**ASSIGNMENT-11**

**Question-1**

**Problem Statement:** WAP in Java to create two classes such as TwoDArray and OneDArray.

Members of TwoDArray class are:

a) One 2D array.

b) Constructor to allocate memory of size (2 X n) for the array.

c) Input data into the Array.

d) Display the array elements in row wise.

e) Split this array into two different arrays and store them in the array available in two different objects of class OneDArray.

f) Add the arrays(m X n matrices) of two objects of TwoDArray class.

g) Multiply the arrays(m X n matrices) of two objects of TwoDArray class.

Members of OneDArray class are:

a) One 1D array.

b) Constructor to allocate memory of size (n) as the column size of TwoDArray class.

c) Show the array.

**Source Code**

// Class TwoDArray

import java.util.\*;

public class TwoDArray {

int[][] arr;

int n;

TwoDArray(int n)

{

this.n=n;

arr=new int[2][n];

}

public void inputElements() //method to input elements in the 2D-array

{

Scanner sc=new Scanner(System.in);

System.out.println("Enter elements in the 2D array:");

for(int i=0;i<2;i++)

{

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

arr[i][j] = sc.nextInt();

}

}

}

public void display() //method to display elements in the 2D-array

{

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

for(int i=0;i<2;i++)

{

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

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

}

System.out.println();

}

}

// Split the array into two different arrays and store them in the array available in two different objects of class OneDArray

public void splitIntoOneDArray(OneDArray obj1, OneDArray obj2) {

for (int j = 0; j < arr[0].length; j++) {

obj1.arr[j] = arr[0][j];

obj2.arr[j] = arr[1][j];

}

}

// Add the arrays (m X n matrices) of two objects of TwoDArray class

public static TwoDArray addArrays(TwoDArray obj1, TwoDArray obj2) {

int[][] resultarr = new int[2][obj1.arr[0].length];

for (int i = 0; i < 2; i++) {

for (int j = 0; j < obj1.arr[0].length; j++) {

resultarr[i][j] = obj1.arr[i][j] + obj2.arr[i][j];

}

}

TwoDArray result = new TwoDArray(obj1.arr[0].length);

result.arr = resultarr;

return result;

}

// Multiply the arrays (m X n matrices) of two objects of TwoDArray class

public static TwoDArray multiplyArrays(TwoDArray obj1, TwoDArray obj2) {

int[][] resultarr = new int[2][obj1.arr[0].length];

for (int i = 0; i < 2; i++) {

for (int j = 0; j < obj1.arr[0].length; j++) {

for (int k = 0; k < obj1.arr.length; k++) {

resultarr[i][j] += obj1.arr[i][k] \* obj2.arr[k][j];

}

}

}

TwoDArray result = new TwoDArray(obj1.arr[0].length);

result.arr = resultarr;

return result;

}

}

// Class OneDArray

class OneDArray {

int[] arr;

public OneDArray(int size) {

arr = new int[size];

}

public void display() {

System.out.println("The 1D Array:");

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

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

}

System.out.println();

}

}

//Class TestArray (with main method)

import java.util.\*;

public class TestArray {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter the size of the 2D array (n): ");

int n = scanner.nextInt();

TwoDArray obj1 = new TwoDArray(n);

TwoDArray obj2 = new TwoDArray(n);

System.out.println("Enter elements for first 2D array:");

obj1.inputElements();

System.out.println("Enter elements for second 2D array:");

obj2.inputElements();

System.out.println("Displaying first 2D array:");

obj1.display();

System.out.println("Displaying second 2D array:");

obj2.display();

OneDArray oneDArray1 = new OneDArray(n);

OneDArray oneDArray2 = new OneDArray(n);

obj1.splitIntoOneDArray(oneDArray1, oneDArray2);

System.out.println("Displaying first 1D array of the first 2-D array:");

oneDArray1.display();

System.out.println("Displaying second 1D array of the first 2-D array:");

oneDArray2.display();

obj2.splitIntoOneDArray(oneDArray1, oneDArray2);

System.out.println("Displaying first 1D array of the second 2-D array:");

oneDArray1.display();

System.out.println("Displaying second 1D array of the second 2-D array:");

oneDArray2.display();

TwoDArray additionResult = TwoDArray.addArrays(obj1, obj2);

System.out.println("Addition Result:");

additionResult.display();

TwoDArray multiplicationResult = TwoDArray.multiplyArrays(obj1, obj2);

System.out.println("Multiplication Result:");

multiplicationResult.display();

}

}

**OUTPUT:**

Enter the size of the 2D array (n): 3

Enter elements for first 2D array:

Enter elements in the 2D array:

1

2

3

4

5

6

Enter elements for second 2D array:

Enter elements in the 2D array:

7

8

9

10

11

12

Displaying first 2D array:

Array elements:

1 2 3

4 5 6

Displaying second 2D array:

Array elements:

7 8 9

10 11 12

Displaying first 1D array of the first 2-D array:

The 1D Array:

1 2 3

Displaying second 1D array of the first 2-D array:

The 1D Array:

4 5 6

Displaying first 1D array of the second 2-D array:

The 1D Array:

7 8 9

Displaying second 1D array of the second 2-D array:

The 1D Array:

10 11 12

Addition Result:

Array elements:

8 10 12

14 16 18

Multiplication Result:

Array elements:

27 30 33

78 87 96

**Question-2**

**Problem Statement:** There are five brothers and sisters are trying to store their marks in one reference (array) for better analysis. But the number of subjects is different for each child as they are reading in different classes. Child1 has 3 subjects, Child2 has 5 subjects, Child3 has 2 subjects, Child4 has 6 subjects and Child5 has 4 subjects. Help them to achieve this.

Design a class JaggedArray with following members:

a) One Array to hold the marks of five brothers & sisters.

b) Constructor to allocate memory for the Array exactly as the no of subjects specified for five children.

c) Input marks in different subjects for different children.

d) Show the marks row-wise with child name at the beginning.

e) Show the total marks scored by each child.

f) Count the marks which are more than 80 separately for each child.

g) Alert them by showing the marks which are less than 30 for better preparation.

**Source Code**

// Class JaggedArray

public class JaggedArray {

private int[][] marksArray;

public JaggedArray(int[] subjectsPerChild) {

marksArray = new int[5][];

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

marksArray[i] = new int[subjectsPerChild[i]];

}

}

public void inputMarks(String childName, int[] marks) {

int index = getChildIndex(childName);

if (index != -1) {

marksArray[index] = marks;

} else {

System.out.println("Child not found!");

}

}

public void showMarks() {

String[] childNames = {"Child1", "Child2", "Child3", "Child4", "Child5"};

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

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

for (int mark : marksArray[i]) {

System.out.print(mark + " ");

}

System.out.println();

}

}

public void totalMarks() {

String[] childNames = {"Child1", "Child2", "Child3", "Child4", "Child5"};

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

int total = 0;

for (int mark : marksArray[i]) {

total += mark;

}

System.out.println(childNames[i] + " Total Marks: " + total);

}

}

public void countMarksAbove80() {

String[] childNames = {"Child1", "Child2", "Child3", "Child4", "Child5"};

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

int count = 0;

for (int mark : marksArray[i]) {

if (mark > 80) {

count++;

}

}

System.out.println(childNames[i] + " Marks Above 80: " + (count));

}

}

public void alertLowMarks() {

String[] childNames = {"Child1", "Child2", "Child3", "Child4", "Child5"};

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

System.out.print(childNames[i] + " Low Marks (<30) in subjects: ");

for (int mark : marksArray[i]) {

if (mark < 30) {

System.out.print(mark + " ");

}

}

System.out.println();

}

}

private int getChildIndex(String childName) {

String[] childNames = {"Child1", "Child2", "Child3", "Child4", "Child5"};

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

if (childNames[i].equals(childName)) {

return i;

}

}

return -1;

}

public static void main(String[] args) {

int[] subjectsPerChild = {3, 5, 2, 6, 4};

JaggedArray marksArray = new JaggedArray(subjectsPerChild);

marksArray.inputMarks("Child1", new int[]{75, 80, 90});

marksArray.inputMarks("Child2", new int[]{85, 75, 95, 60, 70});

marksArray.inputMarks("Child3", new int[]{70, 65});

marksArray.inputMarks("Child4", new int[]{55, 40, 75, 90, 80, 85});

marksArray.inputMarks("Child5", new int[]{30, 35, 25, 45});

marksArray.showMarks();

marksArray.totalMarks();

marksArray.countMarksAbove80();

marksArray.alertLowMarks();

}

}

**OUTPUT:**

Child1: 75 80 90

Child2: 85 75 95 60 70

Child3: 70 65

Child4: 55 40 75 90 80 85

Child5: 30 35 25 45

Child1 Total Marks: 245

Child2 Total Marks: 385

Child3 Total Marks: 135

Child4 Total Marks: 425

Child5 Total Marks: 135

Child1 Marks Above 80: 1

Child2 Marks Above 80: 2

Child3 Marks Above 80: 0

Child4 Marks Above 80: 2

Child5 Marks Above 80: 0

Child1 Low Marks (<30) in subjects: None

Child2 Low Marks (<30) in subjects: None

Child3 Low Marks (<30) in subjects: None

Child4 Low Marks (<30) in subjects: None

Child5 Low Marks (<30) in subjects: 25