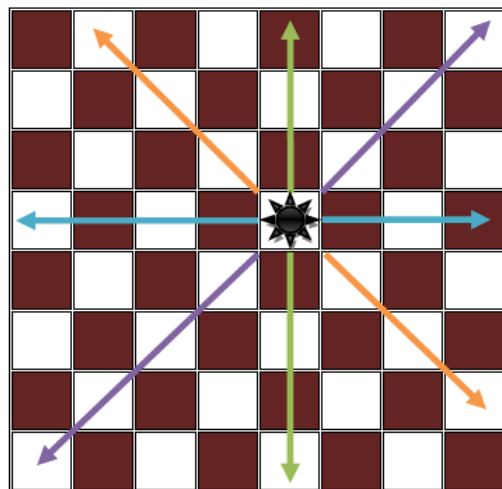


N-Queen Documentation

1 PROJECT DESCRIPTION

The N Queen is the problem of placing N chess queens on an $N \times N$ chessboard so that no two queens attack each other by being in the same row, column or diagonal.

In the chess game Queen is very powerful piece that can approach with multi-directional aspects of horizontal, vertical and diagonal in order that a handful corners remain protected for opponent pieces.



The conventional N-Queen encompasses the assignments of fixed number of Queens over $n \times n$ chessboard in a way no queen access to other. In brief, same row (horizontally), column (vertically) or diagonal (Crossways) should not be common between two Queens.

2 CODE DOCUMENTATION

1. BOARD

Set Board function fills 2D array with zeros

```
public class Board {  
    public Board() {  
  
    }  
  
    public int[][] setBoard(int NQueen){  
        int [][] board = new int[NQueen][NQueen];  
        for(int i = 0; i < NQueen; i++){  
            for(int j = 0; j < NQueen; j++){  
                board[i][j] = 0;  
            }  
        }  
        return board;  
    }  
}
```

2. IS SAFE

Check safety of each Queen

```
public class IsSafe {  
    public static boolean isSafe(int board[][], int row, int column){  
  
        for(int i = 0; i < column; i++){  
            if(board[row][i] == 1)  
                return false;  
        }  
  
        for(int i = row, j = column; i >= 0 && j >= 0; i--, j--){  
            if(board[i][j] == 1)  
                return false;  
        }  
  
        for(int i = row, j = column; i < board.length && j >= 0; i++, j--){  
            if(board[i][j] == 1)  
                return false;  
        }  
        return true;  
    }  
}
```

3. SOLVE N-QUEEN

Run recursively to put 1 or 0 on the board

```
public static boolean solveNQueen(int board[][], int column){  
    if (column >= board.length){  
        return true;  
    }  
    for(int i = 0 ; i < board.length; i++){  
        if(isSafe(board,i, column)){  
            board[i][column] = 1;  
            if (solveNQueen(board, column+1))  
                return true;  
            board[i][column] = 0;  
        }  
    }  
    return false;  
}
```

4. N-QUEEN FORM

Empty form has labels represent the Queen (1: there is a Queen – 0: there is no Queen)

5. START

Read the number of Queens and start a thread to run labels on board

```
public static void main(String args[]) {
    /* Set the Nimbus look and feel */
    Look and feel setting code (optional)

    /* Create and display the form */
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new Start().setVisible(true);
        }
    });
}

public static void fun(int n){
    Board b = new Board();
    Font f = new Font("serif", Font.PLAIN, 24);
    int[][] BOARD = b.setBoard(n);
    new Thread(new Runnable() {
        @Override
        public void run() {
            IsSafe.solveNQueen(BOARD, 0);
            NQueenForm p = new NQueenForm();
            p.setLayout(new GridLayout(n,n));
            p.setFont(new java.awt.Font("Tahoma", 0, 24));
            for (int i = 0 ; i < n ; i++) {
                for (int j = 0; j < n; j++) {
                    JLabel label = new JLabel(" " + BOARD[i][j],JLabel.CENTER);
                    p.add(label).setFont(f);
                    Border b = BorderFactory.createLineBorder(Color.BLACK, 5);
                    label.setBorder(b);
                    p.setVisible(true);
                }
            }
        }
    }).start();
}
```

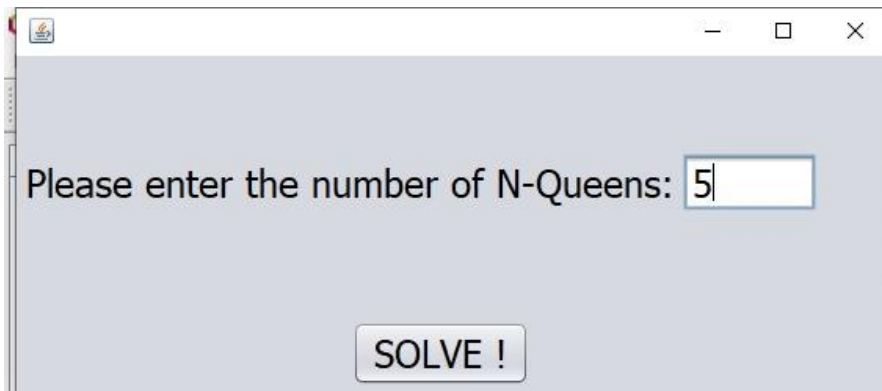
6. GUI

Please enter the number of N-Queens:

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    String N = jTextField2.getText();
    if (N.equals("")) {
        JOptionPane.showMessageDialog(null, "Please enter a number !");
    } else {
        int n = Integer.parseInt(N);
        fun(n);
        this.dispose();
    }
}

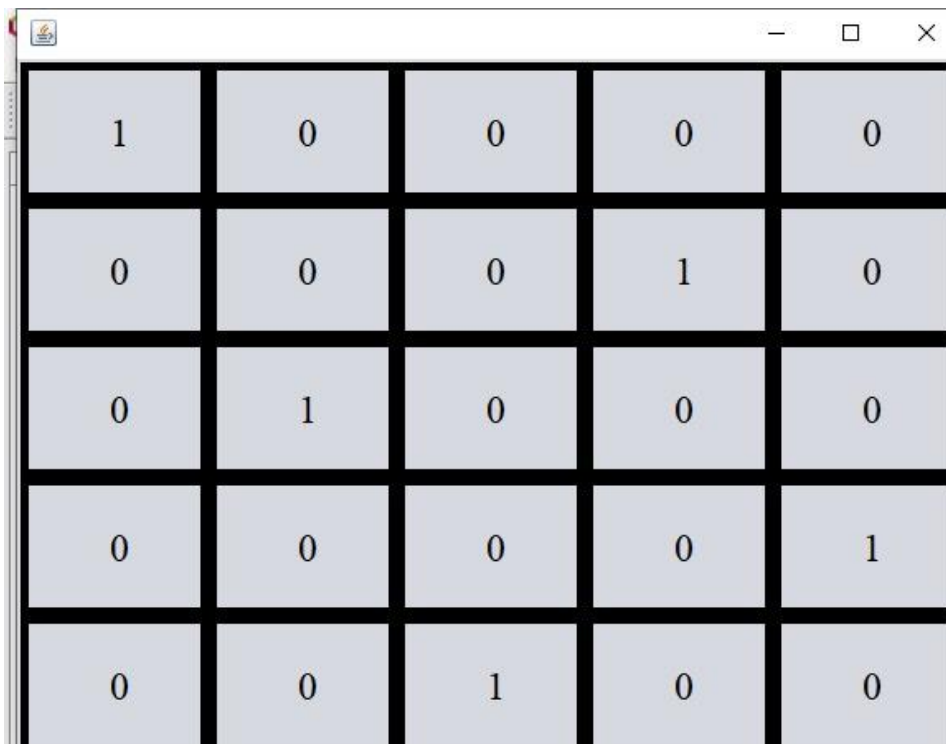
private void jTextField2KeyTyped(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
    char c = evt.getKeyChar();
    if (!Character.isDigit(c)) {
        evt.consume();
    }
}
```

3 OUTPUT



Please enter the number of N-Queens:

SOLVE !



1	0	0	0	0
0	0	0	1	0
0	1	0	0	0
0	0	0	0	1
0	0	1	0	0

4 TEAM MEMBERS ROLE

- | | |
|---------------------------------|-----------|
| • Abdallah Roshdy | 201900461 |
| Solve N-Queen | |
| • Mohamed Said | 201900685 |
| Is Safe function | |
| • Abdelmoneim Faisal | 201900468 |
| Board function – GUI | |
| • Abdallah Assem | 201900463 |
| Is Safe function | |
| • Mohamed Tarek | 201900696 |
| Thread creation - Documentation | |
| • Karim Walid | 201900585 |
| Thread creation | |