Experiment No. 6
Implement a program on 2D array & strings functions.
Date of Performance:
Date of Submission:

Aim: To use 2D arrays and Strings for solving given problem.

Objective: To use 2D array concept and strings in java to solve real world problem

Theory:

- An array is used to store a fixed-size sequential collection of data of the same type.
- An array can be init in two ways:
  1. Initializing at the time of declaration:
     dataType[] myArray = {value0, value1, ..., valuek};
  2. Dynamic declaration:
     dataType[] myArray = new dataType[arraySize];
     myArray[index] = value;
- 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...},..}
  2. Dynamic declaration:
     dataType[][] myArray = new dataType[x][y];
     myArray[row_index][column_index] = value;

In Java, string is basically an object that represents sequence of char values. An array of characters works same as Java string. **Java String** class provides a lot of methods to perform operations on strings such as compare(), concat(), equals(), split(), length(), replace(), compareTo(), intern(), substring() etc.

1.String literal

To make Java more memory efficient (because no new objects are created if it exists already in the string constant pool).

Example:

String demoString = "GeeksforGeeks";

2. Using new keyword

- String s = new String("Welcome");
- In such a case, JVM will create a new string object in normal (non-pool) heap memory and the literal "Welcome" will be placed in the string constant pool. The variable s will refer to the object in the heap (non-pool)

Example:

String demoString = new String ("GeeksforGeeks");

Code:

1) 1-D ARRAY

```
2) import java.util.Scanner;
3) class Array
4) {
5) public static void main(String[] args)
6)     {
7)     int i, n;
8)     Scanner sc=new Scanner(System.in);
9)     System.out.println("Enter Number of elements in an Array :");
10)    n = sc.nextInt();
11)    int Array[] = new int[n];
12)    System.out.println("Enter the elements  :");
13)    for(i=0;i<n;i++)
14)    {
15)        Array[i]=sc.nextInt();
16)}
17)System.out.println("elements in the array are:");
18)for(i=0;i<n;i++)
19){
20)System.out.println(Array[i]);
21)}
22)}
23)}
```

OUTPUT:-

```
C:\Users\Dnyanesh\Desktop\java>javac Array.java

C:\Users\Dnyanesh\Desktop\java>java Array.java
Enter Number of elements in an Array :
3
Enter the elements  :
6 7 8
elements in the array are:
6
7
8
```

2) 2-D ARRAY

```
import java.util.Scanner;
class Matrix
{
public static void main(String[] args)
    {
    int i,j,r,c;
    Scanner sc=new Scanner(System.in);
    System.out.println("Enter Number of rows :");
    r = sc.nextInt();
        System.out.println("Enter number of columns:");
        c = sc.nextInt();
    int Matrix[][]= new int[r][c];
    System.out.println("Enter the elements  :");
    for(i=0;i<r;i++)
    {
         for(j=0;j<c;j++)
{
        Matrix[i][j]=sc.nextInt();
}
System.out.println(" ");
}

System.out.println("elements in the array are:");
for(i=0;i<r;i++)
{
for(j=0;j<c;j++)
{
System.out.print(Matrix[i][j] + " ");
}
System.out.println(" ");
}
```

```
}
}
```

OUTPUT:-

```
C:\Users\Dnyanesh\Desktop\java>java mATRIX.java
Enter Number of rows :
2
Enter number of columns:
2
Enter the elements  :
2 3 4 5


elements in the array are:
2 3
4 5
```

3. STRINGS

```
class Tp
{
int id;
String name;
Tp(String s,int i)
{id=i;
name=s;
}
public static void main(String args[])
{
Tp tp=new Tp("Dnyanesh",22);
System.out.println("Emp name:\t"+ tp.name+"\tid"+tp.id);
}
}
```

OUTPUT:-

```
C:\Users\Dnyanesh\Desktop\java>javac Tp.java

C:\Users\Dnyanesh\Desktop\java>java Tp.java
Emp name:       Dnyanesh        id22

C:\Users\Dnyanesh\Desktop\java>
```

Conclusion:

Comment on how you have used the concept of string and 2D array.

Use of string: The **Tp** class has a constructor with two parameters, **String s** and **int i**. In this context, the **String s** parameter is used to initialize the **name** attribute of a **Tp** object, and the **int i** parameter is used to initialize the **id** attribute.
In the **main** method, a **Tp** object named **tp** is created. The constructor is called with the string "Dnyanesh" and the integer 22 as arguments. This sets the **name** attribute of the **tp** object to "Dnyanesh" and the **id** attribute to 22.
Finally, a message is printed to the console using **System.out.println()**. The message includes the values of the **name** and **id** attributes of the **tp** object, which are concatenated into the output string. The **name** attribute is accessed using **tp.name**, and the **id** attribute is accessed using **tp.id**. Both attributes are used in the output string to display the employee's name and ID.
Use of 2D array : The program first uses the **Scanner** class to read input from the user. It prompts the user to enter the number of rows and columns for the matrix, and these values are stored in the variables **r** and **c**, respectively.
An integer 2D array is declared and initialized with the dimensions specified by the user's input. The array is defined as **int Matrix[][] = new int[r][c]**, creating a matrix with **r** rows and **c** columns.
The program then prompts the user to enter the elements of the matrix. It uses nested **for** loops to iterate over each cell in the 2D array. The outer loop iterates over rows (from 0 to **r-1**), and the inner loop iterates over columns (from 0 to **c-1**). The user-provided values are read with **sc.nextInt()** and stored in the corresponding cell of the 2D array **Matrix[i][j]**.
After the user enters all the elements, the program displays the elements stored in the 2D array. It uses another set of nested **for** loops to iterate over the entire array, printing each element with **System.out.print()** and separating them with spaces. A newline character is added after each row to format the output.