

Lösningsförslag, omkontrollskrivning PTDC

2012-01-10

```
1. public class MinesweeperBoard {
    private int size;
    private int mines;
    private MinesweeperSquare[][] board;

    public MinesweeperBoard(int size, int mines) {
        this.size = size;
        this.mines = mines;
        board = new MinesweeperSquare[size + 2][size + 2];
        reset();
    }

    private void reset() {
        Random rand = new Random();
        for (int row = 0; row <= size + 1; row++) {
            for (int col = 0; col <= size + 1; col++) {
                board[row][col] = new MinesweeperSquare();
            }
        }
        for (int m = 1; m <= mines; m++) {
            int row, col;
            do {
                row = rand.nextInt(size) + 1;
                col = rand.nextInt(size) + 1;
            } while (board[row][col].isMined());
            board[row][col].putMine();
        }
    }

    public boolean open(int row, int col) {
        board[row][col].open();
        return !board[row][col].isMined();
    }

    public void mark(int row, int col) {
        board[row][col].mark();
    }

    private int getMinesAround(int row, int col) {
        int mines = 0;
        for (int r = row - 1; r <= row + 1; r++) {
            for (int c = col - 1; c <= col + 1; c++) {
                if (board[r][c].isMined()) {
                    mines++;
                }
            }
        }
        if (board[row][col].isMined()) {
            mines--;
        }
        return mines;
    }
}
```

```

    public boolean win() {
        for (int row = 1; row <= size; row++) {
            for (int col = 1; col <= size; col++) {
                MinesweeperSquare s = board[row][col];
                if (s.isMined() && !s.isMarked() || !s.isMined() && s.isMarked()) {
                    return false;
                }
            }
        }
        return true;
    }

    public void print() {
        for (int row = 1; row <= size; row++) {
            for (int col = 1; col <= size; col++) {
                MinesweeperSquare s = board[row][col];
                if (s.isMarked()) {
                    System.out.print("M");
                } else if (!s.isOpened()) {
                    System.out.print("?");
                } else if (s.isMined()) {
                    System.out.print("X");
                } else {
                    System.out.print(getMinesAround(row, col));
                }
                System.out.print(" ");
            }
            System.out.println();
        }
    }
}

```

```

2. public class Minesweeper {
    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);
        int size = scan.nextInt();
        int mines = scan.nextInt();

        MinesweeperBoard b = new MinesweeperBoard(size, mines);

        b.print();
        System.out.print("Ange kommando (V/H row col): ");
        String clicked = scan.next();
        while (clicked.equals("V") || clicked.equals("H")) {
            int row = scan.nextInt();
            int col = scan.nextInt();
            if (clicked.equals("V")) {
                if (!b.open(row, col)) {
                    System.out.println("BOOM!");
                    b.print();
                    System.exit(0);
                }
            } else {
                b.mark(row, col);
            }
            b.print();
            System.out.print("Ange kommando (V/H row col): ");
            clicked = scan.next();
        }
    }
}

```

```
        if (b.win()) {
            System.out.println("WINNER");
        } else {
            System.out.println("LOSER");
        }
    }
}
```

3. `public static ArrayList<String> startsWith(ArrayList<String> list, char ch) {`
 `ArrayList<String> temp = new ArrayList<String>();`
 `for (int i = 0; i < list.size(); i++) {`
 `String s = list.get(i);`
 `if (s.length() > 0 && s.charAt(0) == ch) {`
 `temp.add(s);`
 `}`
 `}`
 `return temp;`
 `}`

`public static void reverse(ArrayList<String> list) {`
 `ArrayList<String> temp = new ArrayList<String>();`
 `for (int i = list.size() - 1; i >= 0; i--) {`
 `temp.add(list.get(i));`
 `}`
 `list.clear();`
 `for (int i = 0; i < temp.size(); i++) {`
 `list.add(temp.get(i));`
 `}`
 `}`

`public static ArrayList<String> lengthOrder(ArrayList<String> list) {`
 `ArrayList<String> temp = new ArrayList<String>();`
 `for (int i = 0; i < list.size(); i++) {`
 `int len = list.get(i).length();`
 `int act = temp.size() - 1;`
 `while (act >= 0 && temp.get(act).length() < len) {`
 `act--;`
 `}`
 `temp.add(act + 1, list.get(i));`
 `}`
 `return temp;`
 `}`
