

카피킬러라이트 표절 검사 결 과 확 인 서

문서표절률

19%

확 인 서 정 보

검사번호 : 00070539066

발급일자 : 2021.03.03 00:36

아이디 : alan0221@naver.com

닉네임 : alan0221

검사일자 : 2020.06.25 15:47

검사문서 : 12191574김은서_지뢰찾기.docx

비고 :

본 확인서는 alan0221@naver.com 사용자가 카피킬러에서 표절검사를 수행한 표절분석 결과에 대한 문서로 카피킬러 표절검사 시스템이 자동으로 생성한 자료입니다. 문서 작성 기준이 각 학교, 기관마다 다르기 때문에 최종 평가자의 표절평가 결과와는 다를 수 있습니다.

표절 검사 상세 결과

문서표절률	전체문장	동일문장	의심문장	인용/출처	범형/경전
19%	54	2	18	5	0

표절 결과 문서명

12191574김은서_지뢰찾기.docx

```

import java.awt.Border Layout;
import java.awt.Color;
import java.awt.GridLayout;
import java.awt.event.ActionEvent;
import java.awt.event.Action Listener;
import java.awt.event.MouseEvent;
import java.awt.event.MouseListener;
import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.DataInputStream;
import java.io.File;
import java.io.FileNotFoundException;
import java.io.FileOutputStream Stream;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;

import javax.swing.Icon;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JMenu;
import javax.swing.JMenuBar;
import javax.swing.JMenuItem;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
import javax.swing.SwingUtilities;

public class Minsweeper_12191574 extends JFrame implements Action Listener {
    int counter = 0;
    javaTimer time = new javaTimer();
    int row;
    int column;
    mineButton j[][] = new mineButton[row][column];
    mineButton minesearch[] = new mineButton[10];
    JPanel panel = new JPanel();
    int minenumberofgame;
    int[][] minerandom = new int[row][column];
    String level;
    String timesave= "0";

```

```
//게임 실행 시 필요한 이벤트
```

```
class mouseRight implements MouseListener{

    @Override
    public void mouseClicked(MouseEvent e) {
        mineButton a = (mineButton) e.getSource();
        if(SwingUtilities.isRightMouseButton(e) && e.getClickCount() == 1){
if(a.getBackground() == Color.blue){
            if(SwingUtilities.isRightMouseButton(e) && e.getClickCount() == 1){
                a.setBackground(Color.white);
                minenumberofgame++;
                numberOfMine.setText("지뢰 개수 : " + minenumberofgame + " 개");
            }
        }
        else{
            a.setBackground(Color.blue);
            minenumberofgame--;
            numberOfMine.setText("지뢰 개수 : " + minenumberofgame + " 개");
        }
    }
    if(a.getBackground()==Color.lightGray){
        a.setEnabled(false);
    }
    if(j.length == 10 && j[0].length == 10){

if (a.getBackground() == Color.blue && a.number == 9){
        counter = counter+1;

        if (counter == 13 && minenumberofgame == 0){
            JOptionPane.showMessageDialog(null, "모든 지뢰를 다 찾았습니다!", "게임 승리", JOptionPane.INFORMATION_
MESSAGE);
for(int i = 0; i < j.length; i++){
            for(int k = 0; k< j[0].length; k++){
                j[i][k].setEnabled(false);
            }
        }
    }
}
}
if(j.length == 10 && j[0].length == 20){
    if (a.getBackground() == Color.blue && a.number == 9){
        counter++;
        if (counter == 26 && minenumberofgame == 0){
JOptionPane.showMessageDialog(null, "모든 지뢰를 다 찾았습니다!", "게임 승리", JOptionPane.INFORMATION_MESSAG
E);
for(int i = 0; i < j.length; i++){
            for(int k = 0; k< j[0].length; k++){
                j[i][k].setEnabled(false);
            }
        }
    }
}
}
```

```
}
}
if(j.length == 20 && j[0].length == 30){
    if (a.getBackground() == Color.blue && a.number == 9){
        counter++;
        if (counter == 78 && minenumberofgame == 0){
            JOptionPane.showMessageDialog(null, "모든 지뢰를 다 찾았습니다!", "게임 승리", JOptionPane.INFORMATION_
MESSAGE);
for(int i = 0; i < j.length; i++){
    for(int k = 0; k < j[0].length; k++){
        j[i][k].setEnabled(false);
    }
}
}
}
}

}

@Override
public void mousePressed(MouseEvent e) { }
@Override
public void mouseReleased(MouseEvent e) { }
@Override
public void mouseEntered(MouseEvent e) { }
@Override
public void mouseExited(MouseEvent e) { }

}
class myAction implements ActionListener{

@Override
public void actionPerformed(ActionEvent e) {

    mineButton a = (mineButton) e.getSource();
    a.setText(Integer.toString(a.number));
a.setBack ground(Color.lightGray);
    if (a.number==9){
        a.setBack ground(Color.red);
        a.setText(" ");
        time.setStop();
        for(int i = 0; i < minesearch.length; i++){
            minesearch[i].setBack ground(Color.red);
        }
        JOptionPane.showMessageDialog(null, "지뢰를 밟았습니다!", "게임 패배", JOptionPane.INFORMATION_MESSAGE);
for(int i = 0; i < j.length; i++){
    for(int k = 0; k < j[0].length; k++){
        j[i][k].setEnabled(false);
    }
}
}
```

```
    }
    if(a.number == 0){
        a.setText(" ");
    }
}

}

void newGameStart_1level(){ //초급
    int minenumber = 0;
    remove(panel);
    panel = new JPanel();
panel.setLayout(new GridLayout(10, 10));
    setSize(500,500);
    j = new mineButton[10][10];
    minesearch = new mineButton[13];
    row = 10;
    column = 10;
    minenumberofgame = 13;
    minerandom = new int[10][10];
    randomMine();
    numberOfMine.setText("지뢰 개수 : " + minenumberofgame + " 개");
timer1.add(myTimer, Border Layout.CENTER);
    timer.add(numberOfMine, Border Layout.NORTH);
    time = new javaTimer();
    time.start();
    for(int i = 0; i < 10; i++){
        for(int k = 0; k <10; k++){

            j[i][k] = new mineButton();
            j[i][k].setBackground(Color.white);
panel.add(j[i][k]);
            j[i][k].number = minerandom[i][k];
            if(j[i][k].number == 9){
                minesearch[minenumber++] = j[i][k];
            }
            j[i][k].addActionListener(new myAction());
            j[i][k].addMouseListener(new mouseRight());

        }
    }
    add(panel);
setVisible(true);

}

void newGameStart_2level(){//중급
    int minenumber = 0;
    remove(panel);
    panel = new JPanel();
panel.setLayout(new GridLayout(10, 20));
    setSize(900,450);
    j = new mineButton[10][20];
    minesearch = new mineButton[26];
```

```
row = 10;
column = 20;
minenumberofgame = 26;
minerandom = new int[10][20];
randomMine();
numberOfMine.setText("지뢰 개수 : " + minenumberofgame + " 개");
timer.add(myTimer);
time = new javaTimer();
time.start();
for(int i = 0; i < 10; i++){
for(int k = 0; k < 20; k++){
    j[i][k] = new mineButton();
    j[i][k].setBackground(Color.white);
    panel.add(j[i][k]);
    j[i][k].number = minerandom[i][k];
    if(j[i][k].number == 9){
        minesearch[minenumber++] = j[i][k];
    }
    j[i][k].addActionListener(new myAction());
j[i][k].addMouseListener(new mouseRight());
    }
}
add(panel);
setVisible(true);
}
void newGameStart_3level(){ //고급
    int minenumber = 0;
    remove(panel);
    panel = new JPanel();
    setSize(1250,950);
    panel.setLayout(new GridLayout(20, 30));
j = new mineButton[20][30];
minesearch = new mineButton[78];
row = 20;
column = 30;
minenumberofgame = 78;
minerandom = new int[20][30];
randomMine();
numberOfMine.setText("지뢰 개수 : " + minenumberofgame + " 개");
timer.add(myTimer);
time = new javaTimer();
time.start();
for(int i = 0; i < 20; i++){
for(int k = 0; k < 30; k++){
    j[i][k] = new mineButton();
    j[i][k].setBackground(Color.white);
    panel.add(j[i][k]);
    j[i][k].number = minerandom[i][k];
    if(j[i][k].number == 9){
minesearch[minenumber++] = j[i][k];
    }
    j[i][k].addActionListener(new myAction());
```

```
j[i][k].addMouseListener(new mouseRight());
}
}
add(panel);
setVisible(true);
}

void randomMine(){

    for(int i = 0; i < row; i++) {
        for(int j = 0; j < column; j++){
            minerandom[i][j] = 0;
        }
    }
    for(int i = 0; i < (row*column*0.13); i++){
        int x, y;
        x = (int)(Math.random()*(row));
        y = (int)(Math.random()*(column));
        if(minerandom[x][y] == 9){
            i--;
        }
        else{
            minerandom[x][y] = 9;
        }
    }
    for(int i = 0; i < row; i++) {
    for(int j = 0; j < column; j++) {
        if(minerandom[i][j]==9){
            continue;
        }
        int count = 0;
        for(int a = i-1; a <= i+1; a++) {
            for(int b = j-1; b <= j+1; b++){
                if(i==a && j==b) continue;
                if(a==-1 || a==row || b==-1 || b==column) continue;
                if(minerandom[a][b]==9){
count++;
                }
            }
        }
        minerandom[i][j] = count;
    }
}

Minsweeper_12191574(){
    setTitle("지뢰찾기");
    makeMenu();
    Timer();
    newGameStart_1level();
    randomMine();
    setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
```

```
setVisible(true);
setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

}
JPanel timer = new JPanel();
JPanel timer1 = new JPanel();
JPanel time0 = new JPanel();
JLabel numberOfMine = new JLabel();
JLabel myTimer = new JLabel();

void Timer(){
    timer.add(numberOfMine);
timer1.add(myTimer);
    time0.add(timer, BorderLayout.WEST);
    time0.add(timer1, BorderLayout.EAST);
    add(time0, BorderLayout.NORTH);

}
class javaTimer extends Thread{
    private boolean run=false;
    public void setStop(){
        run = true;
    }
    public void run(){
        int second = (Integer.parseInt(timesave)!=0?Integer.parseInt(timesave):0);
        while(!run){
            second++;
            myTimer.setText(Integer.toString(second));
            try{
                Thread.sleep(1000);
            }catch(InterruptedException e){
e.printStackTrace();
            }
        }
    }
}
void saveMine(){
    String num = "";

    int etc = 0;
    if(j.length == 10 && j[0].length == 10){
        level = "초급";
    }
    if(j.length == 10 && j[0].length == 20){
        level = "중급";
    }
    if(j.length == 20 && j[0].length == 30){
        level = "고 급";
    }
    timesave = myTimer.getText();
}
```



```
for(int i = 0; i < j.length; i++){
    for(int k = 0; k < j[i].length; k++){
        if(j[i][k].getBackground() == Color.blue){
            etc = j[i][k].number + 50;
            num = num + Integer.toString(etc) + " ";
        }
        else if(j[i][k].getBackground() == Color.lightGray){
            etc = j[i][k].number + 30;
            num = num + Integer.toString(etc) + " ";
        }
        else {
            num = num + j[i][k].number+ " ";
        }
    }
}
try {

    BufferedWriter myfile = new BufferedWriter(new FileWriter("number.txt", false));
    myfile.write(level);
    myfile.newLine();
    myfile.write(timesave);
    myfile.newLine();
    myfile.write(num);
    myfile.close();
} catch (IOException e1) {
    // TODO Auto-generated catch block
    JOptionPane.showMessageDialog(null, "저장할 수 없습니다.");
    e1.printStackTrace();
}

}

JMenuBar mb = new JMenuBar();
JMenuItem newGame0, newGame1, newGame2, newGame3, file1, file2, help;
void makeMenu(){
    JMenu m0 = new JMenu("새게임");
    mb.add(m0);
    newGame0 = new JMenuItem("새게임");
    newGame0.addActionListener(this);
    m0.add(newGame0);
    JMenu newGame00 = new JMenu("레벨 선택");
    m0.add(newGame00);
    newGame1 = new JMenuItem("초급");
    newGame1.addActionListener(this);
    newGame00.add(newGame1);
    newGame2 = new JMenuItem("중급");
    newGame2.addActionListener(this);
    newGame00.add(newGame2);
    newGame3 = new JMenuItem("고급");
    newGame3.addActionListener(this);
```

```
newGame00.add(newGame3);
JMenu m2 = new JMenu("파일");
mb.add(m2);
file1 = new JMenuItem("저장");
m2.add(file1);
file1.addActionListener(this);
file2 = new JMenuItem("불러오기");
m2.add(file2);
JMenu m3 = new JMenu("도움말");
mb.add(m3);
help = new JMenuItem("도움말");
m3.add(help);
help.addActionListener(this);
file2.addActionListener(this);
setJMenuBar(mb);
}

public static void main(String[] args){
    new Minsweeper_12191574();
}

//메뉴 클릭했을때 발생하는 이벤트들
@Override
public void actionPerformed(ActionEvent e) {

    if(e.getSource()== newGame0){
        if((j.length == 10) && (j[0].length == 20)){
time.setStop();
            counter = 0;
            timesave = "0";
            newGameStart_2level();
        }
        else if((j.length == 20) && (j[0].length == 30)){
            time.setStop();
            counter = 0;
            timesave = "0";
            newGameStart_3level();
        }
        else if((j.length == 10) && (j[0].length == 10)){
time.setStop();
            counter = 0;
            timesave = "0";
            newGameStart_1level();
        }
    }
    if(e.getSource()== newGame1){
        time.setStop();
        counter = 0;
        timesave = "0";
        newGameStart_1level();
    }
}
```

```
if(e.getSource()== newGame2){
    time.setStop();
    counter = 0;
timesave = "0";
    newGameStart_2level();
}
if(e.getSource()== newGame3){
    time.setStop();
    counter = 0;
    timesave = "0";
    newGameStart_3level();
}
if(e.getSource()== file1){
    saveMine();
}
if(e.getSource()== file2){
    time.setStop();

BufferedReader b = null;
    try {
        b = new BufferedReader(new FileReader("number.txt"));
    } catch (FileNotFoundException e2) {
        // TODO Auto-generated catch block
        e2.printStackTrace();
    }
    String numbers;
    String level;
    int count = 0; //지뢰 개수 세는 카운터
try {
    level = b.readLine();
    if (level.equals("초급")){
        newGameStart_1level();
    }
    else if (level.equals("중급")){
        newGameStart_2level();
    }
    else if (level.equals("고 급")){
        newGameStart_3level();
    }
    time.setStop();
    time = new javaTimer();
timesave = b.readLine();
    System.out.println(timesave);
    time.start();

    while ( (numbers = b.readLine()) != null )
    {
        String[] oldnum = numbers.split("\\s");
        int che=0;
        for(int i = 0; i < j.length; i++){
            for(int k = 0; k < j[0].length; k++){
j[i][k].number = Integer.parseInt(oldnum[che++]);
```

```

        if(j[i][k].number < 30){
            //클릭안한 칸

        }
        else if(j[i][k].number < 50){
            //클릭한 숫자들
            j[i][k].number = j[i][k].number%30;
            j[i][k].setBackground(Color.lightGray);
        }
        if(j[i][k].number != 0){
            j[i][k].setText(Integer.toString(j[i][k].number));
        }
        else if(j[i][k].number >= 50){
            //파란색 칸들
            j[i][k].number = j[i][k].number%50;
            j[i][k].setBackground(Color.blue);
            count++;
        }

    }
}

minenumberofgame = minenumberofgame - count;
numberOfMine.setText("지뢰 개수 : " + minenumberofgame + " 개");

}
} catch (IOException e1) {
    // TODO Auto-generated catch block
    e1.printStackTrace();
}

}
if(e.getSource()== help){
    JOptionPane.showMessageDialog(null, "지뢰찾기 게임입니다.\n 좌측 상단은 이번 게임의 지뢰개수를 나타내며, 우클릭을
하면 한개씩 줄어듭니다."
+ "\n 좌측상단은 타이머로, 게임 시작과 동시에 시작합니다."
+ "\n 좌클릭으로 칸을 선택할수 있습니다.\n만약 지뢰를 선택하면 빨간색으로 표시되고 게임이 종료됩니다."
+ "\n우클릭으로 예상하는지뢰의 위치를 표시할 수 있으며 모든 지뢰를 표시하면 게임이 끝나게 됩니다.\n"
+ "새 게임을 누르면 현재 난이도 그대로 새로운 게임을 시작할 수 있습니다. \n"
+ "레벨 선택의 초급, 중급, 고급을 누르면 난이도를 선택할 수 있습니다.\n"
+ "저장하기를 누르면 현재의 선택 상황과 지뢰의 위치, 타이머 까지 모두 저장할 수 있습니다.\n"
+ "불러오기를 누르면 저장한게임의 레벨과 타이머까지 포함하여 이전 게임을 불러오게 됩니다.", "도움말", JOptionPane.INFORMA
TION_MESSAGE);
}
}

}

class mineButton extends JButton{
    int number;
}

```

