File - C:\aprendizado\java\nélioUdemy\chess-system\src\chess\pieces\Rook.java

```java
package chess.pieces;

import boardgame.Board;
import boardgame.Piece;
import boardgame.Position;
import chess.ChessPiece;
import chess.Color;

import java.util.Scanner;

public class Rook extends ChessPiece {

    public Rook(Board board, Color color) {
        super(board, color);
    }

    @Override
    public String toString() {
        return "R";
    }

    public boolean[][] possibleMoves(Position position){
        boolean[][] mat = new boolean[getBoard().getRows()][getBoard().getColumns()];
        Position p = new Position(0,0);

        // above
        p.setValues(position.getRow()-1, position.getColumn());
        while (p.getRow() >= 0 && !getBoard().thereIsAPiece(p) ){
            mat[p.getRow()][p.getColumn()] = true;
            p.setRow(p.getRow()-1);
        }
        if (getBoard().positionExists(p) && isThereOpponentPiece(p)){
            mat[p.getRow()][p.getColumn()] = true;
        }

        // left
        p.setValues(position.getRow(), position.getColumn() -1);
        while (p.getColumn() >= 0 && !getBoard().thereIsAPiece(p) ){
            mat[p.getRow()][p.getColumn()] = true;
            p.setColumn(p.getColumn()-1);
        }
        if (getBoard().positionExists(p) && isThereOpponentPiece(p)){
            mat[p.getRow()][p.getColumn()] = true;
        }

        // below
        p.setValues(position.getRow()+1, position.getColumn());
        while (p.getRow() < getBoard().getRows() && !getBoard().thereIsAPiece(p) ){
            mat[p.getRow()][p.getColumn()] = true;
            p.setRow(p.getRow()+1);
        }
        if (getBoard().positionExists(p) && isThereOpponentPiece(p)){
            mat[p.getRow()][p.getColumn()] = true;
        }

        // right
        p.setValues(position.getRow(), position.getColumn() +1);
        while (p.getColumn() < getBoard().getColumns() && !getBoard().thereIsAPiece(p) ){
            mat[p.getRow()][p.getColumn()] = true;
            p.setColumn(p.getColumn()+1);
        }
        if (getBoard().positionExists(p) && isThereOpponentPiece(p)){
            mat[p.getRow()][p.getColumn()] = true;
        }

        return mat;
    }


    @Override
    public boolean[][] possibleMoves() {
        return possibleMoves(position);
    }

}

```