| **Experiment No.8** |
| --- |
| **Arrays** |
| Date of Performance:09/08/24 |
| Date of Submission:16/08/24 |

**Aim:-** Implementation of creation, manipulation, and utilization of 2D arrays in programming for efficient data storage and retrieval.

**Objective:-** Implement a code in Java to manage student grades for a school. The application should utilize both one-dimensional (1D) and two-dimensional (2D) arrays to efficiently store and process data. The 1D array will store the names of the students, allowing for easy retrieval and display of student information. The 2D array will be used to store the grades for multiple subjects for each student, to calculate the average grade per student, determine the highest and lowest grades

**Theory:-**Arrays are a fundamental data structure in Java used to store multiple values of the same type in a single variable. Java supports both one-dimensional (1D) and two-dimensional (2D) arrays, which are essential for organizing and manipulating data efficiently.

#### One-Dimensional Arrays (1D Arrays)

One-dimensional arrays, often referred to simply as arrays, are a linear collection of elements. They are essentially a list of values, where each value can be accessed by its index.

* An array can be init in two ways:

1. Initializing at the time of declaration:

dataType[] myArray = {value0, value1, ..., valuek};

1. Dynamic declaration:

dataType[] myArray = new dataType[arraySize];

myArray[index] = value;

#### Two-Dimensional Arrays (2D Arrays)

Two-dimensional arrays are essentially arrays of arrays. They represent a matrix or a table with rows and columns. Each element in a 2D array is accessed using two indices: one for the row and one for the column.

* Two-dimensional array is the simplest form of a multidimensional array. Data of only same data type can be stored in a 2D array.Data in a 2D Array is stored in a tabular manner which can be represented as a matrix.
* A 2D Array can be declared in 2 ways:

1. Intializing at the time of declaration:

dataType[][] myArray = { {valueR1C1, valueR1C2...}, {valueR2C1, valueR2C2...},..}

1. Dynamic declaration:

dataType[][] myArray = new dataType[x][y];

myArray[row\_index][column\_index] = value;

**Code:-**

import java.util.Scanner;

public class StudentsGrade {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter number of students: ");

int numStudents = scanner.nextInt();

System.out.print("Enter number of subjects: ");

int numSubjects = scanner.nextInt();

String[] studentNames = new String[numStudents];

double[][] grades = new double[numStudents][numSubjects];

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

System.out.print("\nEnter name of student " + (i + 1) + ": ");

scanner.nextLine(); // Consume the newline character

studentNames[i] = scanner.nextLine();

System.out.println("Enter grades for " + studentNames[i] + ": ");

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

System.out.print("Grade for subject " + (j + 1) + ": ");

grades[i][j] = scanner.nextDouble();

}

}

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

System.out.println("\nStudent: " + studentNames[i]);

double sum = 0;

double highestGrade = grades[i][0];

double lowestGrade = grades[i][0];

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

sum += grades[i][j];

if (grades[i][j] > highestGrade) {

highestGrade = grades[i][j];

}

if (grades[i][j] < lowestGrade) {

lowestGrade = grades[i][j];

}

}

double averageGrade = sum / numSubjects;

System.out.printf("Average Grade: %.2f\n", averageGrade);

System.out.printf("Highest Grade: %.2f\n", highestGrade);

System.out.printf("Lowest Grade: %.2f\n", lowestGrade);

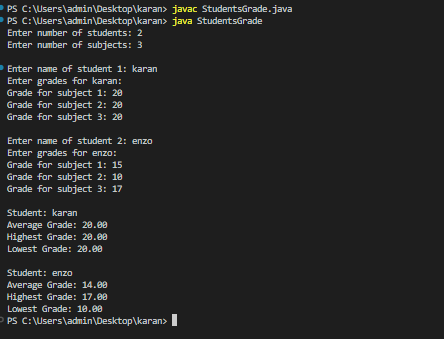
}

scanner.close();

}

}

**Output:-**

****

**Conclusion:-**

A 3D array in Java enables the storage of complex data structures, such as matrices of matrices. It can represent multidimensional data, like 3D models or spatial data, facilitating operations in simulations, graphics, and scientific computations efficiently.