array to print the default values. Then accept records from the user and print the updated values of the array.
2. package org.example.str1;
3. import java.util.Scanner;
4. public class q1 {
5. public static void main(String[] args) {
6. Scanner sc = new Scanner(System.***in***);
7. int[] arr = new int[5];
8. for(int ind=0; ind<arr.length; ind++) {
9. System.***out***.println("Value at arr[" + ind + "] is: " + arr[ind]);
10. }
11. for(int ind=0; ind<arr.length; ind++) {
12. System.***out***.print("Enter value for arr[" + ind + "]: ");
13. arr[ind] = sc.nextInt();
14. }
15. for(int ind=0; ind<arr.length; ind++) {
16. System.***out***.println("Value at arr[" + ind + "] is: " + arr[ind]);
17. }
18. sc.close();
19. }
20. }



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.str2;

import java.util.Scanner;

public class Q2 {

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

public static void acceptRecord(int[] array) {

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

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

array[ind] = *sc*.nextInt();

}

}

public static void printRecord(int[] array) {

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

System.***out***.println("Value at arr[" + ind + "] is: " + array[ind]);

}

}

public static void closeResource() {

*sc*.close();

}

public static void main(String[] args) {

int[] arr = new int[5];

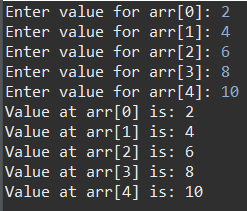
*acceptRecord*(arr);

*printRecord*(arr);

*closeResource*();

}

}



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

*Q3.java*

package org.example.str2;

public class Q2 {

public static void main(String[] args) {

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

Action ac = new Action();

System.***out***.println(ac.getMax(arr));

System.***out***.println(ac.getMin(arr));

}

}

*Action.java*

package org.example.str2;

public class Action {

private int temp;

public String getMax(int[] arr) {

temp = arr[0];

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

if(temp<arr[ind]) {

temp = arr[ind];

}

}

return "Maximum value in array is : " + temp;

}

public String getMin(int[] arr) {

temp = arr[0];

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

if(temp>arr[ind]) {

temp = arr[ind];

}

}

return "Minimum value in array is : " + temp;

}

}



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

*Q4.java*

package org.example.str3;

public class Q4 {

public static void main(String[] args) {

int[] arr1 = { 5, 6, 1, 8, 4, 3, 9, 7, 2, 4, 7, 2 };

rmDuplicate rd = new rmDuplicate();

arr1 = rd.romover(arr1);

for(int row=0; row<arr1.length; row++) {

System.***out***.print(arr1[row] + " ");

}

}

}

*rmDuplicate.java*

package org.example.str3;

public class rmDuplicate {

public int[] romover(int[] arr) {

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

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

if(arr[row] == arr[col] && row != col) {

arr[col] = 0;

}

}

}

return arr;

}

}



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

Q5.java

package org.example.str5;

public class Q5 {

public static void main(String[] args) {

int[] arr1 = { 5, 6, 1, 8, 4, 3, 9, 7, 2 };

int[] arr2 = { 4,3,5,2,6,7,9,0 };

int[] arr3;

Action ac = new Action();

arr3 = ac.intersect(arr1, arr2);

ac.getIntersected(arr3);

}

}

Action.java

package org.example.str5;

public class Action {

private int[] temp = new int[10];

public int[] intersect(int[] arr1, int[] arr2) {

int count = 0;

for(int row=0; row<arr1.length; row++) {

for(int col=0; col<arr2.length; col++) {

if(arr1[row] == arr2[col]) {

temp[count] = arr1[row];

count++;

}

}

}

return temp;

}

public String getIntersected(int[] arr) {

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

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

}

return "Minimum value in array is : " + temp;

}

}



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

*Q6.java*

package org.example.str6;

public class Q6 {

public static void main(String[] args) {

int[] arr1 = { 1,2,4,5,8,9 };

FindMissing ac = new FindMissing();

ac.intersect(arr1);

}

}

*FindMissing.java*

package org.example.str6;

public class FindMissing {

public void intersect(int[] arr1) {

for(int row=0; row<(arr1.length-1); row++) {

if((arr1[row+1]-arr1[row]) > 1) {

System.***out***.print("Missing numbers are: ");

for(int count=1; count<(arr1[row+1]-arr1[row]); count++) {

System.***out***.print((arr1[row] + count) + " ");

}

}

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

}

}

}



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.

Q7.java

package org.example.str7;

public class Q7 {

public static void main(String[] args) {

ArrayConstructor ac = new ArrayConstructor();

ac.acceptRecord();

ac.printRecord();

}

}

ArrayConstruction.java

package org.example.str7;

import java.util.Scanner;

public class ArrayConstructor {

int[] arr = null;

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

public ArrayConstructor(){

arr = new int[5];

}

public void acceptRecord() {

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

System.***out***.print("Enter element for array: ");

this.arr[count] = sc.nextInt();

}

}

public void printRecord() {

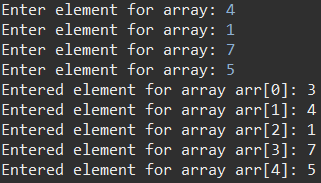
for(int count=0; count<arr.length; count++) {

System.***out***.println("Entered element for array arr[" + count + "]: " + arr[count]);

}

}

}



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

*Q8.java*

package org.example.str8;

import java.util.Scanner;

public class Q8 {

public static void main(String[] args) {

int[] arr = new int[5];

ArrayConstructor2 ac = new ArrayConstructor2();

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

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

System.***out***.print("Enter element for array: ");

arr[count] = sc.nextInt();

}

ac.setArr(arr);

System.***out***.print(ac.toString());

sc.close();

}

}

*ArrayConstructor2.java*

package org.example.str8;

public class ArrayConstructor2 {

int[] arr = null;

public ArrayConstructor2(){

arr = new int[5];

}

public int[] getArr() {

return arr;

}

public void setArr(int[] arr) {

this.arr = arr;

}

public String toString() {

String str = "Array is: ";

for (int array : this.getArr()) {

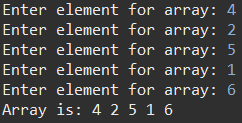
str += array + " ";

}

return str;

}

}



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.

*Booking.java*

package org.example.arr9;

public class Booking {

public static void main(String[] args) {

FlightSeat2 fs = new FlightSeat2();

fs.displayChart();

fs.checkAvailability();

fs.cancelBooking();

fs.displayChart();

fs.checkAvailability();

}

}

***FlighSeat2.java***

package org.example.arr9;

import java.util.Scanner;

public class FlightSeat2 {

public enum *Options*{

***YES***, ***NO***

}

int selectedSeat;

int calRow;

int calCol;

*Options* tempMark;

private static String[][] *arrangement*;

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

public FlightSeat2() {

FlightSeat2.*arrangement* = new String[12][4];

}

public FlightSeat2(String[][] arrangement) {

FlightSeat2.*arrangement* = arrangement;

}

public void displayChart() {

int seatNo = 1;

for(int row=0; row<*arrangement*.length; row++) {

for(int col=0; col<*arrangement*[row].length; col++) {

if(*arrangement*[row][col] != null) {

System.***out***.printf("%5s", "-" + " ");

}

else {

System.***out***.printf("%5s", seatNo + " ");

}

//**TODO** I can put 1 if-else to show if seat is not available print dash else seatNo. This would combine displaChart and checkAvailability functionality

seatNo++;

if(col == (FlightSeat2.*arrangement*[col].length/2)-1) { //checking for mid point of rows

System.***out***.print(" "); //adding extra space in between for better look

}

}

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

}

}

public void checkAvailability() {

System.***out***.print("Enter seat no. to check availability: ");

this.selectedSeat = *sc*.nextInt();

this.calRow = (selectedSeat-1)/4; //calculating row number

this.calCol = (selectedSeat-1)%4; //calculating coulnm number

//System.out.println(calRow + " " + calCol); to check which row and column seat is in.

if(*arrangement*[this.calRow][this.calCol] == null) {

System.***out***.println("Phew! This seat is available");

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

this.markSeat();

}

else{

System.***out***.println("Ohho! This seat is already booked.");

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

//**TODO** if-else if they want to check for another seat. //recursive, required?

}

}

public void markSeat() {

System.***out***.print("Would you like to mark your seat? ");

tempMark = this.menuList();

if(tempMark == *Options*.***NO***) {

*arrangement*[this.calRow][this.calCol] = "Marked";

System.***out***.println("Seat is marked for time being!");

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

this.bookSeat();

}

//**TODO** idk

}

public void bookSeat() {

System.***out***.print("Would you like to confirm your seat booking? ");

tempMark = this.menuList();

if(tempMark == *Options*.***NO***) {

*arrangement*[this.calRow][this.calCol] = "Booked";

System.***out***.println("Congrats! your seat is booked.");

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

}

}

public void cancelBooking() {

System.***out***.print("Would you like to cancel your seat booking of seat no. " + this.selectedSeat + "? ");

tempMark = this.menuList();

if(tempMark == *Options*.***NO***) {

*arrangement*[this.calRow][this.calCol] = null;

System.***out***.println("Your seat is cancelled.");

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

}

}

public *Options* menuList() {

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

System.***out***.println("0. No");

System.***out***.println("1. Yes");

return *Options*.*values*()[*sc*.nextInt()];

}

}

