L23.2: Pseudocode for public .doKnightsTour(), private .moveKnight(), and private .allTakenNearby()

PRIVATE GLOBAL VALUES:
- **ChessBoard[ ][ ]** is a 2D int array that stores all of the moves
- **Horizontal[ ]** is a 1D int array that stores the 8 different horizontal (col) moves
- **Vertical[ ]** is a 1D int array that stores the 8 different vertical (row) moves
- **Taken[ ]** is a 1D boolean array that tracks if the 8 squares around the knight are taken or not taken
- **moveCount** is an int that tracks how many moves have been made on the ChessBoard

```
//as long as the knight can be moved, move the knight
public void doKnightsTour()
{
        while(moveKnight() == true)
        {
                //no code inside loop! On purpose!
                //no code inside loop! On purpose!
        }
        printChessBoard();  //can do this here, or separately in the tester
}


//returns true if the knight was moved successfully
//returns false if the knight could not move
private boolean moveKnight()
{
        initialize local variables you need
        boolean moveMade = false;

        //check move counter
        if I haven't filled the ChessBoard
        {
                //does knight have at least one nearby move?
                if allTakenNearby() is false
                {
                        while moveMade is false
                        {
                                select a random position in the range 1 to 8 using Math.random()

                                //check the taken array
                                if this random position selected above is NOT TAKEN
                                {
                                        change global myRow by adding the Vertical array
                                        change global myCol by adding the Horizontal array
                                        increase move counter by one
                                        place move number in 2D ChessBoard array at this location
                                        moveMade = true     //exit while loop, we made a move
                                }
                        }
                }
        }
        return moveMade;
}
```