



# 소프트웨어프로젝트 프로젝트5 레포트

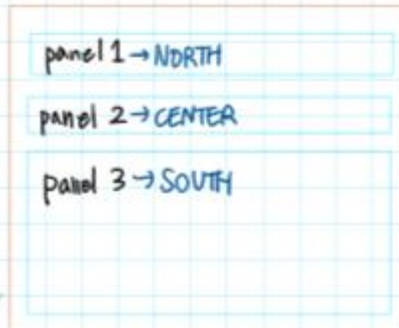
Schedule Planner GUI

소프트웨어프로젝트 02분반  
월5 수 5, 6교시 박창윤 교수님  
서동혁 조교님

소프트웨어학부  
20190323 배인경

## < Project 5 Schedule Planner GUI 설계노트 > 20190323 바인경

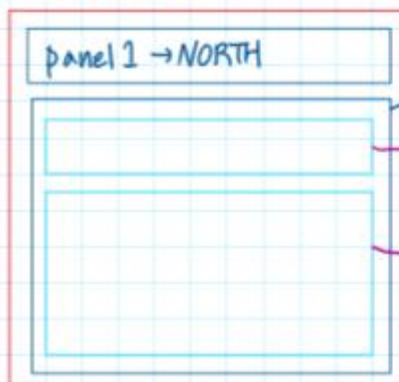
### 1) MonthSchedule class



- panel 1  
: < 버튼, 2020, 05, > 버튼
- panel 2  
: SUN ~ SAT
- panel 3  
6월, 7월 → 날짜 구현  
LocalDate 함수 사용 → 해당 년.월의  
중 일수, 첫 날의 요일 알아내기

→ 문제 발생!

panel 2부분이 창 크기에 따라 크기가 달라짐  
(CENTER)



- borderlayout → CENTER
- panel 2 → BorderLayout의 NORTH
- panel 3 → BorderLayout의 CENTER

### 2) DaySchedule class



- p1  
: Title, Start Time, End Time, Memos
- p2  
: 실제 스케줄 나타내는 부분  
GUI에서 수정이 가능하도록 TextField 클래스
- p3  
: Add, Save, Cancel 버튼

## 1. MonthSchedule class \* 파란글씨는 본인이 스스로 생각해 본 점

```
*MonthSchedule.java DaySchedule.java Test.java
1=import java.awt.*;
2 import java.awt.event.ActionEvent;
3 import java.awt.event.ActionListener;
4 import javax.swing.*;
5 import java.time.*;
6
7 public class MonthSchedule extends JFrame {
8     public MonthSchedule(String input_year, String input_month) {
9
10         JPanel BorderLayout = new JPanel(new BorderLayout());
11         JPanel panel1 = new JPanel();
12         panel1.setLayout(new GridLayout(6, 7));
13         JPanel panel2 = new JPanel(new GridLayout(1,7));
14
15         LocalDate ld = LocalDate.parse(input_year+"-"+input_month+"-01");
16         int monthLength = ld.lengthOfMonth();
17         int startPosition = ld.getDayOfWeek().getValue();
18
19         JLabel day1 = new JLabel("MON");
20         JLabel day2 = new JLabel("TUE");
21         JLabel day3 = new JLabel("WED");
22         JLabel day4 = new JLabel("THU");
23         JLabel day5 = new JLabel("FRI");
24         JLabel day6 = new JLabel("SAT");
25         JLabel day7 = new JLabel("SUN");
26         day1.setHorizontalAlignment(SwingConstants.CENTER);
27         day2.setHorizontalAlignment(SwingConstants.CENTER);
28         day3.setHorizontalAlignment(SwingConstants.CENTER);
29         day4.setHorizontalAlignment(SwingConstants.CENTER);
30         day5.setHorizontalAlignment(SwingConstants.CENTER);
31         day6.setHorizontalAlignment(SwingConstants.CENTER);
32         day7.setHorizontalAlignment(SwingConstants.CENTER);
33
34         panel2.add(day7);
35         panel2.add(day1);
36         panel2.add(day2);
37         panel2.add(day3);
38         panel2.add(day4);
39
40         panel2.add(day5);
41         panel2.add(day6);
42
43         for (int i = 0; i < startPosition; i++) {
44             JButton temp = new JButton();
45             temp.setEnabled(false);
46             panel1.add(temp);
47
48         }
49         JButton[] days = new JButton[31];
50
51         for (int i = 0; i < monthLength; i++) {
52             days[i] = new JButton(Integer.toString(i+1));
53         }
54
55         for (int i = 0; i < monthLength; i++) {
56             panel3.add(days[i]);
57         }
58
59         int last = 42 - monthLength - startPosition;
60
61         for(int i=0;i<last;i++) {
62             JButton temp = new JButton();
63             temp.setEnabled(false);
64             panel3.add(temp);
65         }
66
67         JPanel panel1 = new JPanel(new BorderLayout());
68         panel1.setLayout(new GridLayout(1, 3));
69         JPanel p2middle = new JPanel(new BorderLayout());
70         p2middle.setLayout(new GridLayout(2,1));
71
72         JButton left = new JButton("<");
73         JLabel year = new JLabel(input_year);
74         JLabel month = new JLabel(input_month);
75         year.setHorizontalAlignment(SwingConstants.CENTER);
76         month.setHorizontalAlignment(SwingConstants.CENTER);
```

line 1 ~ 5

외부 클래스 사용을 위한 import

line 7 ~ 111

월별 스케줄 GUI 구현을 위한 MonthSchedule class  
JFrame을 상속

line 8

클래스 생성 시 인자로 나타내고 싶은 년도(input\_year)  
와 달(input\_month)을 받아 해당 년, 월에 대해 구현하  
고자 함

line 10 ~ 13

panel1 -> < 2020, 05 > 나타냄 + GridLayout(1, 3)  
panel2 -> SUN ~ SAT 나타냄 + GridLayout(1, 7)  
panel3 -> 달력 나타냄 + GridLayout(6, 7)

line 15 ~ 17

LocalDate를 이용해 해당 월의 길이(28, 29, 30, 31일)  
와 해당 월의 첫 번째 요일을 구하고자 함

line 19 ~ 32

Label로 요일 나타냄 -> 수정이 안되는 부분

\* JLabel day1 ~ day7 생성과 동시에 가운데 정렬을  
시도하였더니 자꾸 오류가 떠서 생성 후 가운데 정렬을  
시도함

line 34 ~ 40

panel2 (1,7)에 순서대로 add해 줌

line 42 ~ 65

① 해당 월의 1일 전 요일까지 빈칸 생성

② 최대 일 수가 31이므로 JButton[] days[31]을 만들  
어 해당 월의 일 수 만큼 버튼형으로 날짜를 넣음

③ panel3의 GridLayout을 (6,7)로 지정 했기 때문에  
총 42칸에서 남은 칸 수 만큼 빈칸 생성

\* 3번 과정은 필요 없다고 생각했으나 정확하게 42칸을  
add하지 않는 경우 GUI가 정확하게 구현되지 않아 시도  
하게 됨 (문의게시판 힌트를 통해 생각해 냄)

line 67 ~ 70

p2middle -> input\_year와 input\_month를 Grid  
Layout(2,1) 형태로 출력하기 위해 만듦

line 72 ~ 82

Button으로 '<'와 '>' 나타내고 Label로 input\_year와  
input\_month를 나타내 줌



```

77 p2middle.add(year);
78 p2middle.add(month);
79 JButton right = new JButton(">");
80 panel1.add(left);
81 panel1.add(p2middle);
82 panel1.add(right);
83
84 add(panel1, BorderLayout.NORTH);
85 BorderLayout.add(panel3, BorderLayout.CENTER);
86 BorderLayout.add(panel2, BorderLayout.NORTH);
87 add(borderLayout, BorderLayout.CENTER);
88
89 left.addActionListener( new ActionListener() {
90     public void actionPerformed(ActionEvent e) {
91         System.out.println("Previous Month Clicked");
92     }
93 });
94
95 right.addActionListener( new ActionListener() {
96     public void actionPerformed(ActionEvent e) {
97         System.out.println("Next Month Clicked");
98     }
99 });
100
101 for(int i=0;i<monthLength;i++) {
102     days[i].addActionListener(new ActionListener(){
103         public void actionPerformed(ActionEvent e) {
104             System.out.println("See Day Schedule");
105         }
106     });
107 }
108
109 }
110
111 }

```

\* 년도 다음 줄에 월을 삽입하기 위해 고민하다가 panel 을 하나 더 생성해 add하는 방법을 생각해 냄

line 84 ~ 87

BorderLayout 방식으로 panel들을 배치 해 줌

\* CENTER부분이 창 크기에 따라 크기가 조절이 되기 때문에 panel2를 CENTER에 배치할 경우 SUN ~ SAT 글자가 불필요하게 커지거나 작아졌음 -> oanel3를 만 드시 CENTER에 배치시키고 싶은데 NORTH에 panel1 과 panel2를 한 번에 넣을 수 없어 계속 고민함 -> JPanel BorderLayout을 생성해 BorderLayout의 NORTH에 panel2를 배치시키고 CENTER에 panel3를 배치시킴. 그 후 panel1은 가장 윗 줄에 표현되어야 하 기 때문에 NORTH에 배치하고 CENTER에 BorderLayout을 배치시킴으로써 panel3부분이 크기가 조절되도록 구현함

line 89 ~ 107

버튼 클릭 시의 동작을 이해하기 위해 콘솔창에 간단한 문장을 출력 시키기 위한 ActionListener

## 2. DaySchedule class \* 파란글씨는 본인이 스스로 생각해 본 점

```

MonthSchedule.java DaySchedule.java Test.java
1 *import java.awt.*;
6
7
8 public class DaySchedule extends JFrame {
9
10     public DaySchedule() {
11         JPanel p1 = new JPanel();
12         p1.setLayout(new GridLayout(1, 4));
13         JPanel p2 = new JPanel();
14         p2.setLayout(new GridLayout(2, 4));
15         JPanel p3 = new JPanel();
16         p3.setLayout(new GridLayout(1, 3));
17
18         JLabel label_title = new JLabel("Title");
19         JLabel label_stime = new JLabel("Start Time");
20         JLabel label_etime = new JLabel("End Time");
21         JLabel label_memos = new JLabel("Memos");
22         label_title.setHorizontalAlignment(SwingConstants.CENTER);
23         label_stime.setHorizontalAlignment(SwingConstants.CENTER);
24         label_etime.setHorizontalAlignment(SwingConstants.CENTER);
25         label_memos.setHorizontalAlignment(SwingConstants.CENTER);
26
27         p1.add(label_title);
28         p1.add(label_stime);
29         p1.add(label_etime);
30         p1.add(label_memos);
31
32         add(p1, BorderLayout.NORTH);
33
34         JTextField text_title = new JTextField("Title");
35         JTextField text_stime = new JTextField("StartTime");
36         JTextField text_etime = new JTextField("EndTime");
37         JTextField text_memos = new JTextField("Memos");
38         p2.add(text_title);
39         p2.add(text_stime);
40         p2.add(text_etime);
41         p2.add(text_memos);
42

```

line 1 ~ 5

외부 클래스 사용을 위한 import

line 8 ~ 79

일별 스케줄 GUI 구현을 위한 DaySchedule class JFrame을 상속

line 10 ~ 13

p1 -> 각 항목의 이름을 나타냄 + GridLayout(1, 4)  
p2 -> 일정 내용 나타냄(입력, 삭제, 수정 가능한 부분) + GridLayout(2, 4)  
p3 -> 취소, 추가, 저장 버튼 + GridLayout(1, 3)

line 18 ~ 32

Label형으로 젤 윗줄에 Title, Start Time, End Time, Memos를 나타냄  
해당 글자를 가운데 정렬함  
p1 panel을 NORTH에 BorderLayout형식으로 배치 시켜 가장 윗 줄에 출력되게 함

line 34 ~ 46

TextField형으로 스케줄의 세부 사항을 나타냄  
수정이 가능한 형태로 만들기 위해 TextField 사용  
현재는 GridLayout(2,4)로 고정 시켰으나 추후에 행을 추가 시킬 수 있어야 할 것으로 예상

```

43     for (int i = 0; i < 4; i++) {
44         p2.add(new JTextField());
45     }
46     add(p2, BorderLayout.CENTER);
47
48     JButton button_right = new JButton("Save");
49     JButton button_mid = new JButton("Add");
50     JButton button_left = new JButton("Cancel");
51
52     button_right.setHorizontalAlignment(SwingConstants.CENTER);
53     button_mid.setHorizontalAlignment(SwingConstants.CENTER);
54     button_left.setHorizontalAlignment(SwingConstants.CENTER);
55
56     p3.add(button_left);
57     p3.add(button_mid);
58     p3.add(button_right);
59
60     button_left.addActionListener( new ActionListener() {
61         public void actionPerformed(ActionEvent e) {
62             System.out.println("Cancel Clicked");
63         }
64     });
65     button_mid.addActionListener( new ActionListener() {
66         public void actionPerformed(ActionEvent e) {
67             System.out.println("Add Clicked");
68         }
69     });
70     button_right.addActionListener( new ActionListener() {
71         public void actionPerformed(ActionEvent e) {
72             System.out.println("Save Clicked");
73         }
74     });
75
76     add(p3, BorderLayout.SOUTH);
77 }
78 }
79 }
80

```

p2 panel을 CENTER에 BorderLayout형식으로 배치 시켜 가운데에 출력되게 함

\* 프로젝트 3의 데이터 파일에서 읽어서 스케줄 내용을 표시하려다가 6/6 문의게시판 댓글을 읽고 프로젝트3과 연동시키지 않게 됨. 따라서 임의의 내용을 넣음

#### line 48 ~ 58

Button형으로 젤 아랫 줄에 Save, Add, Cancel 나타냄 해당 글자를 가운데 정렬함

p3 panel을 SOUTH에 BorderLayout형식으로 배치 시켜 가장 아랫 줄에 출력되게 함

#### line 60 ~ 74

버튼 클릭 시의 동작을 이해하기 위해 콘솔창에 간단한 문장을 출력 시키기 위한 ActionListener

### 3. Test class \* 파란글씨는 본인이 스스로 생각해 본 점

```

MonthSchedule.java *DaySchedule.java *Test.java
1 import javax.swing.JFrame;
2
3 public class Test {
4
5     public static void main(String[] args) {
6         MonthSchedule frame1 = new MonthSchedule("2020", "05");
7         frame1.setTitle("Schedule Planner");
8         frame1.setSize(600,600);
9         frame1.setLocationRelativeTo(null);
10        frame1.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
11        frame1.setVisible(true);
12
13        DaySchedule frame2 = new DaySchedule();
14        frame2.setTitle("Day Schedule");
15        frame2.setSize(600,600);
16        frame2.setLocationRelativeTo(null);
17        frame2.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
18        frame2.setVisible(true);
19
20    }
21
22 }
23

```

#### line 1

외부 클래스 사용을 위한 import

#### line 3 ~ 22

MonthSchedule, DaySchedule을 테스트하기 위한 main을 별도의 Test Class를 만들어 실행하는 조건을 따르기 위해 만든 Test class


#### line 6 ~ 11

MonthSchedule class에서 JFrame을 상속 받았기 때문에 JFrame관련 함수 사용 가능 및 frame 구현 가능 보이고 싶은 년도와 월을 인자에 입력함 GUI의 Title, Size를 지정해 줌

#### line 13 ~ 18

DaySchedule class에서 JFrame을 상속 받았기 때문에 JFrame관련 함수 사용 가능 및 frame 구현 가능 GUI의 Title, Size를 지정해 줌

## ● 구현한 GUI 모습 및 콘솔창 출력 결과

 Schedule Planner

2020  
05

< >

SUN	MON	TUE	WED	THR	FRI	SAT
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31						

Console Problems Javadoc Declaration

Test [Java Application] C:\eclipse-java-2020-03-R-win32-x86\_64\bin\javaw.exe (2020. 6. 7. 오후 5:16:53)

Previous Month Clicked

## Next Month Clicked

[See Day Schedule](#)

[See Day Schedule](#)

[See Day Schedule](#)

See Day Schedule

[See Day Schedule](#)

[See Day Schedule](#)

▲ 위에서부터 순서대로 < 버튼 눌렀을 때 콘솔 창 출력 결과, > 버튼 눌렀을 때 콘솔 창 출력 결과, 1 일 버튼 눌렀을 때 콘솔 창 출력 결과, 2 일 버튼 눌렀을 때 콘솔 창 출력 결과, 3 일 버튼 눌렀을 때 콘솔 창 출력 결과, 29 일 버튼 눌렀을 때 콘솔 창 출력 결과, 30 일 버튼 눌렀을 때 콘솔 창 출력 결과, 31 일 버튼 눌렀을 때 콘솔 창 출력 결과

Day Schedule

Title	Start Time	End Time	Memos
개강	2020-05-16 13:00	2020-05-16 13:50	
개강2	2020-05-16 15:00	2020-05-16 16:50	

Cancel

Add

Save

Console

Problems

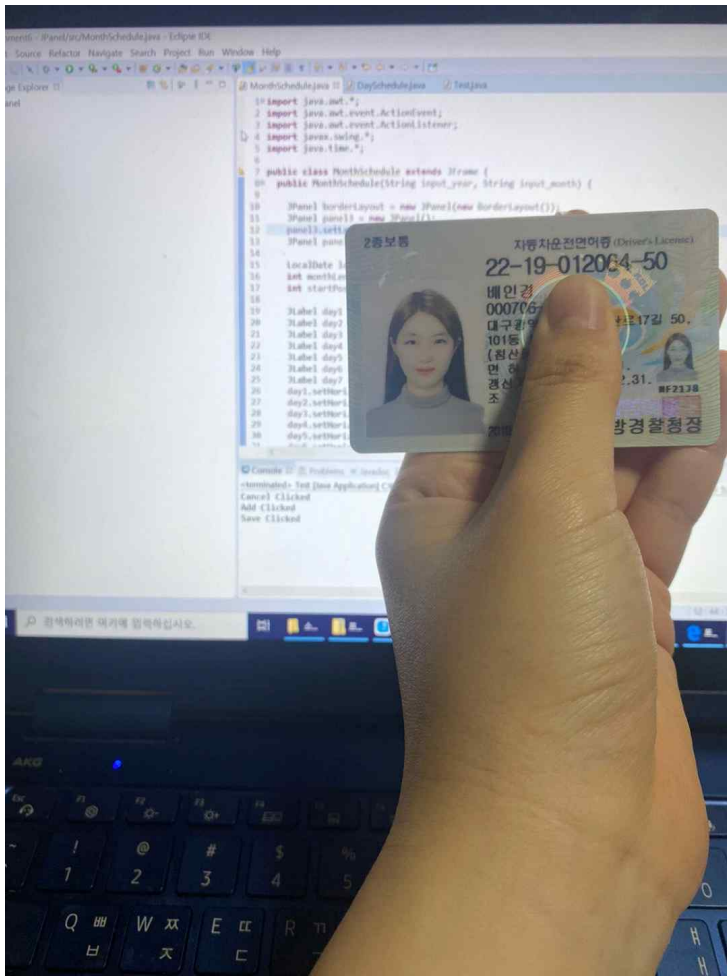
Javadoc

Declaration

Test [Java Application] C:\Weclipse-java-2020-03-R-win32-x86\_64\bin\javaw.exe (2020. 6. 7. 오후 5:20:24)  
Cancel Clicked  
Add Clicked  
Save Clicked

▲ 위에서부터 순서대로 Cancel 버튼 눌렀을 때 콘솔 창 출력 결과, Add 버튼 눌렀을 때 콘솔 창 출력 결과, Save 버튼 눌렀을 때 콘솔 창 출력 결과

▲ TextField 부분에 원하는 내용을 작성할 수 있음을 확인함



▲ 본인이 직접 구현함을 증명하기 위한 사진 첨부



## 평가표 (즉, 리포트 작성시 체크 사항)

평가 항목	학생 자체 평가 (리포트 해당 부분 표시 및 간단한 의견)	평가 (빈칸)	점수 (빈칸)
<p>완성도 (동작 여부)</p> <ul style="list-style-type: none"> <li>- GUI 화면 완성도</li> <li>- 클릭 시 동작 (콘솔 출력)</li> </ul>	<ul style="list-style-type: none"> <li>- GUI 화면이 프로젝트5 설명서 속 모습과 유사하므로 성공함 + 2020년 5월 화면을 보고서에 제출함</li> <li>- 클릭 시 동작을 통해 ActionListener의 개념/용도를 이해하였으며 콘솔창에 출력된 모습을 보고서에 제출함</li> </ul>		
<p>설계 노트</p> <ul style="list-style-type: none"> <li>- 클래스 구조</li> <li>- GUI 설계 시 착안 사항</li> <li>- 시행 착오 내용 및 해결 방안</li> </ul>	<ul style="list-style-type: none"> <li>- Class 구조 ( p. 2에 나타냄 )</li> <li>- GUI 설계 시 착안 사항을 보고서 중간 중간에 ‘* 파란글씨’ 로 나타냄</li> <li>- 시행 착오 내용 및 해결 방안을 보고서 중간 중간에 ‘* 파란글씨’ 로 나타냄</li> <li>- 직접 느낀 점 및 고안한 점을 작성함으로써 본인이 직접 작성한 프로그램임을 알리고자 함</li> <li>- 과도한 GUI 추가하지 않고 요구사항에 충실함</li> <li>- 별도의 Test Class를 만들어서 main() 실행</li> </ul>		
<p>리포트</p> <ul style="list-style-type: none"> <li>- 평가자 시각으로 리포트 검토</li> <li>- 위의 평가 요소들이 명확하게 기술되었는가?</li> <li>- 기타 건의 사항</li> </ul>	<ul style="list-style-type: none"> <li>- 동작 여부를 확연히 알아볼 수 있는 결과</li> <li>- 평가 요소들을 명확히 기술함</li> </ul>		
<p>총평/계</p>	<ul style="list-style-type: none"> <li>- JPanel 및 JFrame 이해 및 사용법 익힘</li> <li>- GUI에 대해 좀 더 친숙해 짐</li> </ul>		

\* 학생 자체 평가는 점수에 반영되지 않음.

\* 학생 스스로 자신의 보고서를 평가하면서, 체계적으로 프로젝트를 마무리하도록 유도하는 것이 목적임.