

Joao Paulo Dos Santos Ferreira
Fundamentals of Programming I – CSIT 111_04
Professor Jiayin Wang
November 30, 2017

Lab 09 Report

Exercise 1: Magic Squares

```
// *****  
// Square.java  
//  
// Define a Square class with methods to create and read in  
// info for a square matrix and to compute the sum of a row,  
// a col, either diagonal, and whether it is magic.  
//  
// *****  
import java.util.Scanner;  
public class Square  
{  
    int[][] square;  
    int x;  
    int i;  
    int j;  
  
    //-----  
    //create new square of given size  
    //-----  
    public Square(int size)  
    {  
        x = size;  
        int col = x, row = x;  
        square = new int[row][col];  
    }  
  
    //-----  
    //return the sum of the values in the given row  
    //-----  
    public int sumRow(int row)  
    {  
        int sumRow = 0;  
        for (i=0; i < x; i++) {  
            sumRow = sumRow + square[row][i];  
        }  
        return sumRow;  
    }  
}
```

```

//-----
//return the sum of the values in the given column
//-----
public int sumCol(int col)
{
    int sumCol = 0;
    for (i=0; i < x; i++) {
        sumCol = sumCol + square[i][col];
    }
    return sumCol;
}

//-----
//return the sum of the values in the main diagonal
//-----
public int sumMainDiag()
{
    int sumMD = 0;
    for (i=0; i < x; i++ ) {
        sumMD = sumMD + square[i][i];
    }
    return sumMD;
}

//-----
//return the sum of the values in the other ("reverse") diagonal
//-----
public int sumOtherDiag()
{
    int sumOD = 0;
    for (i=0; i < x; i++) {
        for (j=0; j < x; j++) {
            if (i+j == x-1) {
                sumOD = sumOD + square[i][j];
            }
        }
    }
    return sumOD;
}

//-----
//return true if the square is magic (all rows, cols, and diags have
//same sum), false otherwise
//-----
public boolean magic()
{
    int w;
    int z;
    boolean y = true;
    for (w=0; w < x; w++) {
        for (z = 0; z < x; z++) {

```

```

        if (sumCol(w) == sumRow(z) && sumOtherDiag() == sumMainDiag()
            && sumRow(w) == sumMainDiag())
        {
            y = true;
        } else {
            y = false;
        }
    }
}
return y;
}

//-----
//read info into the square from the input stream associated with the
//Scanner parameter
//-----
public void readSquare(Scanner scan)
{
    for (int row = 0; row < square.length; row++)
        for (int col = 0; col < square.length; col ++)
            square[row][col] = scan.nextInt();
}

//-----
//print the contents of the square, neatly formatted
//-----
public void printSquare()
{
    for (int row = 0; row < square.length; row++) {
        System.out.println();
        for (int col = 0; col < square.length; col ++) {
            if (square[row][col] < 10 && square[row][col] >= 0) {
                System.out.print(square[row][col] + " ");
            } else if (square[row][col] >= 10) {
                System.out.print(square[row][col] + " ");
            }
        }
    }
}
}
}

```

```

// *****
// SquareTest.java
//
// Uses the Square class to read in square data and tell if
// each square is magic.
//
// *****
import java.util.Scanner;
import java.io.*;

public class SquareTest
{
    public static void main(String[] args) throws IOException
    {
        int i=0;
        int j=0;

        Scanner scan = new Scanner(new File("magicData"));
        int count = 1; //count which square we're on
        int size = scan.nextInt(); //size of next square
        //Expecting -1 at bottom of input file
        while (size != -1)
        {

            //create a new Square of the given size

            Square ss = new Square(size);

            //call its read method to read the values of the square

            ss.readSquare(scan);

            System.out.println("\n***** Square " + count
+ " *****");

            //print the square
            ss.printSquare();
            System.out.println("\n");

            //print the sums of its rows
            for (i=0; i < size; i++) {
                System.out.println("Sum of row " + (i+1) + " is: "
+ ss.sumRow(i));
            }

            //print the sums of its columns
            System.out.println();
            for (i=0; i < size; i++) {
                System.out.println("Sum of column " + (i+1) + " is: "
+ ss.sumCol(i));
            }
        }
    }
}

```

```

        //print the sum of the main diagonal
        System.out.println("\nSum of main Diagonal is: "
+ ss.sumMainDiag());

        //print the sum of the other diagonal
        System.out.println("\nSum of other Diagonal is: "
+ ss.sumOtherDiag());

        //determine and print whether it is a magic square
        boolean magic = ss.magic();
        if (magic == true) {
            System.out.println("\n**This is a Magic Square**");
        } else {
            System.out.println("\nThis is not a magic square");
        }

        //get size of next square
        size = scan.nextInt();
        count++;
    }
}

```