模版·回溯法(N皇后问题)

```
public class NQueens{

    private static int[][] board;
    private static int[] cols;
    private static int count=0;
    public static int NQueens=8;


    public static void solveNQueens(int n){
        board=new int[n][n];
        cols=new int[n];

        backtrack(0,n);

        System.out.println("解法总数： "+count);
    }

    private static void backtrack(int row,int n){
        if(row==n){
            printSolution(n);
            count++;
            return;
        }

        for(int col=0;col<n;col++){
            if(isValid(row,col)){
                board[row][col]=1;
                cols[row]=col;

                backtrack(row+1,n);
```

```
                board[row][col]=0;
            }
        }
    }

    private static boolean isValid(int row,int col){
        for(int i=0;i<row;i++){
            if(cols[i]==col){
                return false;
            }
        }

        for(int i=row-1,j=col-1;i>=0&&j>=0;i--,j--){
            if(board[i][j]==1){
                return false;
            }
        }

        for(int i=row-1,j=col+1;i>=0&&j<NQueens;i--,j++){
            if(board[i][j]==1){
                return false;
            }
        }

        return true;
    }

    private static void printSolution(int n){
        System.out.println("解法"+(count+1)+":");
        for(int i=0;i<n;i++){
            for(int j=0;j<n;j++){
                System.out.print(board[i][j]==1?"Q":".");
```

```
                }
                System.out.println();
            }
            System.out.println();
        }

        public static void main(String[] args){
            solveNQueens(NQueens);
        }
}
```