**8 QUEENS PROGRAM DOCUMENTATION**

EightQueens.java

TableComponents.java

ViewerFrame.java

Location.java

1. **Location.java:** This is a class for creating the instances of the boxes that we place the queens on the chess table. It helps the program to keep the track of each queen, and also each Location instance is stored in an ArrayList in order to keep track of the moves.

It has two methods called getRow(), for returning the attribute “int row”, and getColumn(), for returning the attribute “int column”.

1. **TableComponents.java:** This is the class, which contains the algorithm for solving the 8 Queens puzzle. The chessboard is represented as a two-dimensional integer array, boxes[][], which has all 0 elements at the beginning.

There is a one-dimensional ArrayList for the location instances called “queens” for tracking the moves, and backtracking process. There is also a three-dimensional ArrayList called combinations, which records all the combinations.

The first method is called isSafe(int cx, int cy), which returns a Boolean variable after applying horizontal, vertical, and cross check in case of any threats from other queens to the current location.

The second method is queenDraw(int x, int y), which is a recursive method. Two parameters are the coordinates of the current location. There 5 possible outcomes of this method.

* 1. If y=1 and x= 8, it means the location came back to the end of the second column, which means computer has tried all possible combinations, and there isn’t any valid one left. So the program ends here.
  2. If y = 8, it means computer has successfully reached to the last column, and created a valid combination of 8 queens. Now, it saves this combination, backtracks to the previous position, and recurs in order to find another valid combination. The combination is also printed on the console if this condition is met.
  3. If x=8, it means the computer have reached to the end of a column and hasn’t find any valid combinations. So the computer backtracks to the previous position, removes the last move from the “queens” list, and recurs.
  4. If isSafe(int cx, int cy) returns true, it means the current location is valid for placing a new queen. So the method places the queen on the “boxes” array, the Location is saved to the “queens” list and programs moves to the next column by setting x to 0, and increment the y.
  5. Finally, if none of these conditions are met, the program passes to the next row in the same column, and recurs.

The last method is printQueen(), which simply uses a nested for loop to create the String game, which will represent each combination in the console.

1. **ViewerFrame.java:**
2. **EightQueens.java:**