

최종 보고서

이동통신요금제 금액 계산 프로그램

201455106 김희영, 201255104 김용학

1. 조별 일 분담, 계획 수행여부에 관한 설명

조원	일분담	수행 여부
김용학	swing 컴포넌트 자료수집	인터넷 검색을 통해 공부 후 코딩 작성 (o)
	JCheckBox 자료수집	중복 선택을 위한 checkbox 공부 후 코딩 작성 (o)
	코딩 작성	여러 함수를 가지고 코딩 작성 (o)
	제안서 작성	원하는 프로그램 만든 후 그에 따른 제안서 작성 (o)
김희영	swing 컴포넌트 자료수집	자바 책을 통해 공부 후 코딩 작성 (o)
	CheckBox 계산 자료수집	중복 선택에 대한 checkbox sum값 공부 후 코딩 작성 (o)
	코딩 작성	여러 함수를 가지고 코딩 작성 (o)
	제안서 작성	원하는 프로그램 만든 후 그에 따른 제안서 작성 (o)

2. 프로젝트 코드와 실행 창.

코드-1

```
1 import java.awt.*;
2 import java.awt.event.*;
3 import javax.swing.*;
4 import javax.swing.border.EmptyBorder;
5
6 public class Telecom {
7     public static void main(String[] args) {
8         JFrame frame = new JFrame("요금제");
9         PhonePanel panel = new PhonePanel();
10
11         frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE); // close버튼에 따른 액션
12         frame.getContentPane().add(panel, BorderLayout.SOUTH);
13         // 프레임에 연결된 콘텐츠패널을 알아냄, add패널을 컨테이너에 각각상입, BorderLayout 배치관리자
14         frame.pack(); // 윈도우를 디스플레이한다
15         frame.setVisible(true); //화면에 보여지는 함수
16     }
17 }
18
19 class Orders extends JFrame { // JFrame 자바 어플리케이션을 만들 때 사용하는 것으로 Java.awt 패키지에서 정의된다
20     private JPanel contentPane; // contentPane 컨테이너 객체, 틀을 만드는 기초 공사
21     public static void showUI(JCheckBox[] cb,JCheckBox[] cb1,JCheckBox[] cb2,
22         JCheckBox[] cb3,JCheckBox[] cb4) {
23
24         EventQueue.invokeLater(new Runnable() {
25             public void run(){
26                 try{ // 예외상황 처리
27                     Orders frame = new Orders(cb,cb1,cb2,cb3,cb4);
28                     frame.setVisible(true); // setVisible(true) 화면에 보여지는 함수 호출
29                 }
30                 catch (Exception e){
31                     e.printStackTrace(); // 에러 메시지의 발생 근원지를 찾아 단계별로 에러 출력
32                 }
33             }
34         });
35     }
}
```

코드-2

```
36 //////////////////////////////////////////////////
37 public Orders(JCheckBox[] cb, JCheckBox[] cb1,
38             JCheckBox[] cb2, JCheckBox[] cb3, JCheckBox[] cb4) {
39
40     setDefaultCloseOperation(JFrame.HIDE_ON_CLOSE); // 선택 후 close에 따른 액션
41     setBounds(100, 100, 950, 300); // 선택 후 창 크기
42     contentPane = new JPanel();
43     contentPane.setBorder(new EmptyBorder(10, 10, 10, 10)); // 위, 왼쪽, 아래, 오른쪽
44     contentPane.setLayout(new BorderLayout(0, 0));
45     setContentPane(contentPane);
46
47     StringBuffer sb = new StringBuffer(); // 자바 프로그램 내에서 변하는 문자열을 다룰 때 사용한다.
48     // 한 번 생성된 후에도 계속하여 저장하고 있는 문자열의 내용을 변경할 수 있다. 그러므로,
49     // StringBuffer 클래스의 메소드는 문자열 처리 후의 결과를 원래의 StringBuffer
50     // 객체에 반영하고, 메소드 리턴 타입은 void 이다.
51     for (int i = 0; i < cb.length; i++) // apped : JCheckBox[] 데이터를 현재 문자열 끝에 추가
52         if (cb[i].isSelected()){
53             sb.append("통신사 : " + cb[i].getText() + " // ");
54         }
55     for (int i = 0; i < cb1.length; i++)
56         if (cb1[i].isSelected()){
57             sb.append("기종 : " + cb1[i].getText() + " // ");
58         }
59     for (int i = 0; i < cb2.length; i++)
60         if (cb2[i].isSelected()){
61             sb.append("할부 : " + cb2[i].getText() + " // "); // 주문내역에 고른거 표시 !!
62         }
63     for (int i = 0; i < cb3.length; i++)
64         if (cb3[i].isSelected()){
65             sb.append("요금제 : " + cb3[i].getText() + " // ");
66         }
67     sb.append("부가서비스 : ");
68     for (int i = 0; i < cb4.length; i++)
69         if (cb4[i].isSelected()){
70             sb.append(cb4[i].getText() + ", ");
71         }
72     sb.append(" // ");
```

코드-3

```

73  //////////////////////////////////
74  int selected = 1;
75  int sum=0,sum1=0,sum2=0;
76
77  if(cb[0].isSelected()){ // 통신사 선택후 기종 선택
78      if(cb1[0].isSelected())
79          sum=sum+selected*800000;
80      else if(cb1[1].isSelected())
81          sum=sum+selected*750000;
82      else if(cb1[2].isSelected())
83          sum=sum+selected*700000;
84      else if(cb1[3].isSelected())
85          sum=sum+selected*650000;
86  }
87  else if(cb[1].isSelected()){
88      if(cb1[0].isSelected())
89          sum=sum+selected*850000;
90      else if(cb1[1].isSelected())
91          sum=sum+selected*800000;
92      else if(cb1[2].isSelected())
93          sum=sum+selected*650000;
94      else if(cb1[3].isSelected())
95          sum=sum+selected*600000;
96  }
97  else if(cb[2].isSelected()){
98      if(cb1[0].isSelected())
99          sum=sum+selected*750000;
100     else if(cb1[1].isSelected())
101         sum=sum+selected*800000;
102     else if(cb1[2].isSelected())
103         sum=sum+selected*700000;
104     else if(cb1[3].isSelected())
105         sum=sum+selected*650000;
106  }
107  //////////////////////////////////

```

코드-4

```

108         if(cb2[0].isSelected()) // 할부
109             sum=sum/12;
110         else if(cb2[1].isSelected())
111             sum=sum/24;
112         else if(cb2[2].isSelected())
113             sum=sum/36;
114         else if(cb2[3].isSelected())
115             sum=sum/1;
116         ///////////////////////////////////
117         if(cb3[0].isSelected()) // 요금제
118             sum1=sum+35000;
119         else if(cb3[1].isSelected())
120             sum1=sum+45000;
121         else if(cb3[2].isSelected())
122             sum1=sum+55000;
123         else if(cb3[3].isSelected())
124             sum1=sum+65000;
125         ///////////////////////////////////
126         if(cb4[0].isSelected()){ // 부가서비스
127             sum2=sum1+selected*2000;
128             if(cb4[1].isSelected()){
129                 sum2=sum2+3000;
130                 if(cb4[2].isSelected()){
131                     sum2=sum2+5000;
132                     if(cb4[3].isSelected())
133                         sum2=sum2+6000;
134                 }
135                 else if(cb4[3].isSelected())
136                     sum2=sum2+6000;
137             }
138             else if(cb4[2].isSelected()){
139                 sum2=sum2+5000;
140                 if(cb4[3].isSelected())
141                     sum2=sum2+6000;
142             }
143             else if(cb4[3].isSelected())
144                 sum2=sum2+6000;
145         } //

```

코드-5

```

146     else if(cb4[1].isSelected()){
147         sum2=sum1+selected*3000;
148         if(cb4[2].isSelected()){
149             sum2=sum2+5000;
150             if(cb4[3].isSelected())
151                 sum2=sum2+6000;
152         }
153         else if(cb4[3].isSelected())
154             sum2=sum2+6000;
155     } //
156     else if(cb4[2].isSelected()){
157         sum2=sum1+selected*5000;
158         if(cb4[3].isSelected())
159             sum2=sum2+6000;
160     } //
161     else if(cb4[3].isSelected()){
162         sum2=sum1+selected*6000;
163     } //
164     sb.append("총 계산된 금액 : " + sum2 + "원");
165
166
167     //////////////////////////////////////
168     JLabel lblNewLabel = new JLabel(sb.toString());
169
170     lblNewLabel.setHorizontalAlignment(SwingConstants.CENTER); // 컴포넌트 정렬
171     contentPane.add(lblNewLabel, BorderLayout.CENTER);
172
173     JPanel panel = new JPanel();
174     contentPane.add(panel, BorderLayout.SOUTH);
175
176     //////////////////////////////////////
177     JButton btnNewButton = new JButton("예"); // 선택 후 예 누르면 창 꺼지는 기능
178     btnNewButton.addActionListener(new ActionListener() {
179
180         public void actionPerformed(ActionEvent e) {
181             setVisible(false);
182         }
183     });
184     panel.add(btnNewButton);
185
186     //////////////////////////////////////

```


코드-6

```

187         JButton btnNewButton_1 = new JButton("아니오"); // 선택 후 아니오 누르면 창 꺼지는 기능
188         btnNewButton_1.addActionListener(new ActionListener() {
189
190             public void actionPerformed(ActionEvent e) {
191                 setVisible(false);
192             }
193         });
194         panel.add(btnNewButton_1);
195
196         //////////////////////////////////////
197         JLabel lblNewLabel_1 = new JLabel("선택한 내용은 다음과 같습니다."); // 선택 후 복쪽 라벨 문구
198         contentPane.add(lblNewLabel_1, BorderLayout.NORTH);
199     }
200 }
201
202 //////////////////////////////////////
203 class PhonePanel extends JPanel {
204     private JButton button = new JButton("선택");
205     private JButton button1 = new JButton("닫기");
206     private JCheckBox[] checkBox, checkBox1, checkBox2, checkBox3, checkBox4;
207     private JLabel label0 = new JLabel("요금제 계산하자!!");
208     private JLabel label = new JLabel("선택한 것");
209     private JLabel label1 = new JLabel("선택한 것");
210     private JLabel label2 = new JLabel("선택한 것");
211     private JLabel label3 = new JLabel("통신사");
212     private JLabel label4 = new JLabel("기종");
213     private JLabel label5 = new JLabel("할부개월선택");
214     private JLabel label6 = new JLabel("요금제선택");
215     private JLabel label7 = new JLabel("부가서비스(중복가능)");
216     private JLabel label8 = new JLabel("선택한 것");
217     private JLabel label9 = new JLabel("선택한 것");
218     private JTextField textField = new JTextField(10);
219     private JTextField textField1 = new JTextField(10);
220     private JTextField textField2 = new JTextField(10);
221     private JTextField textField3 = new JTextField(10);
222     private JTextField textField4 = new JTextField(10);
223     private String[] PhoneText = { "SK", "KT", "LG", "기타" };
224     private String[] PhoneText1 = { "iPhone", "Galaxy", "G5", "넥서스" };
225     private String[] PhoneText2 = { "12개월", "24개월", "36개월", "일시불" };
226     private String[] PhoneText3 = { "35", "45", "55", "무제한" };
227     private String[] PhoneText4 = { "문자무제한", "음악 스트리밍", "통화무제한", "데이터 +56" };

```

코드-7

```
228 public PhonePanel() {
229     checkBox = new JCheckBox[PhoneText.length];
230     checkBox1 = new JCheckBox[PhoneText1.length];
231     checkBox2 = new JCheckBox[PhoneText2.length];
232     checkBox3 = new JCheckBox[PhoneText3.length];
233     checkBox4 = new JCheckBox[PhoneText4.length];
234
235     PhoneListener listener = new PhoneListener();
236
237     for (int i = 0; i < checkBox.length; i++) {
238         checkBox[i] = new JCheckBox(PhoneText[i]);
239         checkBox1[i] = new JCheckBox(PhoneText1[i]);
240         checkBox2[i] = new JCheckBox(PhoneText2[i]);
241         checkBox3[i] = new JCheckBox(PhoneText3[i]);
242         checkBox4[i] = new JCheckBox(PhoneText4[i]);
243         setLayout(null);
244         checkBox[i].setBounds(50, 100 + 50 * i, 80, 30);
245         checkBox1[i].setBounds(200, 100 + 50 * i, 80, 30);
246         checkBox2[i].setBounds(350, 100 + 50 * i, 80, 30);
247         checkBox3[i].setBounds(500, 100 + 50 * i, 80, 30);
248         checkBox4[i].setBounds(650, 100 + 50 * i, 110, 30);
249         add(checkBox[i]);
250         add(checkBox1[i]);
251         add(checkBox2[i]);
252         add(checkBox3[i]);
253         add(checkBox4[i]);
254         checkBox[i].addActionListener(listener);
255         checkBox1[i].addActionListener(listener);
256         checkBox2[i].addActionListener(listener);
257         checkBox3[i].addActionListener(listener);
258         checkBox4[i].addActionListener(listener);
259         button.addActionListener(listener);
260         button1.addActionListener(listener);
261     }
262 }
```


코드-8

```

263 button.setBounds(300, 380, 60, 30);
264 button.addActionListener(new ActionListener() {
265
266     public void actionPerformed(ActionEvent e) {
267         Orders.showUI(checkBox, checkBox1, checkBox2, checkBox3, checkBox4);
268     }
269 });
270 button1.setBounds(380, 380, 60, 30);
271 button1.addActionListener(new ActionListener() {
272
273     public void actionPerformed(ActionEvent e) {
274         System.exit(0);
275     }
276 });
277
278 label0.setBounds(340, 10, 200, 80);
279 label1.setBounds(50, 270, 80, 80);
280 label11.setBounds(200, 270, 80, 80);
281 label12.setBounds(350, 270, 80, 80);
282 label13.setBounds(50, 45, 80, 80);
283 label14.setBounds(200, 45, 80, 80);
284 label15.setBounds(350, 45, 80, 80);
285 label16.setBounds(500, 45, 80, 80);
286 label17.setBounds(650, 45, 130, 80);
287 label18.setBounds(500, 270, 80, 80);
288 label19.setBounds(650, 270, 80, 80);
289
290 textField.setBounds(50, 330, 100, 30);
291 textField1.setBounds(200, 330, 100, 30);
292 textField2.setBounds(350, 330, 100, 30);
293 textField3.setBounds(500, 330, 100, 30);
294 textField4.setBounds(650, 330, 100, 30);
295
296 add(button); // 항목을 추가해주는 함수
297 add(button1);
298
299 add(label0);
300 add(label1);
301 add(label11);
302 add(label12);
303 add(label13);
304 add(label14);
305 add(label15);
306 add(label16);
307 add(label17);
308 add(label18);
309 add(label19);
310
311 add(textField);
312 add(textField1);
313 add(textField2);
314 add(textField3);
315 add(textField4);
316
317 setPreferredSize(new Dimension(800, 500)); // 창 크기
318 setBackground(Color.white); // 배경색
319
320 }

```

```

322 private class PhoneListener implements ActionListener {
323     public void actionPerformed(ActionEvent event) {
324         StringBuffer Phone = new StringBuffer();
325         StringBuffer Phone1 = new StringBuffer();
326         StringBuffer Phone2 = new StringBuffer();
327         StringBuffer Phone3 = new StringBuffer();
328         StringBuffer Phone4 = new StringBuffer();
329
330         for (int i = 0; i < checkBox.length; i++)
331             if (checkBox[i].isSelected()){
332                 Phone.append(checkBox[i].getText() + " ");
333             }
334         for (int i = 0; i < checkBox1.length; i++)
335             if (checkBox1[i].isSelected()){
336                 Phone1.append(checkBox1[i].getText() + " ");
337             }
338         for (int i = 0; i < checkBox2.length; i++)
339             if (checkBox2[i].isSelected()){
340                 Phone2.append(checkBox2[i].getText() + " ");
341             }
342         for (int i = 0; i < checkBox.length; i++)
343             if (checkBox3[i].isSelected()){
344                 Phone3.append(checkBox3[i].getText() + " ");
345             }
346         for (int i = 0; i < checkBox.length; i++)
347             if (checkBox4[i].isSelected()){
348                 Phone4.append(checkBox4[i].getText() + " ");
349             }
350
351         textField.setText(Phone.toString());
352         textField1.setText(Phone1.toString());
353         textField2.setText(Phone2.toString());
354         textField3.setText(Phone3.toString());
355         textField4.setText(Phone4.toString());
356     }
357 }
358 }

```

실행창-1

요금제

요금제 계산하자!!

통신사	기종	할부개월선택	요금제선택	부가서비스(중복가능)
<input checked="" type="checkbox"/> SK	<input type="checkbox"/> iPhone	<input type="checkbox"/> 12개월	<input type="checkbox"/> 35	<input type="checkbox"/> 문자무제한
<input type="checkbox"/> KT	<input checked="" type="checkbox"/> Galaxy	<input checked="" type="checkbox"/> 24개월	<input checked="" type="checkbox"/> 45	<input checked="" type="checkbox"/> 음악 스트리밍
<input type="checkbox"/> LG	<input type="checkbox"/> G5	<input type="checkbox"/> 36개월	<input type="checkbox"/> 55	<input type="checkbox"/> 통화무제한
<input type="checkbox"/> 기타	<input type="checkbox"/> 넥서스	<input type="checkbox"/> 일시불	<input type="checkbox"/> 무제한	<input type="checkbox"/> 데이터 +5G
선택한 것	선택한 것	선택한 것	선택한 것	선택한 것
SK	Galaxy	24개월	45	음악 스트리밍

선택 닫기

실행창-2

선택한 내용은 다음과 같습니다.

통신사 : SK // 기종 : Galaxy // 할부 : 24개월 // 요금제 : 45 // 부가서비스 : 음악 스트리밍, // 총 계산된 금액 : 79250원

예 아니오

3. 프로그램 설명서

1) 프로그램 설명

세 개의 통신사 별로 다른 스마트폰의 값에 따라 원하는 통신사를 정한 후 스마트폰 기기를 선택한다. 그 후 그에 따른 개월 수와 요금제를 선택하고, 마지막으로 원하는 부가서비스까지 중복선택하면 한 달에 내야하는 금액을 계산해주는 일을 해준다.

2) 프로그램 부분별 기능 설명

요금제

1 통신사
☐ SK
☐ KT
☐ LG
☐ 기타

2 기종
☐ iPhone
☐ Galaxy
☐ G5
☐ 넥서스

3 요금제 계산하자!!
☐ 12개월
☐ 24개월
☐ 36개월
☐ 일시불

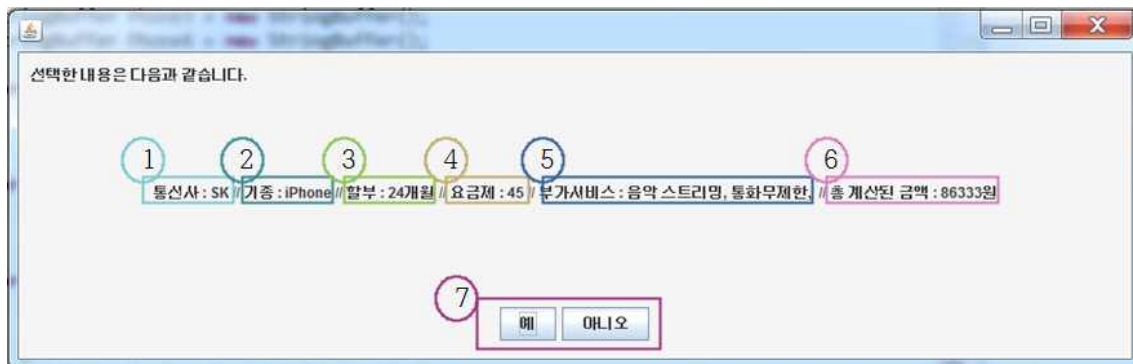
4 요금제선택
☐ 35
☐ 45
☐ 55
☐ 무제한

5 부가서비스(중복가능)
☐ 문자무제한
☐ 음악 스트리밍
☐ 통화무제한
☐ 데이터 +5G

6
선택한 것 선택한 것 선택한 것 선택한 것 선택한 것

선택 닫기

- ① : 세 개의 통신사 중에 선택할 수 있는 부분.
- ② : 각 통신사마다 다른 요금을 가진 스마트폰 기종 4가지 중 한 가지를 선택할 수 있는 부분.
- ③ : 스마트폰의 가격을 몇 개월에 나눠 돈을 낼지 선택할 수 있는 부분.
- ④ : 원하는 요금제를 선택할 수 있는 부분.
- ⑤ : 원하는 부가서비스를 다중 선택할 수 있는 부분.
- ⑥ : 선택한 통신사, 선택한 스마트폰의 종류, 선택한 할부 개월, 선택한 요금제, 선택한 부가서비스들을 나타내는 부분.



- ① : 선택한 통신사 표기.
- ② : 선택한 스마트폰 기기 표기.
- ③ : 선택한 할부 개월 표기.
- ④ : 선택한 요금제 표기.
- ⑤ : 선택한 부가서비스들 표기.
- ⑥ : 총 선택한 것들을 계산한 한 달에 내야하는 금액 표기.
- ⑦ : 맞았는지 틀렸는지 확인버튼 표기.

3) 코드부분 설명

(1)

```
public static void main(String[] args) {
    JFrame frame = new JFrame("요금제");
    PhonePanel panel = new PhonePanel();

    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    frame.getContentPane().add(panel, BorderLayout.SOUTH);
    frame.pack();
    frame.setVisible(true);
}
```

JFrame은 javax.Swing의 패키지.

JFrame에 대한 frame객체를 생성. 이때 frame의 문구로는 “요금제”를 넣음.

JPanel에 상속받은 PhonePanel에 대한 panel객체를 생성.

첫줄에서는 프레임 종료버튼이 클릭될 때 프레임을 닫고 응용 프로그램이 종료

두 번째 줄에서는 getContentPane을 써줌으로 인해 컨텐트팬을 알아냄.

add패널을 컨테이너에 각각 삽입 BorderLayout(배치관리자)로 위치를 설정.

frame.pack으로 윈도우를 디스플레이. frame.setVisible(true)를 써줌으로 화면에 출력.

(2)

```
class Orders extends JFrame {
    private JPanel contentPane;
    public static void showUI(JCheckBox[] cb,JCheckBox[] cb1,JCheckBox[] cb2,
        JCheckBox[] cb3,JCheckBox[] cb4) {

        EventQueue.invokeLater(new Runnable() {
            public void run(){
                try{
                    Orders frame = new Orders(cb,cb1,cb2,cb3,cb4);
                    frame.setVisible(true);
                }
                catch (Exception e){
                    e.printStackTrace();
                }
            }
        });
    }
}
```

JFrame에 상속받은 Orders class를 만든 후 JPanel에 대한 객체 contenPane을 선언.

contentPane은 컨테이너의 객체로 틀을 만드는 기초.

배열5개를 파라미터로 받는 showUI 함수를 만들. 그 배열 값을 인자로 받는 Orders에 대한 frame객체를 생성.

- * EventQueue.invokeLater(new Runnable()): 새로운 스레드가 사용자 인터페이스에 접근할 수 있도록 EventQueue를 사용.
- * printStackTrace(): : 에러 메시지의 발생근원지를 찾아 단계별로 에러를 출력하는 명령어.

(3)

```
public Orders(JCheckBox[] cb,JCheckBox[] cb1,
    JCheckBox[] cb2, JCheckBox[] cb3, JCheckBox[] cb4) {

    setDefaultCloseOperation(JFrame.HIDE_ON_CLOSE);
    setBounds(100, 100, 950, 300);
    contentPane = new JPanel();
    contentPane.setBorder(new EmptyBorder(10, 10, 10, 10));
    contentPane.setLayout(new BorderLayout(0, 0));
    setContentPane(contentPane);
}
```

위에서 만든 함수 Orders를 여기서 작업.

배열 5개를 파라미터로 받고, JFrame에 대해close에 따른 액션을 취함.

setBounds(좌우 위치, 상하 위치, 좌우 간격, 상하 간격)를 씌므로 인해 창의 크기와 위치를 조절.

contentPane.setBorder(new EmptyBorder(10,10,10,10)) JFrame에 상하좌우 간격을 조절.

contentPane.setLayout(new BorderLayout(0,0)) 컴포넌트패에 배치관리자 설정.

(4)

```
StringBuffer sb = new StringBuffer();

for (int i = 0; i < cb.length; i++)
    if (cb[i].isSelected()){
        sb.append("회사 : " + cb[i].getText() + " // ");
    }
for (int i = 0; i < cb1.length; i++)
    if (cb1[i].isSelected()){
        sb.append("가동 : " + cb1[i].getText() + " // ");
    }
for (int i = 0; i < cb2.length; i++)
    if (cb2[i].isSelected()){
        sb.append("주소 : " + cb2[i].getText() + " // ");
    }
for (int i = 0; i < cb3.length; i++)
    if (cb3[i].isSelected()){
        sb.append("요금제 : " + cb3[i].getText() + " // ");
    }
sb.append("서비스 : ");
for (int i = 0; i < cb4.length; i++)
    if (cb4[i].isSelected()){
        sb.append(cb4[i].getText() + ", ");
    }
sb.append(" // ");
```

StringBuffer sb = new StringBuffer(); : StringBuffer에 대한 객체 sb를 생성.

StringBuffer는 자바 프로그램 내에서 변하는 문자열을 다룰 때 사용. 한 번 생성된 후에도 계속하여 저장하고 있는 문자열의 내용을 변경할 수 있음.

Orders클래스에서 가져온 체크박스 내용을 append를 이용하여 text문자로 화면에 띄움.

(5)

```
JLabel lblNewLabel = new JLabel(sb.toString());

lblNewLabel.setHorizontalAlignment(SwingConstants.CENTER);
contentPane.add(lblNewLabel, BorderLayout.CENTER);
```

Swing에 있는 JLabel에 대한 객체를 생성 후 sb에 저장 되어있는 것들을 String으로 받음.

setHorizontalAlignment(SwingConstants.CENTER); : 중앙부분에 수평으로 정렬한다는 뜻.

contentPane에 이 라벨을 add 해줌으로써 라벨에 문구가 들어감.

(6)

```
JPanel panel = new JPanel();
contentPane.add(panel, BorderLayout.SOUTH);

JButton btnNewButton = new JButton("예");
btnNewButton.addActionListener(new ActionListener() {

    public void actionPerformed(ActionEvent e) {
        setVisible(false);
    }
});
panel.add(btnNewButton);

JButton btnNewButton_1 = new JButton("아니오");
btnNewButton_1.addActionListener(new ActionListener() {

    public void actionPerformed(ActionEvent e) {
        setVisible(false);
    }
});
panel.add(btnNewButton_1);

JLabel lblNewLabel_1 = new JLabel("선택한 내용을 다음과 같습니다.");
contentPane.add(lblNewLabel_1, BorderLayout.NORTH);
```

JPanel에 대한 panel객체를 생성.

Swing 패키지에 포함된 JButton에 대한 객체를 생성 후 title값을 입력.

actionListener는 인터페이스. '예'와 '아니오' 버튼 생성 후 actionPerformed 메소드에 그 버튼을 누르면 창이 꺼지는 setVisible(false);를 넣음.

contentPane.add(panel, BorderLayout.SOUTH); : 아래쪽에 배치.

* 액션리스너는 인터페이스라 반드시 actionPerformed 메소드를 구현 해줘야함.

이벤트를 쓰고 싶은 곳에 이벤트를 감지하는 actionListener를 add시켜주면 이벤트가 발생했을 시, actionPerformed의 파라미터로 ActionEvent가 넘어옴.

(프로그래머가 actionPerformed를 호출하는 것이 아니라 가상머신이 이벤트가 발생하면 자동으로 호출하게 하는 것.)

(7)

```
class PhonePanel extends JPanel {
    private JButton button = new JButton("선택"); // 버튼 객체 생성
    private JButton button1 = new JButton("입기"); // 버튼 객체 생성
    private JCheckBox[] checkBox, checkBox1, checkBox2, checkBox3, checkBox4; // 체크박스 배열 객체 선언
    private JLabel label0 = new JLabel("요금제 계산하기!!"); // JLabel에 대한 객체 생성
    private JLabel label = new JLabel("선택한 것"); // JLabel에 대한 객체 생성
    private JLabel label1 = new JLabel("선택한 것"); // JLabel에 대한 객체 생성
    private JLabel label2 = new JLabel("선택한 것"); // JLabel에 대한 객체 생성
    private JLabel label3 = new JLabel("통신사"); // JLabel에 대한 객체 생성
    private JLabel label4 = new JLabel("기종"); // JLabel에 대한 객체 생성
    private JLabel label5 = new JLabel("할인금액선택"); // JLabel에 대한 객체 생성
    private JLabel label6 = new JLabel("요금제선택"); // JLabel에 대한 객체 생성
    private JLabel label7 = new JLabel("부가서비스(선택가능)"); // JLabel에 대한 객체 생성
    private JLabel label8 = new JLabel("선택한 것"); // JLabel에 대한 객체 생성
    private JLabel label9 = new JLabel("선택한 것"); // JLabel에 대한 객체 생성
    private JTextField textField = new JTextField(10); // JTextField에 대한 객체 생성, (10)은 글자의 문자가 들어갈 공간의 길이로 보통 10글자 정도
    private JTextField textField1 = new JTextField(10); // JTextField에 대한 객체 생성
    private JTextField textField2 = new JTextField(10); // JTextField에 대한 객체 생성
    private JTextField textField3 = new JTextField(10); // JTextField에 대한 객체 생성
    private JTextField textField4 = new JTextField(10); // JTextField에 대한 객체 생성
    private String[] PhoneText = { "SK", "KT", "LG", "기타" }; // String 배열
    private String[] PhoneText1 = { "iPhone", "Galaxy", "G5", "서비스" }; // String 배열
    private String[] PhoneText2 = { "12개월", "24개월", "36개월", "일시불" }; // String 배열
    private String[] PhoneText3 = { "35", "45", "55", "무제한" }; // String 배열
    private String[] PhoneText4 = { "문자무제한", "음악 스트리밍", "영상무제한", "데이터 +5G" }; // String 배열
}
```

JPanel에 상속받는 PhonePanel 클래스 생성, 여러 가지 선언, 객체 생성 문.

(8)

```
public PhonePanel() {
    checkBox = new JCheckBox[PhoneText.length];
    checkBox1 = new JCheckBox[PhoneText1.length];
    checkBox2 = new JCheckBox[PhoneText2.length];
    checkBox3 = new JCheckBox[PhoneText3.length];
    checkBox4 = new JCheckBox[PhoneText4.length];

    PhoneListener listener = new PhoneListener();

    for (int i = 0; i < checkBox.length; i++) {
        checkBox[i] = new JCheckBox(PhoneText[i]);
        checkBox1[i] = new JCheckBox(PhoneText1[i]);
        checkBox2[i] = new JCheckBox(PhoneText2[i]);
        checkBox3[i] = new JCheckBox(PhoneText3[i]);
        checkBox4[i] = new JCheckBox(PhoneText4[i]);
        setLayout(null);
        checkBox[i].setBounds(50, 100 + 50 * i, 80, 30);
        checkBox1[i].setBounds(200, 100 + 50 * i, 80, 30);
        checkBox2[i].setBounds(350, 100 + 50 * i, 80, 30);
        checkBox3[i].setBounds(500, 100 + 50 * i, 80, 30);
        checkBox4[i].setBounds(650, 100 + 50 * i, 110, 30);
        add(checkBox[i]);
        add(checkBox1[i]);
        add(checkBox2[i]);
        add(checkBox3[i]);
        add(checkBox4[i]);
        checkBox[i].addActionListener(listener);
        checkBox1[i].addActionListener(listener);
        checkBox2[i].addActionListener(listener);
        checkBox3[i].addActionListener(listener);
        checkBox4[i].addActionListener(listener);
        button.addActionListener(listener);
        button1.addActionListener(listener);
    }
}
```

PhoneListener클래스에 대한 객체 listener 객체를 생성, Swing 패키지에 있는 JCheckBox 객체를 for문을 통해 다중으로 생성.

setLayout(null); : for문이 한번 돌 때 마다 위치 값을 초기화.

setBounds : 크기위치를 설정, add문을 써서 항목을 추가.

addActionListener(listener) : action리스너 등록.

(ActionListener : 텍스트 필드에 글을 입력한 후 엔터를 누르면 입력 처리.

addActionListener : ActionListener를 원하는 텍스트필드에 붙여주는 것.)

(9)

```
button.setBounds(300, 380, 60, 30);
button.addActionListener(new ActionListener() {

    public void actionPerformed(ActionEvent e) {
        Orders.showUI(checkBox,checkBox1,checkBox2,checkBox3,checkBox4);
    }
});
button1.setBounds(380, 380, 60, 30);
button1.addActionListener(new ActionListener() {

    public void actionPerformed(ActionEvent e) {
        System.exit(0);
    }
});
```

PhoneListener클래스가 ActionListener에 implements 되었기 때문에 addActionListener를 사용.

button,button1 을 add.

button - Orders클래스에 체크박스를 파라미터로 받는 함수를 구현.

button1 - System.exit(0) 을 써줌으로 닫기 버튼을 누르면 창이 꺼지는 역할을 수행.

(10)

```
label0.setBounds(340, 10, 200, 80);
label1.setBounds(50, 270, 80, 80);
label11.setBounds(200, 270, 80, 80);
label12.setBounds(350, 270, 80, 80);
label13.setBounds(50, 45, 80, 80);
label14.setBounds(200, 45, 80, 80);
label15.setBounds(350, 45, 80, 80);
label16.setBounds(500, 45, 80, 80);
label17.setBounds(650, 45, 130, 80);
label18.setBounds(500, 270, 80, 80);
label19.setBounds(650, 270, 80, 80);

textField.setBounds(50, 330, 100, 30);
textField1.setBounds(200, 330, 100, 30);
textField2.setBounds(350, 330, 100, 30);
textField3.setBounds(500, 330, 100, 30);
textField4.setBounds(650, 330, 100, 30);
```

각각 크기 위치 설정.

(11)

```
add(button);
add(button1);

add(label0);
add(label1);
add(label11);
add(label2);
add(label3);
add(label4);
add(label5);
add(label6);
add(label7);
add(label8);
add(label9);

add(textField);
add(textField1);
add(textField2);
add(textField3);
add(textField4);
```

항목을 추가해주는 함수.

(12)

```
setPreferredSize(new Dimension(800, 500));
setBackground(Color.white);
```

setPreferredSize를 통해 창의 크기를 설정.

setBackground를 통해 배경 색을 설정.

(13)

```
private class PhoneListener implements ActionListener {
    public void actionPerformed(ActionEvent event) {
        StringBuffer Phone = new StringBuffer();
        StringBuffer Phone1 = new StringBuffer();
        StringBuffer Phone2 = new StringBuffer();
        StringBuffer Phone3 = new StringBuffer();
        StringBuffer Phone4 = new StringBuffer();

        for (int i = 0; i < checkBox.length; i++)
            if (checkBox[i].isSelected()){
                Phone.append(checkBox[i].getText() + " ");
            }
        for (int i = 0; i < checkBox1.length; i++)
            if (checkBox1[i].isSelected()){
                Phone1.append(checkBox1[i].getText() + " ");
            }
        for (int i = 0; i < checkBox2.length; i++)
            if (checkBox2[i].isSelected()){
                Phone2.append(checkBox2[i].getText() + " ");
            }
        for (int i = 0; i < checkBox.length; i++)
            if (checkBox3[i].isSelected()){
                Phone3.append(checkBox3[i].getText() + " ");
            }
        for (int i = 0; i < checkBox.length; i++)
            if (checkBox4[i].isSelected()){
                Phone4.append(checkBox4[i].getText() + " ");
            }

        textField.setText(Phone.toString());
        textField1.setText(Phone1.toString());
        textField2.setText(Phone2.toString());
        textField3.setText(Phone3.toString());
        textField4.setText(Phone4.toString());
    }
}
```

ActionListener에 implements받는 PhoneListener 클래스를 생성.

ActionListener에 상속 받아 ActionEvent를 파라미터로 받는 actionPerformed함수를 구현.

StringBuffer에 대한 5개의 객체를 생성.

for문에서 체크박스 배열에 저장된 값을 선택하게 되면 Phone.append를 통해 값을 넣음.

선택된 값들을 (Phone.toString())을 통해 스트링으로 변환 후 setText통해 저장 되어있는 문자열을 넣음.

4. 프로젝트 후기 (특이점 등)

당췌던 마음과 달리 막상 코드를 짜려하니 막막했다.

체크 박스를 만드는 과정의 헛갈림이 많았다. 부가서비스 선택 같은 경우에는 다중 선택이 가능해야 했기에 더 복잡했다. 그리고 선택한 것에 대해 계산을 하나하나 해주는 계산 프로그래밍을 더해야 했기 때문에 더 어려웠다. 처음 사용해보는 스윙 함수인 만큼 초반에 틀을 만드는 것이 어려웠지만 하나를 만들고 나면 그 다음 것도 쉽게 할 수 있어서 다행이었다. 단지 콘솔 창에 뜨는 것이 아닌 실제 어플리케이션과 같이 창이 새로 뜬다는 것이 굉장히 새로웠고 재미있었다. 아쉬운 점은 두 번째 창에서 한 줄씩 찍어서 글을 나타내려했으나 \n이 듣지 않아 하지 못했다. 또한 원래 android sdk를 사용하여 정말 어플리케이션처럼 만들려고 했으나 하지 못해서 아쉬웠다. 그러나 정말 많은 공부가 되었다. 헛갈리던 함수 부분과 인터페이스 부분, 처음 써보던 스윙 부분도 공부가 되었다.