Greg Witt

CS202

Matrix\_Lab1

Matrix Class

=================================

package matrixlabclient;

public class Matrix {

private int r;

private int c;

private final int MAX = 100;

private int[][] data;

public Matrix(){} //no args constructor

//method to create an array with int args

public Matrix(int row, int col){

r = row; //number of rows

c = col; // number of columns

data = new int[r][c]; //store in new 2d array

for (int i = 0; i < this.r; i++) { //go through the array’s rows

for (int j = 0; j < this.c; j++) { // go through the columns

data[i][j] = (int)(Math.random() \* MAX + 1); // filling with random numbers

}//j

}//i

}//Matrix //constructor

public boolean matrixTest(Matrix m1){

if(this.r != m1.r || this.c != m1.c){

return false;

}//if

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

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

if(this.data[i][j] != m1.data[i][j]){

return false;

}//if

}//for j

}// for i

return true;

}//matrixTest

public Matrix copyMatrix(){

Matrix copyMatrix = new Matrix(this.r,this.c);

for (int i = 0; i < this.r; i++) {

System.arraycopy(this.data[i],0,copyMatrix.data[i],0,c);

}//for

return copyMatrix;

}//copyMatrix

public Matrix multiplyMatrix(Matrix m2)throws IllegalArgumentException{

Matrix m3 = new Matrix(this.r, m2.c);

if(this.c != m2.r){

throw new IllegalArgumentException("Error invalid Argument");

}//if

for (int i = 0; i < this.r; i++) {

for (int j = 0; j < m2.c; j++) {

for (int k = 0; k < m2.r; k++) {

m3.data[i][j] += this.data[i][k]\*m2.data[k][j];

}//k

}//j

}//i

return m3;

}//multiplyMatrix

public Matrix scalarMultiply(int multi){

Matrix m = new Matrix(this.r, this.c);

for (int i = 0; i < this.r; i++) {

for (int j = 0; j < this.c; j++) {

m.data[i][j] = this.data[i][j] \* multi; // multiply the int

}//j

}//i

return m;

} //scalarMultiply

@Override

public String toString(){

String output = "";

for (int i = 0; i < this.r; i++) {

for (int j = 0; j < this.c; j++) {

output += data[i][j] + "\t";

}//j

output += "\n";

}//i

return output;

}//toString

}//class

Client:

==============================

package matrixlabclient;

public class MatrixLabClient {

public static void main(String[] args) {

//testing constructor

Matrix hung = new Matrix(3,3);

Matrix test1 = new Matrix(3,3);

Matrix m4 = new Matrix(3,3);

Matrix equalTest2 = new Matrix(2,3);

//display

System.out.println(hung);

//System.out.println(test1);

//testing scalar method

hung = hung.scalarMultiply(7);

System.out.println(hung);

//testing multiplyMatrix

hung = hung.multiplyMatrix(test1);

System.out.println(hung);

//testing equals method

boolean m2 = hung.matrixTest(equalTest2);

System.out.println("Result of Matrix test:");

System.out.println(m2);

boolean m3 = hung.matrixTest(hung);

System.out.println("Result of Matrix test:");

System.out.println(m3);

//testing copyMatrix Method

System.out.println("Test 1:");

System.out.println(test1);

System.out.println("Matrix 4:");

System.out.println(m4);

System.out.println("copying test1 to Matrix 4:");

m4 = test1.copyMatrix();

System.out.println(m4);

}//main

}//class

Output:

run:

8 25 18

56 58 50

36 95 20

56 175 126

392 406 350

252 665 140

13045 17926 20034

47893 53920 66585

46873 43672 48766

Result of Matrix test:

false

Result of Matrix test:

true

Test 1:

64 31 58

43 38 33

15 75 87

Matrix 4:

28 51 95

44 9 13

81 66 71

copying test1 to Matrix 4:

64 31 58

43 38 33

15 75 87

13045 17926 20034

47893 53920 66585

46873 43672 48766

BUILD SUCCESSFUL (total time: 0 seconds)