

인사이드 업

202101788 이은비

202101807 조예나

202101817 최지희

CONTENTS

01. 프로젝트 개요

- _팀 소개
- _기획 의도

02. 진행 과정

- _개발 환경
- _협업 방식

03. 프로젝트 상세

- _화면 구성
- _기능별 주요 코드
- _내용 보안

04. 시연

프로젝트 개요

팀 소개

- 머리 속에 있는 일들(inside)을 모바일 기기에 저장(upload) 시킨다는 의미를 가진 이름
- 투두리스트, 일기장, 월력, 연력 기능을 가지고 있는 다이어리 프로그램 개발을 목표로 함

기획 의도

- 투두리스트와 일기의 작성, 화면에 출력 가능
- 투두리스트와 일기의 내용은 암호화 시켜 파일에 저장되며, 키 값을 알지 못하면 열람 불가
- 월력에서 날짜를 누르면 그 날의 저장된 투두리스트와 일기 내용을 팝업을 통해 확인 가능
- 연력을 통해 다른 달의 월력으로 빠르게 이동 가능

진행 과정

개발 환경

- 사용 PC

- Samsung LAPTOP-QGAKR1NM
- ASUS ZenBook Pro15
- LG 15U780-PA51K

- 운영체제

- Windows 10
- Windows 11

- S/W

- Eclipse IDE

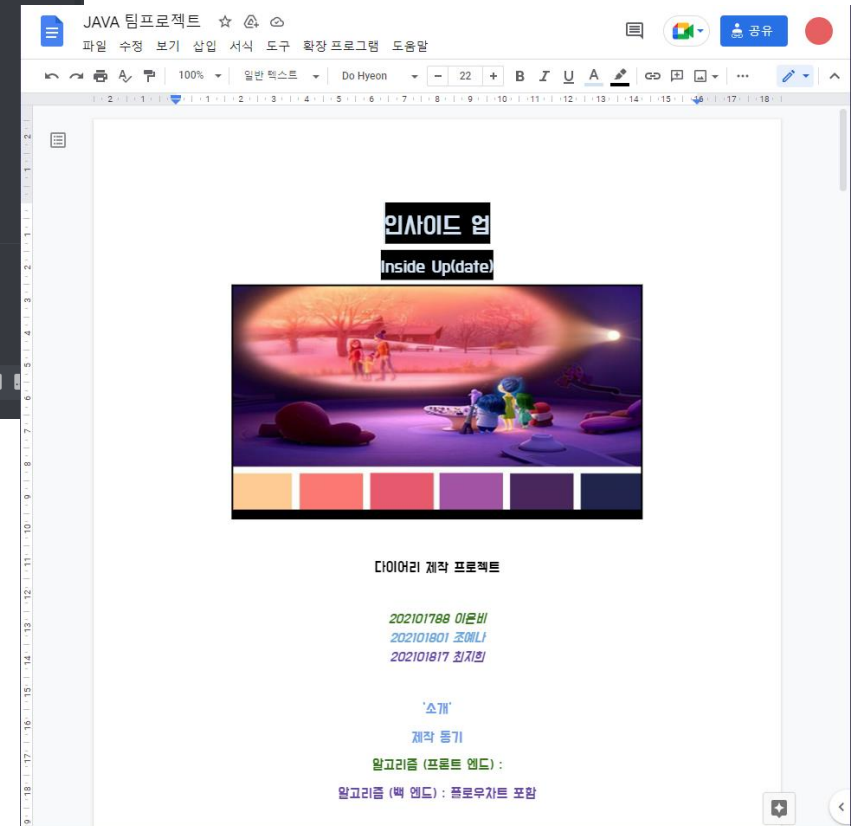
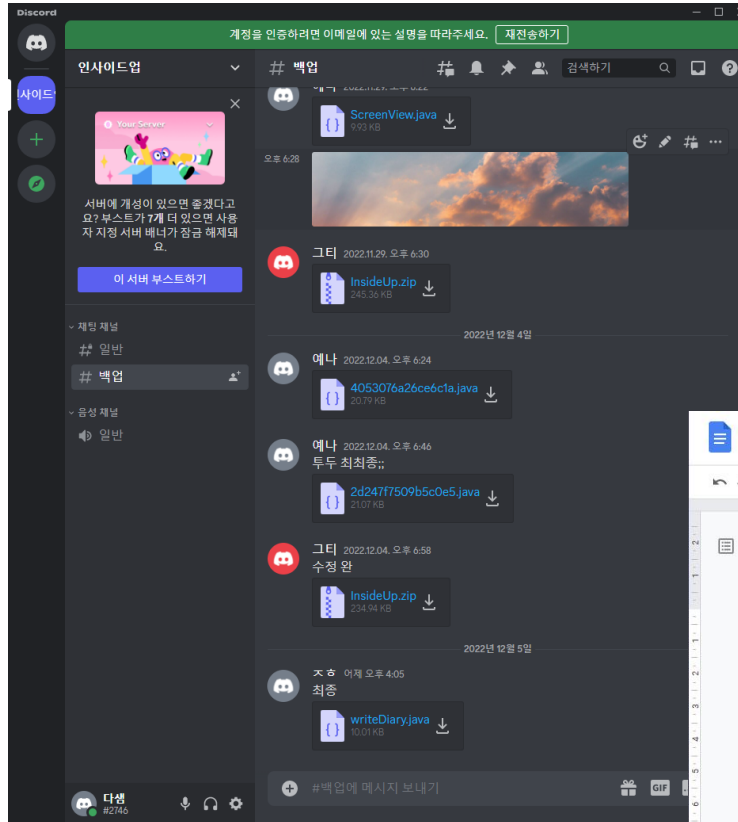
협업 방식

- 디스코드

- 코드 백업
- 원격 화상 토의

- 구글 Docx

- 참고자료 백업
- 토의 내용 기록



프로젝트 상세

기본 화면



- 내용 출력 패널
- 날짜 이동 버튼
 - 현재 날짜 레이블, 현재 연도 레이블
- 입력 텍스트 필드
- 저장 버튼
- 수정 버튼
- 삭제 버튼

```

works = new JLabel[filenames.length+1];
Font f = new Font("Gothic", Font.PLAIN, 15);
try {
    for(int i=0; i< filenames.length; i++) {
        if(file.exists()) {
            BufferedReader reader = new BufferedReader(new FileReader(st+filenames[i]));
            BufferedReader kreader = new BufferedReader(new FileReader(kst+filekeys[i]));
            String s = reader.readLine();
            String k = kreader.readLine();
            String str = decrypt(s.getBytes(), k.getBytes());
            kreader.close();
            reader.close();
            works[i] = new JLabel();
            works[i].setFont(f);
            works[i].setText(str);
        }else{
            break;
        }
    }
} catch (FileNotFoundException e) {
    e.printStackTrace();
}

```

력하여 화면에 출력.
지 않으면 스크롤 바 나타남.

```

workspace=new JPanel();
workspace.setLayout(new BorderLayout(workspace, BorderLayout.Y_AXIS)); // (Gbag);
LoadFile();
for(int i=0; i<works.length; i++) {
    Load(workspace, works[i], 400,10 + (i * 10),0,0);
}
scroll = new JScrollPane(workspace); // 스크롤패널을 선언
scroll.setBounds(10,80,405,630);

ct.add(scroll);

```

```

else if(e.getSource()==plus) {
    String str = input.getText();
    if(str.length()==0) {
        JOptionPane itsNull = new JOptionPane();
        JOptionPane.showMessageDialog(ct, "입력된 내용이 없습니다.",
"Error", JOptionPane.PLAIN_MESSAGE);
    }else {
        Save();
        input.setText("");
    }
}

```

```

File file = new File(new_file_path);
File kfile = new File(new_key_path);
String save = encrypt(sv.getBytes(), k.getBytes());

BufferedWriter kwriter = new BufferedWriter(new FileWriter(kfile));
kwriter.write(k);
kwriter.close();

BufferedWriter writer = new Bu
writer.write(save);
writer.close();

```

```

public static String appendSuffixName(String orgN) {
    String rfn = orgN;

    while(new File(rfn).exists()) {
        String fn = rfn.substring(0, rfn.lastIndexOf("_")+1);
        String nums = rfn.substring(rfn.lastIndexOf("_")+1, rfn.lastIndexOf("."));
        int ps = Integer.parseInt(nums);
        ps = ps+1;

        rfn = fn + ps + ".txt";
    }
    return rfn;
}

```

어장함.

메시지 창을 띄움.

함.

doKey폴더에 있는 해당 날

할 일 3



```

works[0] = new JLabel();
works[0].setFont(f);
works[0].setText("수정할 할 일을 선택하세요.");
try {
    for(int i=1; i< filenames.length+1; i++) {
        if(file.exists()) {
            BufferedReader reader = new BufferedReader(new F
            BufferedReader kreader = new BufferedReader(new
            String k = kreader.readLine();
            String d = reader.readLine();
            String str = decrypt(d.getBytes(), k.getBytes());
            String s = filenames[i-1];
            String sk = filekeys[i-1];
            reader.close();
            kreader.close();
            works[i] = new JLabel();
            works[i].setFont(f);
            works[i].addMouseListener(new MouseAdapter() {
                @Override
                public void mouseClicked(MouseEvent e) {
                    fn = st+s;
                    kn = kst+sk;
                    JLabel lb = (JLabel)e.getSource();
                    String p = st+s;
                    System.gc();
                    lb.setForeground(Color.BLUE);
                    input.setText(lb.getText());
                }
            });
            works[i].setText(str);
        }else{
            break;
        }
    }
} catch (FileNotFoundException e) {
    e.printStackTrace();
}

```

```

void EditSave() {
    try {
        File file= new File(fn);
        File kfile = new File(kn);
        BufferedReader reader = new BufferedReader(new FileReader(kfile));
        String k = reader.readLine();
        reader.close();

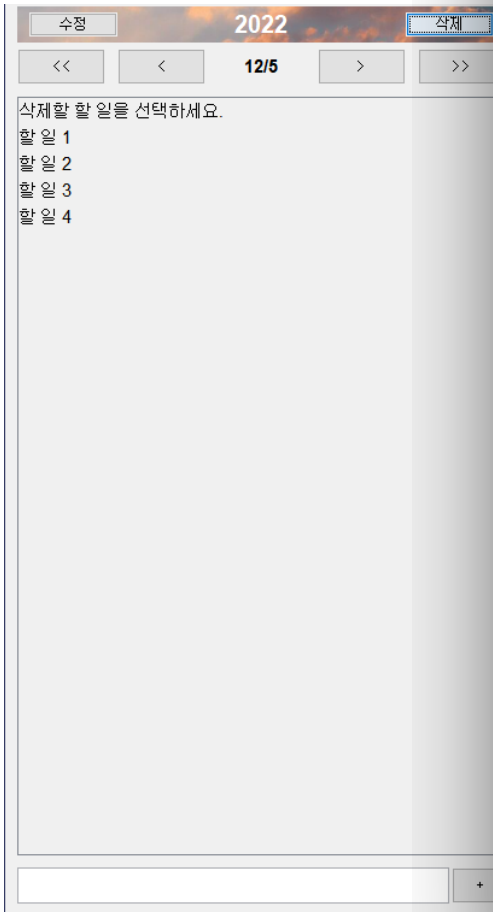
        String sv = input.getText();
        String save = encrypt(sv.getBytes(), k.getBytes());

        BufferedWriter writer = new BufferedWriter(new FileWriter(file));
        writer.write(save);
        writer.close();

        input.setText("");
        editedPlus.setVisible(false);
        plus.setVisible(true);
        SetToday();
        for(int i=0; i<works.length; i++) {
            Load(workspace, works[i], 400,10 + (i * 10),0,0);
        }
    } catch (IOException exc){
        exc.printStackTrace();
    }
}

```

삭제 방법



```
works[0] = new JLabel();
works[0].setFont(f);
works[0].setText("삭제할 할 일을 선택하세요.");
try {
    for(int i=1; i< filenames.length+1; i++) {
        if(file.exists()) {
            BufferedReader reader = new BufferedReader(new FileReader(st+filenames[i-1]));
            BufferedReader kreader = new BufferedReader(new FileReader(kp+filekeys[i-1]));
            String k = r.readLine();
            String str = decrypt(reader.readLine().getBytes(), k.getBytes());
            String fn = filenames[i-1];
            String kn = filekeys[i-1];
            reader.close();
            r.close();
            works[i] = new JLabel();
            works[i].setFont(f);
            works[i].addMouseListener(new MouseAdapter() {
                @Override
                public void mouseClicked(MouseEvent e) {
                    String dp = st+fn;
                    JLabel lb = (JLabel)e.getSource();
                    String p = kp+kn;
                    System.gc();
                    File ff = new File(dp);
                    File rk = new File(p);
                    while(ff.exists()) {
                        ff.delete();
                    }
                    while(rk.exists()) {
                        rk.delete();
                    }
                    lb.setVisible(false);
                }
            });
            works[i].setText(str);
        }else{
            break;
        }
    }
} catch (FileNotFoundException e) {
    e.printStackTrace();
}
```

파일(데이터, 키)이 삭제.

Diary

Diary

2022

<< < 12/9 > >>

Save

- 날짜 출력 레이블
- 날짜 이동 버튼
 - 하루 이동
 - 주 이동
- 내용 입력 TextArea
- 저장 버튼

Diary

해당 날짜를 포함한 일주일의 패널을 출력

JPanel

날짜(DAY)



Diary

Diary

2022

<< < 12/8 > >>

파일 저장 후, 불러오는 예시_1
여러 문장도 가능합니다.

Save

Diary

2022

<< < 12/9 > >>

오늘은 자바 발표를 하는 날이다.
무사히 발표를 끝낼 수 있었으면 좋겠다.

Save

Diary

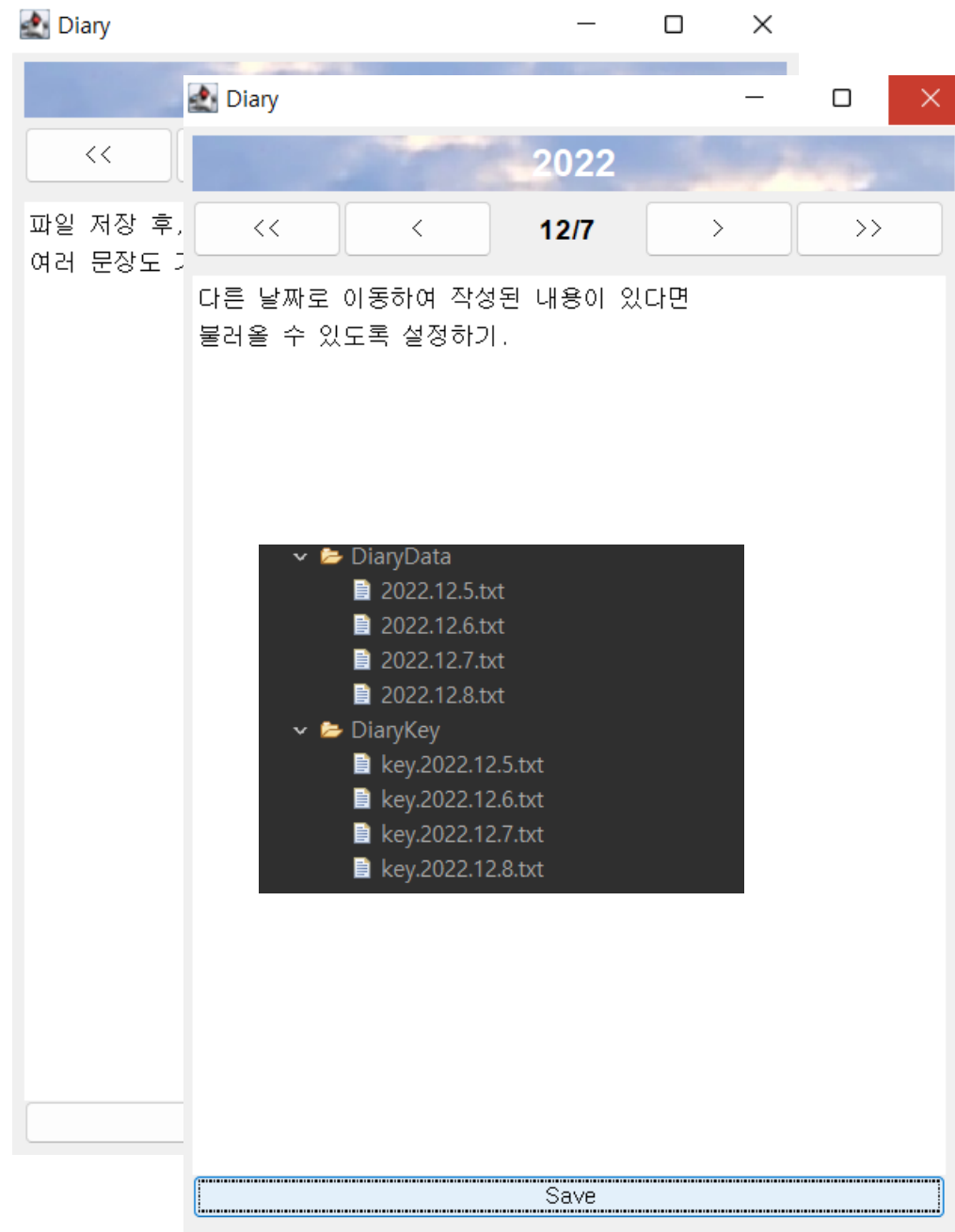
2022

<< < 12/9 > >>

Save

Diary_Save

```
public void makediary() throws Exception {  
  
    String s = input.getText();  
    String k = MakeKey();  
  
    try {  
        String filename = calyear + "." + calmonth + "." + calday + ".txt";  
        File file = new File(FILE_PATH, filename);  
        File kfile = new File(KEY_PATH, "key."+filename);  
  
        String str = encrypt(s.getBytes(), k.getBytes());  
  
        BufferedWriter writer = new BufferedWriter(new FileWriter(file));  
        writer.write(str);  
        writer.close();  
  
        BufferedWriter kwriter = new BufferedWriter(new FileWriter(kfile));  
        kwriter.write(k);  
        kwriter.close();  
  
    } catch (IOException e) {  
        e.printStackTrace();  
    }  
}
```



Diary_encrypt

암호화

-> AES 암호 알고리즘(Advanced Encryption Standard)

```
String str = encrypt(s.getBytes(), k.getBytes());
```

```
private String MakeKey() {  
    char[] a = new char[16];  
    Random ran = new Random();  
    for(int i=0; i<16; i++) {  
        a[i] = (char) ran.nextInt(10);  
    }  
    String k = new String(a);  
    return k;  
}
```

```
public static String encrypt(byte[] target, byte[] key){  
    SecretKeySpec keySpec = null;  
  
    keySpec = new SecretKeySpec(key, "AES");  
    Cipher cipher = null;  
    try {  
        cipher = Cipher.getInstance("AES");  
    } catch (NoSuchAlgorithmException e) {  
        return null;  
    } catch (NoSuchPaddingException e) {  
        return null;  
    }  
  
    try {  
        cipher.init(Cipher.ENCRYPT_MODE, keySpec);  
    } catch (InvalidKeyException e) {  
        return null;  
    }  
  
    try {  
        Encoder encoder = Base64.getEncoder();  
        return new String(encoder.encode(cipher.doFinal(target)));  
    } catch (IllegalBlockSizeException e) {  
  
    } catch (BadPaddingException e) {  
  
    }  
    return null;  
}
```

Diary_encrypt

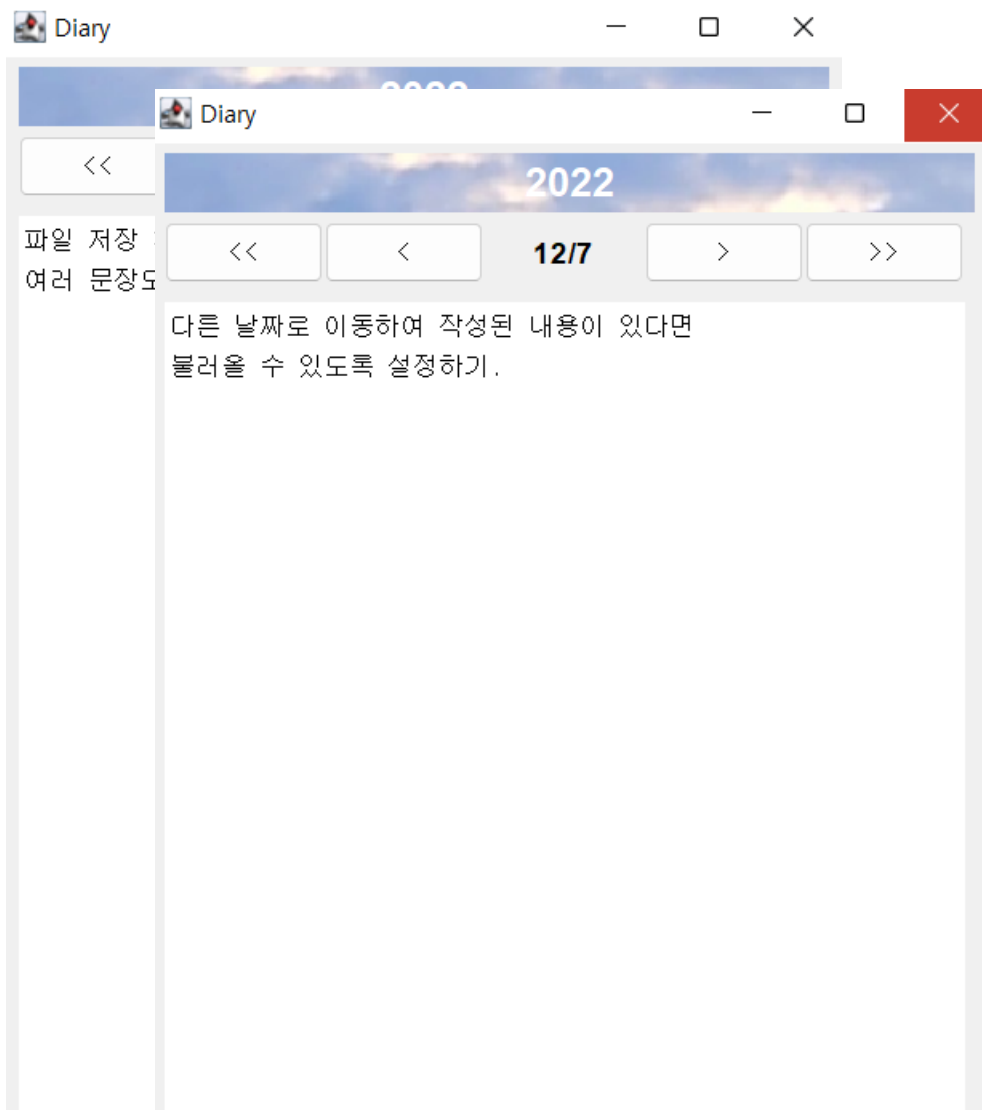
암호화 된 일기 내용

```
94jeE4FzRT67Af77EO7nc2zn5tejCLPqO+drDAUft2m5VZyDt6E7bQ+fePyNEIDXZv+  
Ag+9FyegiRPQTja/kmA==
```

암호화 시킨 'KEY' 값

```
`□|Ě □d.
```

Diary_Load



```
public void showdiary() throws Exception {
    StringBuilder stringBuilder = new StringBuilder();
    String s;

    try {
        String filename = calyear + "." + calmonth + "." + calday + ".txt";
        String kst = "key." + calyear + "." + calmonth + "." + calday + ".txt";
        File file = new File(FILE_PATH, filename);
        File kfile = new File(KEY_PATH, kst);

        if (file.exists()) {
            Scanner scan = new Scanner(kfile);
            String ks = scan.nextLine();

            Scanner sc = new Scanner(file);
            while (sc.hasNextLine()) {
                s = sc.nextLine();
                String str2 = decrypt(s.getBytes(), ks.getBytes());
                stringBuilder.append(str2);
                stringBuilder.append("\n");
            }
            input.setText(stringBuilder.toString());

        } else {
            input.setText("");
        }

    } catch (IOException e) {
        e.printStackTrace();
    }
}
```

Diary_decrypt

복호화

-> AES 암호 알고리즘(Advanced Encryption Standard)

```
public static String decrypt(byte[] target, byte[] key){
    SecretKeySpec keySpec = null;

    keySpec = new SecretKeySpec(key, "AES");
    Cipher cipher = null;
    try {
        cipher = Cipher.getInstance("AES");
    } catch (NoSuchAlgorithmException e) {

        return null;
    } catch (NoSuchPaddingException e) {
        return null;
    }

    try {
        cipher.init(Cipher.DECRYPT_MODE, keySpec);
    } catch (InvalidKeyException e) {

        return null;
    }

    try {
        Decoder encoder = Base64.getDecoder();
        return new String(cipher.doFinal(encoder.decode(target)));
    } catch (IllegalBlockSizeException e) {
        System.out.println("err");
    } catch (BadPaddingException e) {

    }

    return null;
}
```

다이어리 실행

Class DataManager

: 달력을 출력할 때 사용될 매개변수와 함수를 담은 클래스

Void setToday()

Void makeCalData(Calendar cal)

Int leapCheck(int year)

Void moveMonth(int mon)

Public class Main extends DataManager

: 구현한 달력 프로그램을 실행하고 달력 외 기능을 연결하는 클래스

Main() 생성자

Void focusToday()

Void showCal()

Class threadConrol extends Thread

달력 출력



```
private void showCal(){
    for(int i=0;i<CAL_HEIGHT;i++){
        for(int j=0;j<CAL_WIDTH;j++){
            String fontColor="black";
            if(j==0) fontColor="red";
            else if(j==6) fontColor="blue";
            dateButs[i][j].setText("<html><b><font
color="+fontColor+">" + calDates[i][j] + "</font></html>");
            JLabel todayMark = new JLabel("<html><font color=green>*
</html>");

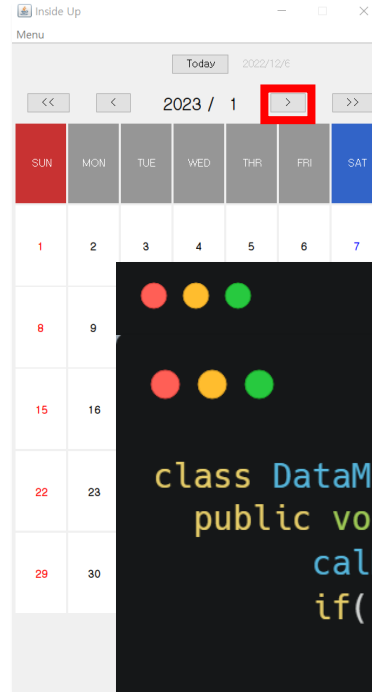
            dateButs[i][j].removeAll();
            if(calMonth == today.get(Calendar.MONTH) &&
               calYear == today.get(Calendar.YEAR) &&
               calDates[i][j] ==
today.get(Calendar.DAY_OF_MONTH)){
                dateButs[i][j].add(todayMark);
                dateButs[i][j].setToolTipText("Today");
            }

            if(calDates[i][j] == 0) dateButs[i]
[j].setVisible(false);
            else dateButs[i][j].setVisible(true);
        }
    }
}
```

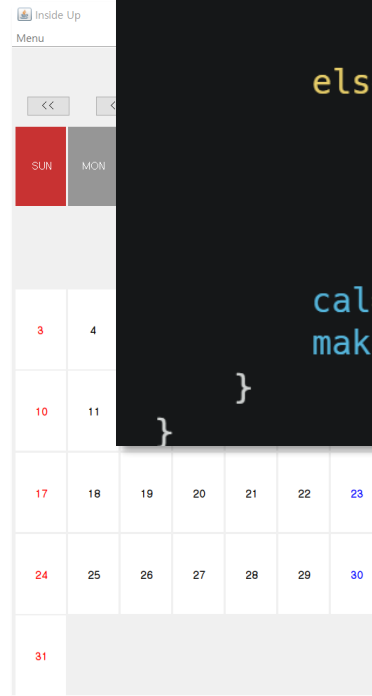

달력 이동



1개월 후

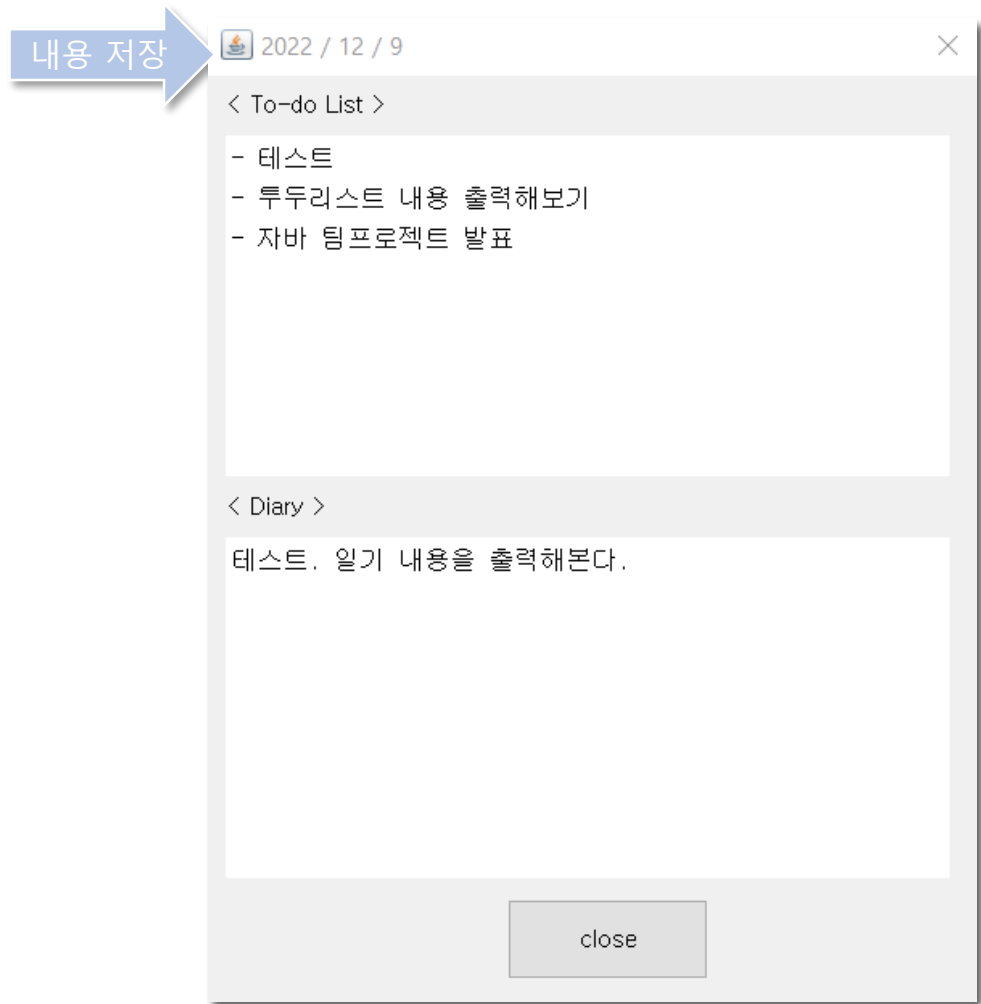


1년 후



```
class DataManager{  
    public void moveMonth(int mon) {  
        calMonth+=mon;  
        if(calMonth>11)  
            while(calMonth>11) {  
                calYear++;  
                calMonth-=12;  
            }  
        else if(calMonth<0)  
            while(calMonth<0) {  
                calYear--;  
                calMonth+=12;  
            }  
        cal=new GregorianCalendar(calYear,calMonth,calDayOfMon);  
        makeCalData(cal);  
    }  
}
```

달력 일자별 팝업

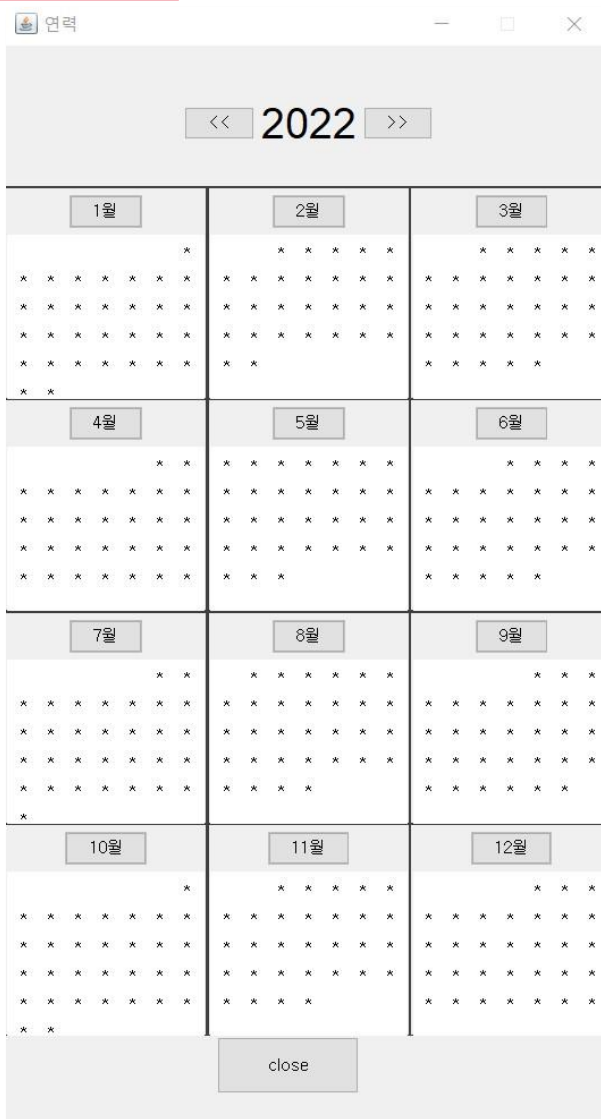


-
-
-
-
-

Pub

```
private void makePopup() {  
    todo = new JTextArea();  
    todo.setBounds(0,20,360,170);  
    todo.setEditable(false);  
    try {  
        StringBuilder sb = new StringBuilder();  
        for(int i=0;i<show_TodoData().length-1;i++) {  
            sb.append("- "+show_TodoData()[i]+"\\n");  
        }  
        String sttodo = sb.toString();  
        todo.setText(sttodo);  
    } catch (IOException e2) {  
        // TODO Auto-generated catch block  
        e2.printStackTrace();  
    }  
    goal = new JTextArea();  
    goal.setBounds(0,220,360,170);  
    goal.setEditable(false);  
    try {  
        goal.setText(show_DiaryData());  
    } catch (Exception e1) {  
        // TODO Auto-generated catch block  
        e1.printStackTrace();  
    }  
}
```

연력

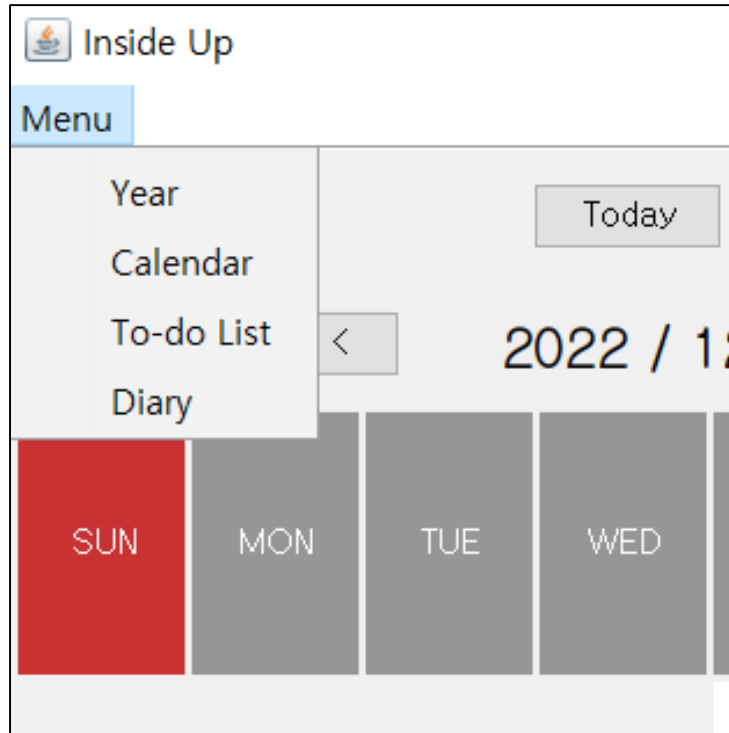


```
private String CalOnYearButs(int mon) {
    cal=new
    GregorianCalendar(Integer.parseInt(label.getText()),mon,calDayOfMon);
    int calStartingPos = (cal.get(Calendar.DAY_OF_WEEK)+7-
    (cal.get(Calendar.DAY_OF_MONTH))%7)%7;
    if(calMonth==1) //2월

    calLastDate=calLastDateOfMonth[mon]+leapCheck(Integer.parseInt(label.getText()));
    else calLastDate=calLastDateOfMonth[mon];
    String[] calArr=new String[calLastDate+1];
    StringBuilder stringBuilder = new StringBuilder();

    for (int i=0;i<calStartingPos;i++) {
        calArr[i]=" ";
        stringBuilder.append(calArr[i]+" ");
    }
    for(int j=1;j<=calLastDate;j++) {
        calArr[j]=" *";
        stringBuilder.append(calArr[j]+" ");
        if((calStartingPos+j)%7==0) {
            stringBuilder.append("\n");
        }
    }
    String ArrToString = stringBuilder.toString();
    return ArrToString;
}
```

메뉴바 생성 및 기능 연결



```
void createMenu() {
    JMenuBar menubar=new
    JMenuItem[] menuItem
    String[] itemTitle =
    JMenu menu=new JMenu
    MenuActionListener l
    for(int i=0;i<menuItem
        menuItem[i]=new
        menuItem[i].addA
        menu.add(menuItem
    }
    menubar.add(menu);
    mainFrame.setJMenuBa
}
```

```
private class MenuActionListener implements ActionListener{
    public void actionPerformed(ActionEvent e) {
        String cmd = e.getActionCommand();
        switch(cmd) {
            case "Year":
                YearMenu();
                break;
            case "Calendar":
                setToday();
                lForDateButs.actionPerformed(e);
                focusToday();
                curMMYYYYLab.setText("<html><table width=100><tr><th><font
                size=5>"+calYear+" / "+((calMonth+1)<10?"&nbsp;":"")+calMonth+1+"></th>
                </tr></table></html>");
                showCal();
                break;
            case "To-do List":
                try {
                    new Todolist();
                } catch (IOException e2) {
                    // TODO Auto-generated catch block
                    e2.printStackTrace();
                }
                break;
            case "Diary":
                try {
                    new showDiary();
                    //mainFrame.setVisible(false);
                } catch (Exception e1) {
                    // TODO Auto-generated catch block
                    e1.printStackTrace();
                }
                break;
        }
    }
}
```

시연