Lab Assignment-05

ROLL: 2005535 | NAME: SAHIL SINGH | DATE: 10/02/22

QUES 1 : Write a program which will overload the area () method and display the area of a circle, triangle and square as per user choice and user entered dimensions. Consider the class as Shape.

SOLUTION:

import java.util.Scanner;

class Area {

*void* area(*int* *a*) {

        System.out.println("\nArea of square :" + *a* \* *a*);

}

*void* area(*float* *a*) {

        System.out.println("\nArea of circle :" + 3.14 \* *a* \* *a*);

}

*void* area(*int* *a*, *int* *b*, *int* *c*) {

*double* temp = (*a* + *b* + *c*);

*double* s = temp / 2;

*double* triarea = Math.sqrt(s \* (s - *a*) \* (s - *b*) \* (s - *c*));

        System.out.println("\nArea of triangle: " + triarea);

}

    public static *void* main(String *args*[]) {

        Scanner sc = new Scanner(System.in);

        Area d = new Area();

        System.out.print("\nShapes available: \n1) Square \n2) Triangle \n3) Circle \nEnter your choice: ");

*int* choice = sc.nextInt();

        switch (choice) {

            case 1:

                System.out.print("\nEnter the side of square: ");

*int* a = sc.nextInt();

                d.area(a);

                break;

            case 2:

                System.out.print("\nEnter first side: ");

*int* x = sc.nextInt();

                System.out.print("\nEnter second side: ");

*int* y = sc.nextInt();

                System.out.print("\nEnter third side: ");

*int* z = sc.nextInt();

                d.area(x, y, z);

                break;

            case 3:

                System.out.print("\nEnter the radius: ");

*int* r = sc.nextInt();

                d.area(r);

                break;

            default:

                System.out.println("Invalid choice!!!");

        }

        sc.close();

    }

}

OUTPUT:

Shapes available:

1) *Square*

2) *Triangle*

3) Circle

Enter your choice: 1

Enter the side of square: 5

*Area* of square :25

QUES 2: Create a class called 'Matrix' containing constructor that initializes the number of rows and number of columns of a new Matrix object. The Matrix class has the following information: number of rows of matrix, number of columns of matrix, elements of matrix in the form of 2D array. The Matrix class has methods for each of the following:

1 - get the number of rows

2 - get the number of columns

3 - set the elements of the matrix at given position (i,j)

4 - adding two matrices. If the matrices are not addable, "Matrices cannot be added" will be displayed.

5 - multiplying the two matrices

SOLUTION:

import java.util.\*;

class Matrix {

    private *double*[][] mat;

*int* row, column;

    // constructor

    Matrix() {

        row = 0;

        column = 0;

}

    // constructor

    Matrix(*int* *r*, *int* *c*) {

        row = *r*;

        column = *c*;

        mat = new *double*[row][column];

}

    // function to accept Matrix values from user

    public *void* get\_input() {

        Scanner s = new Scanner(System.in);

*int* i = 0, j = 0;

        System.out.println("Enter the Matrix elements(row-wise)");

        for (i = 0; i < row; i++) {

            for (j = 0; j < column; j++) {

                mat[i][j] = s.nextDouble();

            }

        }

}

    // function to print Matrix

    public *void* print\_Matrix() {

*int* i = 0, j = 0;

        System.out.println("The Matrix is: ");

        for (i = 0; i < row; i++) {

            System.out.println("");

            for (j = 0; j < column; j++) {

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

            }

        }

}

    // function to add 2 matrices

    public Matrix add(Matrix *m1*) {

        Matrix ans = new Matrix(row, column);

*int* i = 0, j = 0;

        if (this.row != *m1*.row || this.column != *m1*.column) {

            System.out.println("Matrices cannot be added!");

        } else {

            for (i = 0; i < row; i++) {

                for (j = 0; j < column; j++) {

                    ans.mat[i][j] = this.mat[i][j] + *m1*.mat[i][j];

                }

            }

        }

        return ans;

}

    // function to multiply 2 matrices

    public Matrix mul(Matrix *m1*) {

        Matrix ans = new Matrix(row, column);

*int* i = 0, j = 0;

        if (this.column != *m1*.row) {

            System.out.println("Matrices cannot be added!");

        } else {

            for (i = 0; i < row; i++) {

                for (j = 0; j < column; j++) {

                    ans.mat[i][j] += this.mat[i][j] \* *m1*.mat[j][i];

                }

            }

        }

        return ans;

    }

}

class Operations {

private static Matrix m1, m2, ans;

    public static *void* main(String *args*[]) {

*int* i = 0, j = 0, r = 0, c = 0;

        Scanner s = new Scanner(System.in);

        // get the no. of rows and columns of the Matrix from the user

        System.out.println("\nEnter no. of rows:");

        r = s.nextInt();

        System.out.println("Enter no. of columns:");

        c = s.nextInt();

        // accept the first Matrix

        System.out.println("\nEnter the first Matrix: ");

        m1 = new Matrix(r, c);

        m1.get\_input();

        // get the no. of rows and columns of the Matrix from the user

        System.out.println("\nEnter no. of rows:");

        r = s.nextInt();

        System.out.println("Enter no. of columns:");

        c = s.nextInt();

        // accept the second Matrix

        System.out.println("\nEnter the second Matrix: ");

        m2 = new Matrix(r, c);

        m2.get\_input();

        // Add the 2 matrices and print the result

        ans = m1.add(m2);

        System.out.println("\nAddition of the two matrices is: ");

        ans.print\_Matrix();

        // Multiply the 2 matrices and print the result

        ans = m1.mul(m2);

        System.out.println("\nMultiplication of the two matrices is: ");

        ans.print\_Matrix();

        s.close();

    }

}

OUTPUT:

*Enter* no. of rows:

3

Enter no. of columns:

3

Enter the first Matrix:

*Enter* the Matrix elements(row-wise)

1 2 3

4 5 6

7 8 9

*Enter* no. of rows:

3

*Enter* no. of columns:

3

*Enter* the second Matrix:

*Enter* the Matrix elements(row-wise)

9 8 7

6 5 4

3 2 1

*Addition* of the two matrices is:

*The* *Matrix* is:

   10.0   10.0   10.0

   10.0   10.0   10.0

   10.0   10.0   10.0

*Multiplication* of the two matrices is:

*The* *Matrix* is:

   9.0   12.0   9.0

   32.0   25.0   12.0

   49.0   32.0   9.0

**-----------------------------------------------------------------------------------------**