Prince Mohammad bin Fahd University

Data Structures - GEIT 2421

Section 101

Spring 2020

Instructed by Dr. Loay Alzubaidi

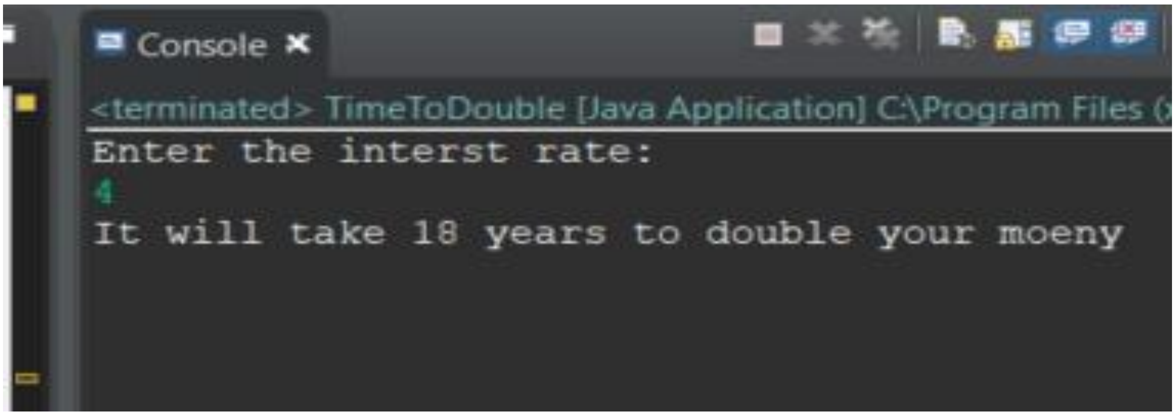
Assignment 1

by Abdulrahman Emad S Aleid

PMU#201800290

1)

```
import java.util.Scanner;

public class TimeToDouble {

    public static void main(String[] args) {

        Scanner input = new Scanner(System.in);
        int interstRate;
        int timeToDouble = 0;

        System.out.println("Enter the interst rate: ");
        interstRate = input.nextInt();

        /*Using the Rule of 72
         (72 / annual interest rate)
         to determine the time it will
        take to double the money.
         */

        timeToDouble = 72 / interstRate;
        System.out.println("It will take "
+ timeToDouble  + " years to double your moeny");

    }

}
```



<terminated> TimeToDouble [Java Application] C:\Program Files (x

```
Enter the interst rate:
4
It will take 18 years to double your moeny
```

2)

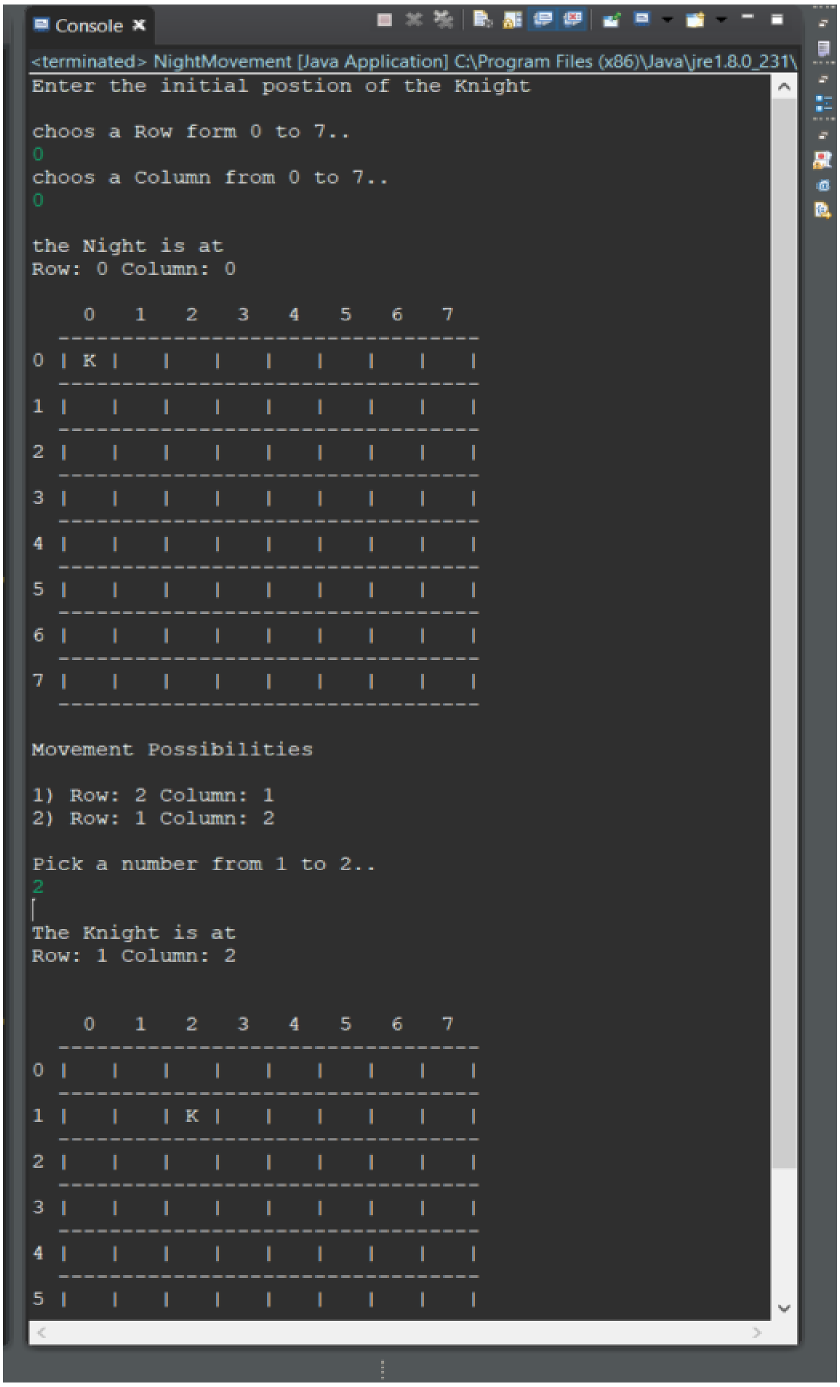
```
import java.util.Scanner;

class Chess {
    private String[][] chessBoard = new String[8][8];

    public void placeKnightOnBoard(int x, int y) {

        String knight = "K";
        String empty = " ";
        for (int i = 0; i < chessBoard.length; i++) {
            for (int j = 0; j < chessBoard.length; j++)
{
                if (i == x && j == y) {
                    chessBoard[i][j] = knight;
                } else {
                    chessBoard[i][j] = empty;
                }
            }
        }
        this.chessBoard[x][y] = knight;
    }

    public void printChess() {
        int n = 0;

        System.out.println("    0   1   2   3   4   5
6   7");
        System.out.println("  --------------------------
-------");

        for (int i = 0; i < chessBoard.length; i++) {
            System.out.print(n + " | ");

            for (int j = 0; j < chessBoard.length; j++)
{
                System.out.print(chessBoard[i][j] + " |
");
            }
            System.out.println("\n  --------------------
-------------");
            n++;
        }
        System.out.println();
    }
}
```

```
class Knight {
    private int x, y;

    Knight() {
        this.x = 0;
        this.y = 0;
    }

    public void setX(int x) {
        this.x = x;
    }

    public int getX() {
        return this.x;
    }

    public void setY(int y) {
        this.y = y;
    }

    public int getY() {
        return this.y;
    }

    public void initializePos() {

        int initialX, initialY;

        System.out.println("Enter the initial postion of
the Knight\n");

        while (true) {
            Scanner input = new Scanner(System.in);
            System.out.println("choos a Row form 0 to
7..");
            initialX = input.nextInt();
            this.x = initialX;

            System.out.println("choos a Column from 0 to
7..");
            initialY = input.nextInt();
            this.y = initialY;

            if ((initialX >= 0 && initialX < 8)
                    && (initialY >= 0 && initialY <
8)) {
```

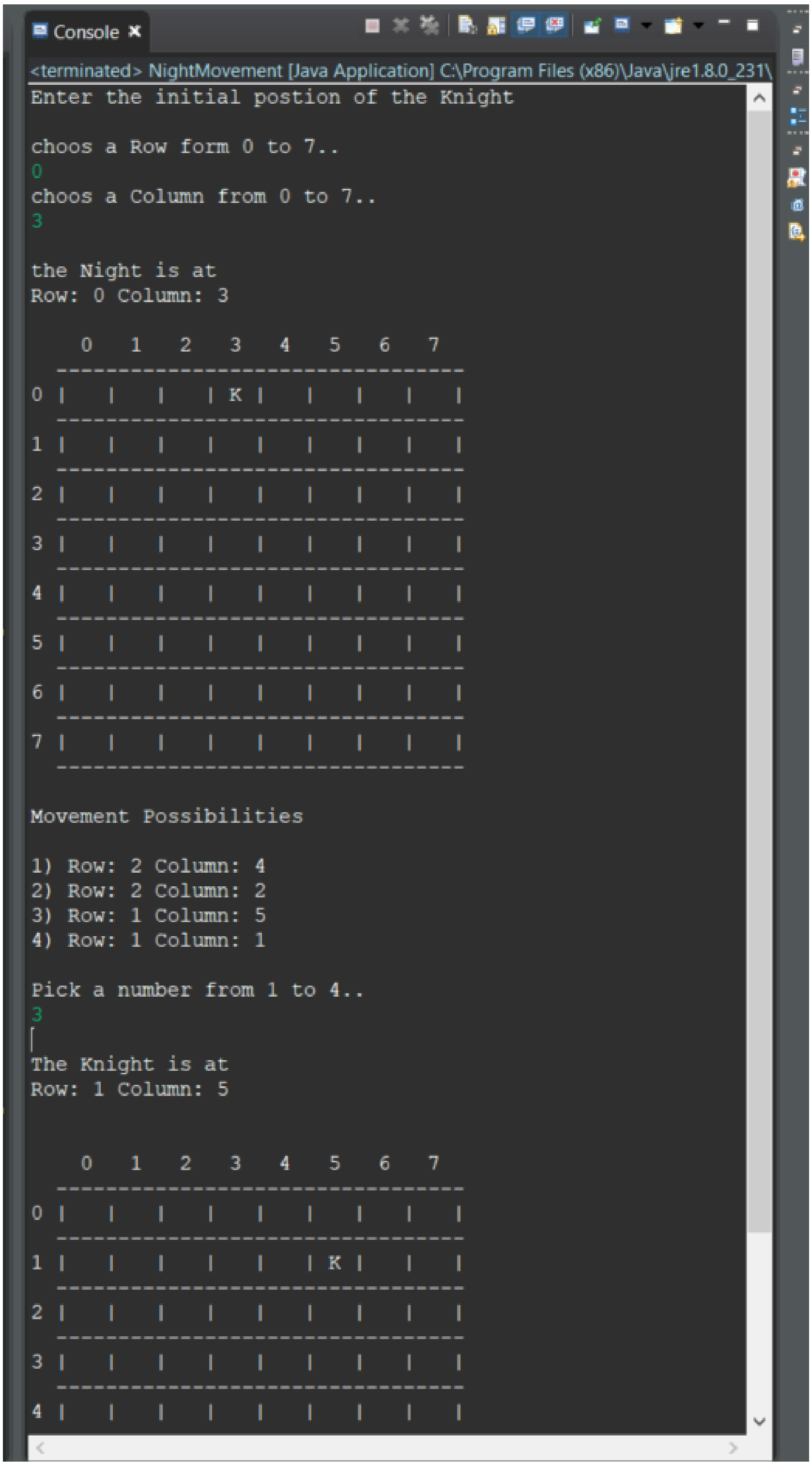
```
                break;

            } else {
                System.out.println("Please make sure
the numbers are "
                        + " between 0 and 7");
                continue;
            }
        }

        System.out.println("\nthe Night is at\n" + "Row:
" + getX()
                + " Column: " + getY() + "\n");

    }

    /*This function calculates the possible moves, takes
the user's pick from the possibilities, and moves the
knight accordingly.
*/
    public void possibilities() {

        int possX, possY, newRow, newCol;
        int[] XPossiableMoves = { 2, 2, -2, -2, 1, 1, -
1, -1 };
        int[] YPossibleMoves = { 1, -1, 1, -1, 2, -2, 2,
-2 };
        int[] emptyX = new int[8];
        int[] emptyY = new int[8];
        int numOfOptions = 0;

        System.out.print("Movement Possibilities\n\n");
        for (int i = 0; i < XPossiableMoves.length; i++)
{
            possX = this.x + XPossiableMoves[i];
            possY = this.y + YPossibleMoves[i];
            if ((possX < 0 || possX > 7) || (possY < 0
|| possY > 7)) {
                continue;
            } else {
                emptyX[numOfOptions] = possX;
                emptyY[numOfOptions] = possY;

                System.out.print((numOfOptions + 1) +
") Row: " + emptyX[numOfOptions] + " Column: "
                        + emptyY[numOfOptions] +
"\n");
```

```
                numOfOptions++;
            }
        }
        int newPosition = pickMovement(numOfOptions) -1;
        newRow = emptyX[newPosition];
        newCol = emptyY[newPosition];

        System.out.println("\nThe Knight is at\n" +
"Row: " + newRow
                + " Column: " + newCol + "\n\n");
        move(newRow, newCol);
    }

    /*This function returns the user’s pick from the
possible moves.
*/
    public int pickMovement(int numOfOptions) {

        int pick = 0;
        System.out.println("\nPick a number from 1 to "
+ numOfOptions + "..");
        while (true) {
            Scanner input = new Scanner(System.in);
            pick = input.nextInt();
            if (pick > 0 && pick <= numOfOptions)
                break;
            else {
                System.out.println("\n Please make sure
to "
                        + "enter a number from 1 to" +
numOfOptions);
                continue;
            }
        }
        return pick;
    }

    /*This function takes the new row and column potions
and place the knight in the new potion.
*/
    public void move(int newRow, int newCol) {
        this.x = newRow;
        this.y = newCol;
    }

}
```

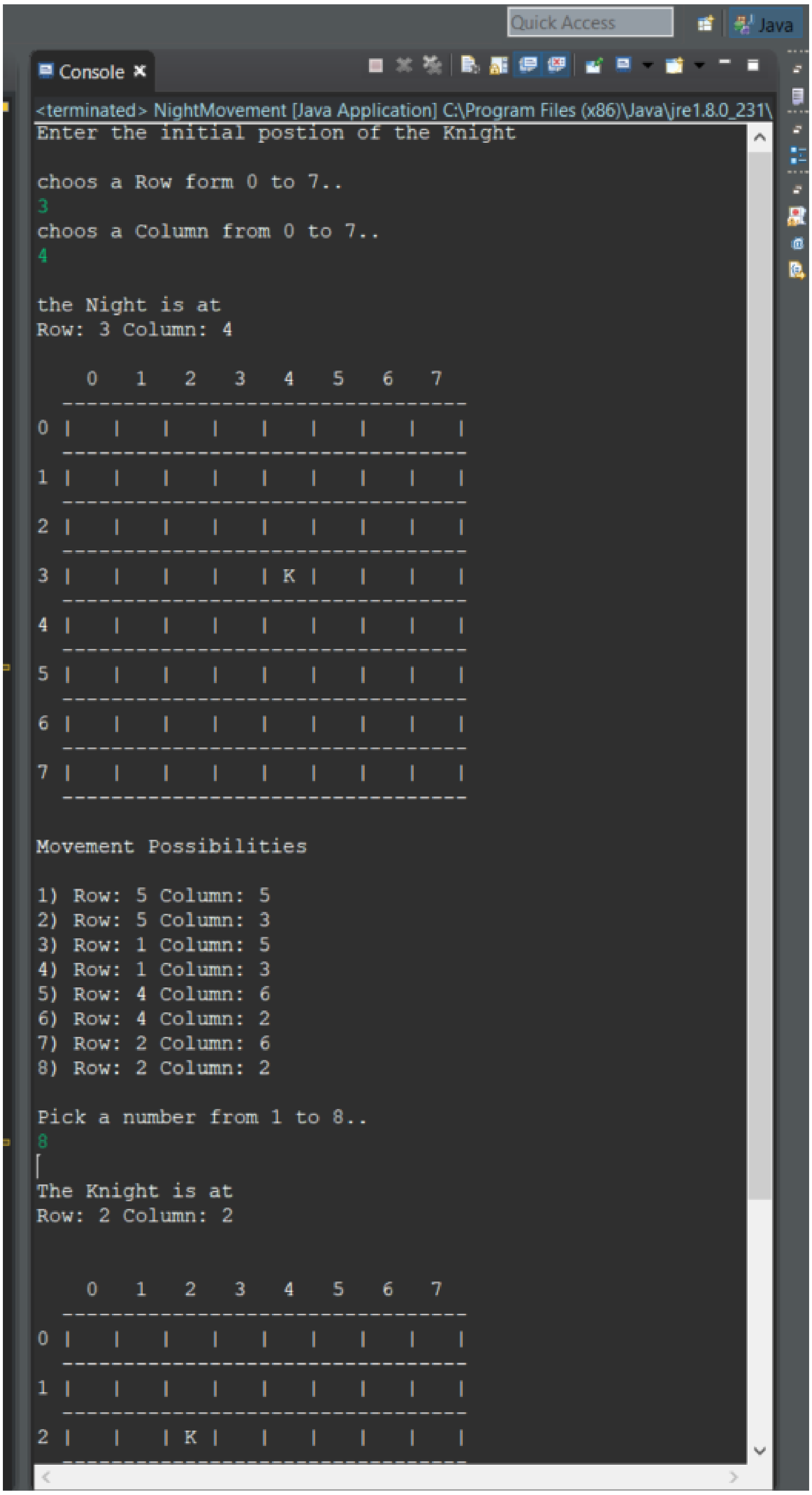
```
public class NightMovement {

    public static void main(String[] args) {

        Chess board = new Chess();
        Knight knight = new Knight();

        knight.initializePos();

        board.placeKnightOnBoard(knight.getX(),
knight.getY());
        board.printChess();

        knight.possibilities();

        board.placeKnightOnBoard(knight.getX(),
knight.getY());
        board.printChess();

    }
}
```



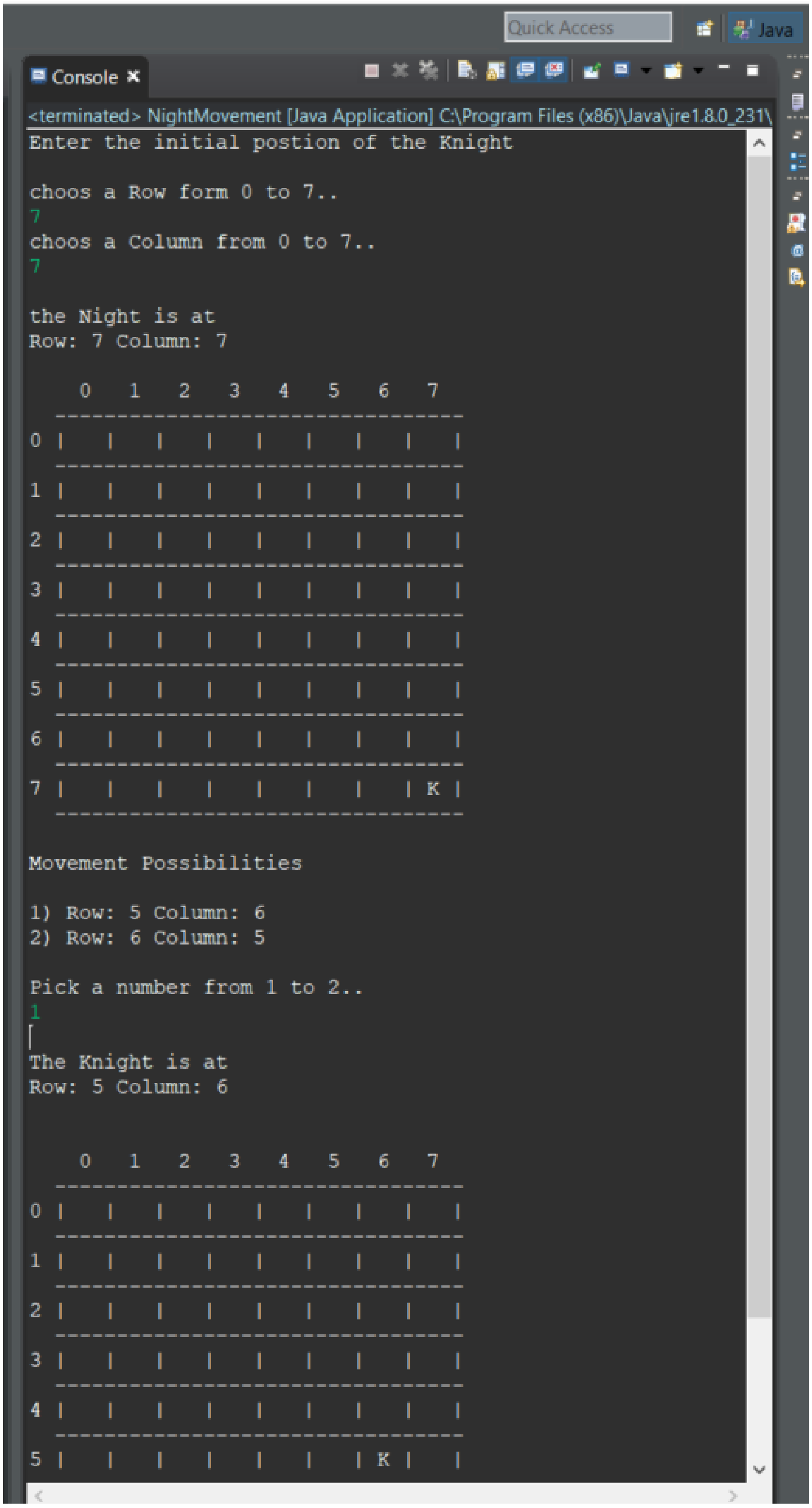
```
<terminated> NightMovement [Java Application] C:\Program Files (x86)\Java\jre1.8.0_231\
Enter the initial postion of the Knight

choos a Row form 0 to 7..
0
choos a Column from 0 to 7..
0

the Night is at
Row: 0 Column: 0

    0   1   2   3   4   5   6   7
   ---------------------------------
0 | K |   |   |   |   |   |   |   |
   ---------------------------------
1 |   |   |   |   |   |   |   |   |
   ---------------------------------
2 |   |   |   |   |   |   |   |   |
   ---------------------------------
3 |   |   |   |   |   |   |   |   |
   ---------------------------------
4 |   |   |   |   |   |   |   |   |
   ---------------------------------
5 |   |   |   |   |   |   |   |   |
   ---------------------------------
6 |   |   |   |   |   |   |   |   |
   ---------------------------------
7 |   |   |   |   |   |   |   |   |
   ---------------------------------

Movement Possibilities

1) Row: 2 Column: 1
2) Row: 1 Column: 2

Pick a number from 1 to 2..
2

The Knight is at
Row: 1 Column: 2


    0   1   2   3   4   5   6   7
   ---------------------------------
0 |   |   |   |   |   |   |   |   |
   ---------------------------------
1 |   |   | K |   |   |   |   |   |
   ---------------------------------
2 |   |   |   |   |   |   |   |   |
   ---------------------------------
3 |   |   |   |   |   |   |   |   |
   ---------------------------------
4 |   |   |   |   |   |   |   |   |
   ---------------------------------
5 |   |   |   |   |   |   |   |   |
```



```
<terminated> NightMovement [Java Application] C:\Program Files (x86)\Java\jre1.8.0_231\
Enter the initial postion of the Knight

choos a Row form 0 to 7..
0
choos a Column from 0 to 7..
3

the Night is at
Row: 0 Column: 3

    0   1   2   3   4   5   6   7
  ---------------------------------
0 |   |   |   | K |   |   |   |   |
  ---------------------------------
1 |   |   |   |   |   |   |   |   |
  ---------------------------------
2 |   |   |   |   |   |   |   |   |
  ---------------------------------
3 |   |   |   |   |   |   |   |   |
  ---------------------------------
4 |   |   |   |   |   |   |   |   |
  ---------------------------------
5 |   |   |   |   |   |   |   |   |
  ---------------------------------
6 |   |   |   |   |   |   |   |   |
  ---------------------------------
7 |   |   |   |   |   |   |   |   |
  ---------------------------------

Movement Possibilities

1) Row: 2 Column: 4
2) Row: 2 Column: 2
3) Row: 1 Column: 5
4) Row: 1 Column: 1

Pick a number from 1 to 4..
3

The Knight is at
Row: 1 Column: 5


    0   1   2   3   4   5   6   7
  ---------------------------------
0 |   |   |   |   |   |   |   |   |
  ---------------------------------
1 |   |   |   |   |   | K |   |   |
  ---------------------------------
2 |   |   |   |   |   |   |   |   |
  ---------------------------------
3 |   |   |   |   |   |   |   |   |
  ---------------------------------
4 |   |   |   |   |   |   |   |   |
```



```
<terminated> NightMovement [Java Application] C:\Program Files (x86)\Java\jre1.8.0_231\
Enter the initial postion of the Knight

choos a Row form 0 to 7..
3
choos a Column from 0 to 7..
4

the Night is at
Row: 3 Column: 4

    0   1   2   3   4   5   6   7
  ---------------------------------
0 |   |   |   |   |   |   |   |   |
  ---------------------------------
1 |   |   |   |   |   |   |   |   |
  ---------------------------------
2 |   |   |   |   |   |   |   |   |
  ---------------------------------
3 |   |   |   |   | K |   |   |   |
  ---------------------------------
4 |   |   |   |   |   |   |   |   |
  ---------------------------------
5 |   |   |   |   |   |   |   |   |
  ---------------------------------
6 |   |   |   |   |   |   |   |   |
  ---------------------------------
7 |   |   |   |   |   |   |   |   |
  ---------------------------------

Movement Possibilities

1) Row: 5 Column: 5
2) Row: 5 Column: 3
3) Row: 1 Column: 5
4) Row: 1 Column: 3
5) Row: 4 Column: 6
6) Row: 4 Column: 2
7) Row: 2 Column: 6
8) Row: 2 Column: 2

Pick a number from 1 to 8..
8

The Knight is at
Row: 2 Column: 2


    0   1   2   3   4   5   6   7
  ---------------------------------
0 |   |   |   |   |   |   |   |   |
  ---------------------------------
1 |   |   |   |   |   |   |   |   |
  ---------------------------------
2 |   |   | K |   |   |   |   |   |
```



<terminated> NightMovement [Java Application] C:\Program Files (x86)\Java\jre1.8.0_231\

```
Enter the initial postion of the Knight

choos a Row form 0 to 7..
7
choos a Column from 0 to 7..
7

the Night is at
Row: 7 Column: 7

    0   1   2   3   4   5   6   7
  ---------------------------------
0 |   |   |   |   |   |   |   |   |
  ---------------------------------
1 |   |   |   |   |   |   |   |   |
  ---------------------------------
2 |   |   |   |   |   |   |   |   |
  ---------------------------------
3 |   |   |   |   |   |   |   |   |
  ---------------------------------
4 |   |   |   |   |   |   |   |   |
  ---------------------------------
5 |   |   |   |   |   |   |   |   |
  ---------------------------------
6 |   |   |   |   |   |   |   |   |
  ---------------------------------
7 |   |   |   |   |   |   |   | K |
  ---------------------------------

Movement Possibilities

1) Row: 5 Column: 6
2) Row: 6 Column: 5

Pick a number from 1 to 2..
1

The Knight is at
Row: 5 Column: 6


    0   1   2   3   4   5   6   7
  ---------------------------------
0 |   |   |   |   |   |   |   |   |
  ---------------------------------
1 |   |   |   |   |   |   |   |   |
  ---------------------------------
2 |   |   |   |   |   |   |   |   |
  ---------------------------------
3 |   |   |   |   |   |   |   |   |
  ---------------------------------
4 |   |   |   |   |   |   |   |   |
  ---------------------------------
5 |   |   |   |   |   |   | K |   |
```