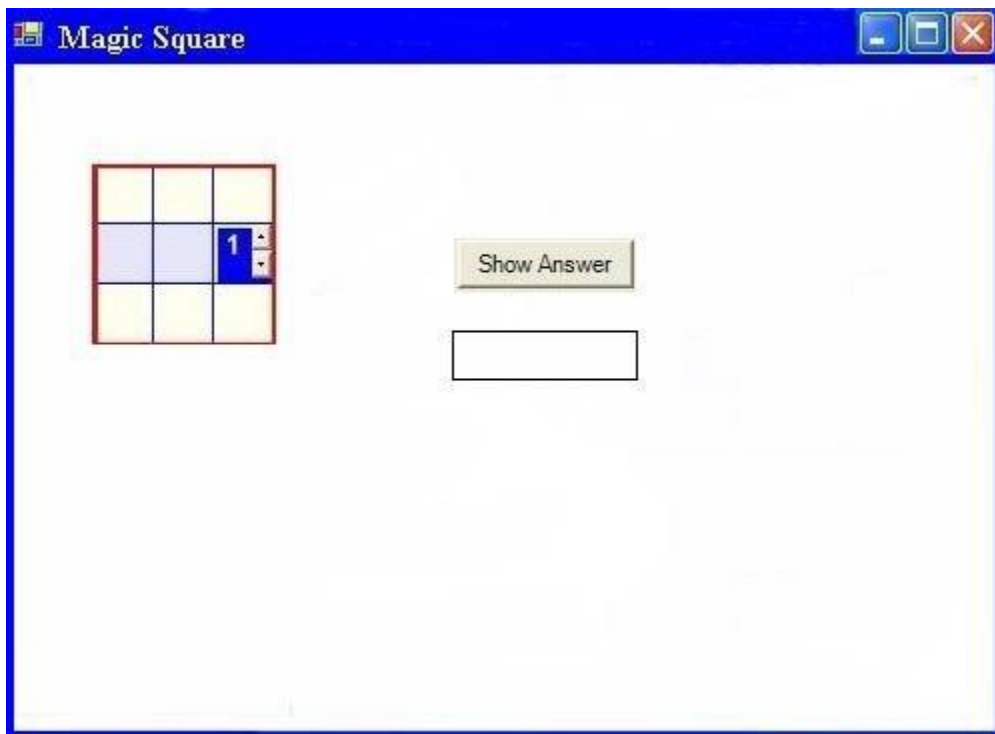


U18CO018
Shubham Shekhaliya
Assignment-3
Subject – ST

Write a java program to enter an integer number 'n'. Create a grid of size $n \times n$. Enter the value of each cell using a dropdown list. Check whether the created grid is a magic square or not. Display the result into the given text box by clicking on "Show Answer" button.

Magic Square:

A square matrix is said to be a Magic Square, if the sum of each row, each column and each diagonal is same.



Code:-

```
package com.company;

import javax.swing.*;
import javax.swing.table.AbstractTableModel;
import javax.swing.table.TableModel;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.util.Scanner;

public class Assignment3 {
    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);
        System.out.print("Enter the Size of Matrix : ");
        int n = sc.nextInt();

        // Todo: initialize matrix first time
        // Name of each column
        String[] columns = new String[n];
        //actual data for the table in a 2d array
        Integer[][] data = new Integer[n][n];
        for (int i = 0; i < n; i++) {
            columns[i] = String.valueOf(i + 1);
            for (int j = 0; j < n; j++) {
                data[i][j] = 0;
            }
        }

        // Todo: Create Editable table
        TableModel model = new EditableTableModel(columns, data);
        // Todo: Create JTable
        JTable table = new JTable(model);
        table.setRowHeight(30);
        // todo: add DropDown List
        JComboBox comboBox = new JComboBox();
        for (int i = 0; i <= n * n; i++) {
            comboBox.addItem(i);
        }
        DefaultCellEditor editor = new DefaultCellEditor(comboBox);
        for (int i = 0; i < n; i++) {
            // added dropdown for each column
            table.getColumnModel().getColumn(i).setCellEditor(editor);
        }
    }
}
```

```

//Todo: TextField to show answer
JTextField tf = new JTextField();
tf.setBounds(200, 200, 200, 30);

//Todo: button to check matrix
JButton button = new JButton("Show Answer");
button.setBounds(50, 200, 125, 30);
//Todo: Action Listener to check matrix
button.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        if (CheckMatrix(data, n)) {
            tf.setText("Magic Matrix.");
        } else {
            tf.setText("Not Magic Matrix");
        }
    }
});

// Todo: Frame Creation
JFrame frame = new JFrame();
frame.setSize(600, 400);
// Todo: create scrollpane to add table in frame
JScrollPane scrollpane = new JScrollPane(table);

// Todo: add button, textField and table in frame
frame.add(button);
frame.add(tf);
frame.add(scrollpane);

// Todo: set constraint of frame
frame.setTitle("Magic Matrix");
frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
frame.pack();
frame.setVisible(true);
}

// extend abstractTableModel to user for our purpose
static class EditableTableModel extends AbstractTableModel {
    String[] columnTitles;
    Integer[][] dataEntries;

    public EditableTableModel(String[] columnTitles, Integer[][] dataEntries)
{
        this.columnTitles = columnTitles;

```

```

        this.dataEntries = dataEntries;
    }

    public int getRowCount() {
        return dataEntries.length;
    }

    public int getColumnCount() {
        return columnTitles.length;
    }

    public Integer getValueAt(int row, int column) {
        return dataEntries[row][column];
    }

    public String getColumnName(int column) {
        return columnTitles[column];
    }

    public Class getColumnClass(int column) {
        return getValueAt(0, column).getClass();
    }

    public boolean isCellEditable(int row, int column) {
        return true;
    }

    @Override
    public void setValueAt(Object aValue, int rowIndex, int columnIndex) {
        dataEntries[rowIndex][columnIndex] = (Integer) aValue;
    }
}

static boolean CheckMatrix(Integer mat[][], int N) {
    // todo: calculate the sum of the prime diagonal
    int sum = 0, sum2 = 0;
    for (int i = 0; i < N; i++)
        sum = sum + mat[i][i];

    // todo: sum of secondary diagonal
    for (int i = 0; i < N; i++)
        sum2 = sum2 + mat[i][N - 1 - i];
    // todo: both sum are same or not
    if (sum != sum2)
        return false;
}

```

```

//todo: sums of each Rows
for (int i = 0; i < N; i++) {
    int rowSum = 0;
    for (int j = 0; j < N; j++)
        rowSum += mat[i][j];
    // todo: if every row's sum is equal to prime diagonal sum
    if (rowSum != sum)
        return false;
}

//todo: sums of each Columns
for (int i = 0; i < N; i++) {
    int colSum = 0;
    for (int j = 0; j < N; j++)
        colSum += mat[j][i];
    // todo: if every column's sum is equal to prime diagonal sum
    if (sum != colSum)
        return false;
}
return true;
}
}

```

Output:-

The screenshot shows an IDE with a Java project named 'Code1'. The file 'Assignment3.java' is open, showing a program that generates a Magic Matrix. The code includes imports for Swing, AbstractTableModel, TableModel, and Scanner. It defines a class 'Assignment3' with a 'main' method that prompts the user to enter the size of the matrix. The output window shows the following matrix:

1	2	3
1	2	3
2	7	6
9	5	1

The output window also shows a 'Show Answer' button and a text field labeled 'Magic Matrix:'. The status bar at the bottom indicates the build completed successfully in 2 s 244 ms (a minute ago).

Code1 - Assignment3.java

Code1 src \ com \ company \ Assignment3

Project

- Code1
- src
- com
- company
- Assignment2
- Assignment3
- Main.java
- Minisweeper
- q1
- q2
- q3
- q4
- q5
- q6
- q7
- q8
- q9
- Code1.mim
- External Libraries
- Scratches and Consoles

Assignment3.java

```
1 package com.company;
2
3 import javax.swing.*;
4 import javax.swing.table.AbstractTableModel;
5 import javax.swing.table.TableModel;
6 import java.awt.event.*;
7 import java.awt.*;
8 import java.util.*;
9
10 public class Assignment3 {
11     public static void main(String[] args) {
12         Scanner scanner = new Scanner(System.in);
13         int n = scanner.nextInt();
14
15         // TODO: Create Editable table
16         String[] columns = {"1", "2", "3"};
17         Integer[][] data = {{1, 2, 3}, {2, 7, 5}, {3, 6, 1}};
18         TableModel model = new EditableTableModel(columns, data);
19         // TODO: Create JTable
20         JTable table = new JTable(model);
21         table.setRowHeight(30);
22     }
23 }
```

Magic Matrix

1	2	3
1	2	3
2	7	5
3	6	1

Show Answer

Not Magic Matrix

Run

Assignment3

"C:\Program Files\Java\jdk-14.0.2\bin\java.exe"

Enter the Size of Matrix : 3

Build completed successfully in 2 s 244 ms (2 minutes ago)

3.1 CRUF UTF-8 4 spaces

ENG 1:03 PM 1/28/2021