

2019 소프트웨어프로젝트

Project 5

FriendList GUI 프로그램



중앙대학교

창의ICT공과대학 소프트웨어학부

20185659 김혜성

목차

Table of contents

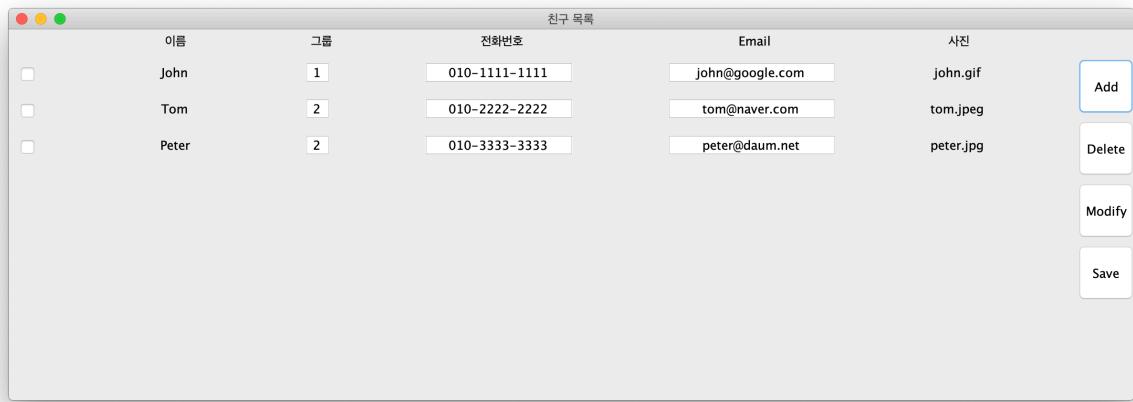
1. Abstract
2. Introduction
3. Proposed Program
 - 3.1. Class diagram
 - 3.2. Analyzation of classes
 - 3.2.1. Friend class
 - 3.2.1.1. Analyzation of the source code
 - 3.2.1.2. Point of view
 - 3.2.2. FriendList class
 - 3.2.2.1. Analyzation of the source code
 - 3.2.2.2. Point of view
 - 3.2.3. FriendListFile class
 - 3.2.3.1. Analyzation of the source code
 - 3.2.3.2. Point of view
 - 3.2.4. FriendListFrame class
 - 3.2.4.1. Analyzation of the source code
 - 3.2.4.2. Point of view
 - 3.2.5. FriendListInformationPanel class
 - 3.2.5.1. Analyzation of the source code
 - 3.2.5.2. Point of view
 - 3.2.6. AddFriendListInformationFrame class
 - 3.2.6.1. Analyzation of the source code
 - 3.2.6.2. Point of view
 4. Program Results
 5. Conclusion
 6. References
 7. Evaluation

1. Abstract

이 프로젝트에서는 프로젝트 3에서 수행했던, FriendList의 GUI 버전을 만들었다. 이는 GUI 화면을 만드는 프로그램으로서, 아래 2가지 화면을 만드는 코드를 작성했다.



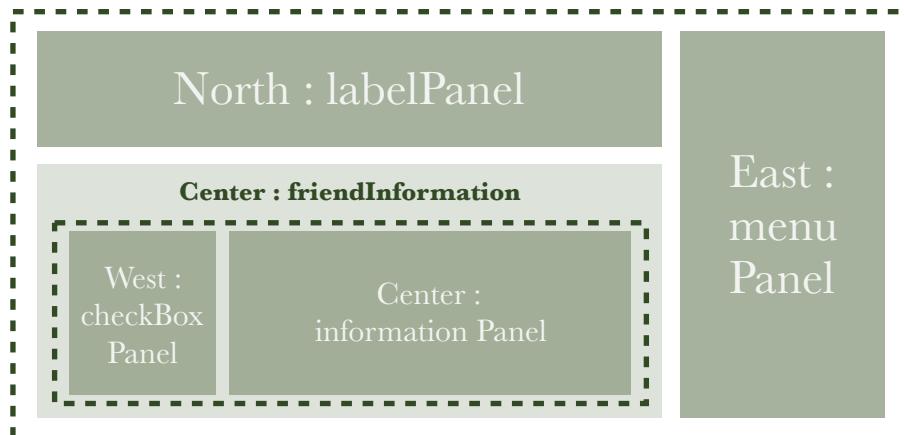
▲ Fig. 1. 추가할 친구 정보 입력 화면



▲ Fig. 2. 친구 목록 정보 화면

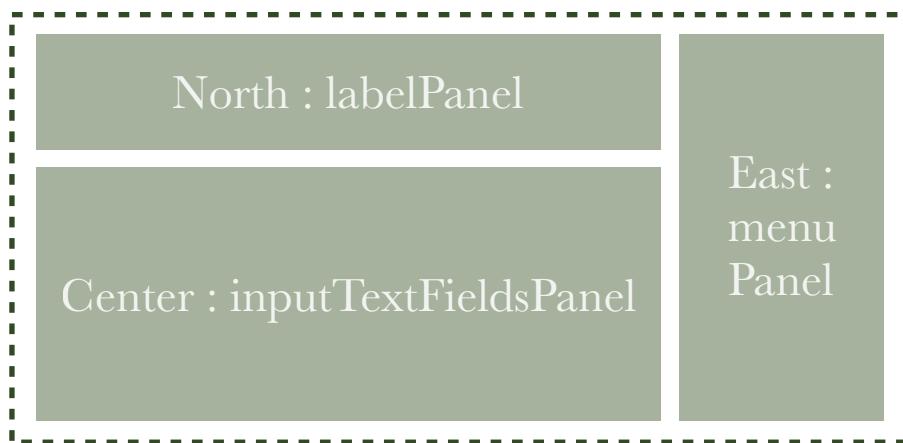
프로젝트 화면의 레이아웃 설정은 다음과 같다.

1. 추가할 친구 정보 입력화면



▲ Fig. 3. Layout of FriendListFrame with FriendListInformationPanel

2. 친구 목록 정보 화면



▲ Fig. 4. Layout of `AddFriendListInformationFrame`

2. Introduction

이 프로젝트의 목적은 다음과 같다.

1. GUI programming 기본

프로젝트의 추진 방법은 다음과 같다.

1. GUI 기본 이해

- 1.1. 대표적 GUI 컴포넌트의 이름, 용도, 기능 이해
- 1.2. JPanel을 이용한 grouping 이해
- 1.3. LayoutManager 이해 및 적합한 용도 파악
- 1.4. JFrame 이해 및 실제 사용 방법 공부
- 1.5. ActionListener의 개념/용도 정도만 이해

2. 프로그래밍 및 점검

이 프로젝트를 진행하면서 panel 생성과 레이아웃 설정과 관련한 시행착오를 통해 GUI programming의 기본을 이해할 수 있었다.

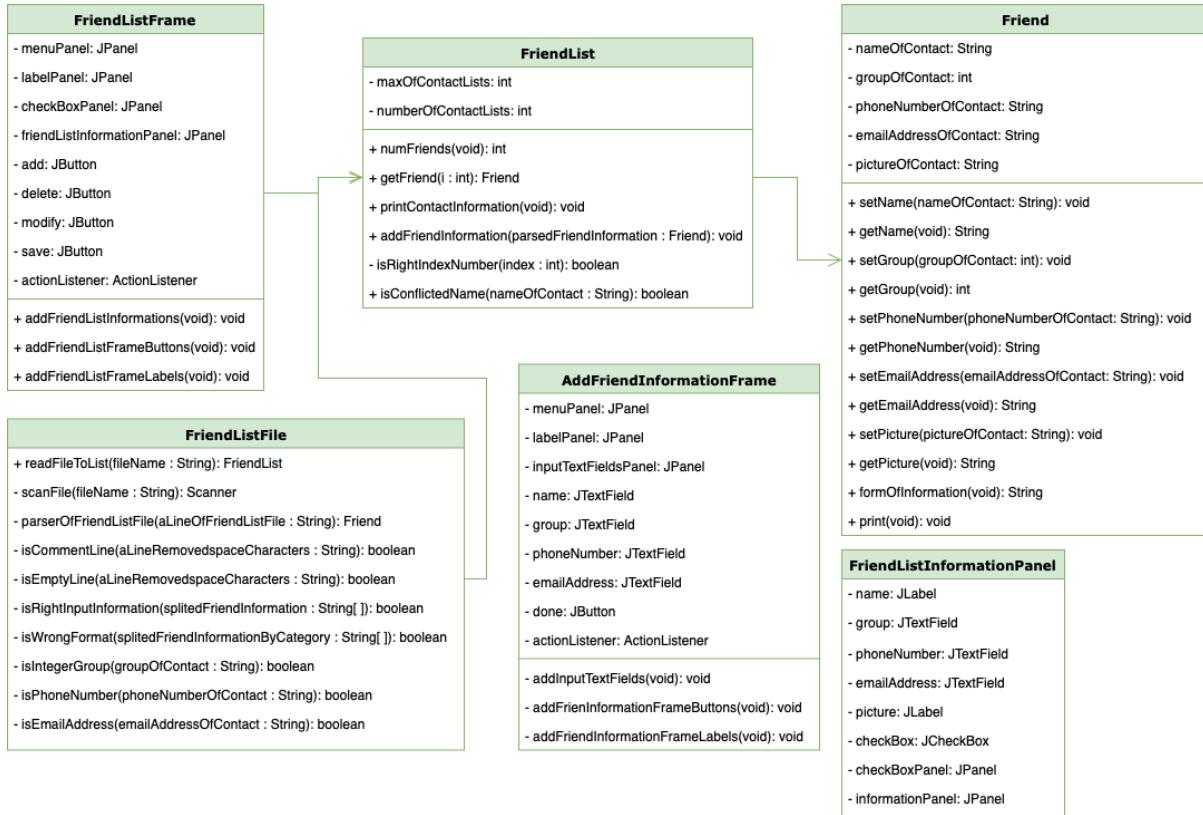
따라서 GUI programming에 있어서 고찰점은 다음과 같다.

1. panel을 생성하여 배치해야 한다.

2. 레이아웃의 충돌에 유의해야 한다.

3. Proposed Program

3.1. Class diagram



▲ Fig. 5. Class diagram

3.2. Analyzation of classes

3.2.1. Friend class

```
1 public class Friend {  
2  
3     // private variables of informations of a contact  
4     private String nameOfContact;  
5     private int groupOfContact;  
6     private String phoneNumberOfContact;  
7     private String emailAddressOfContact;  
8     private String pictureOfContact;  
9  
10    // make setters, getters of private variables of informations of a contact  
11    public void setName(String nameOfContact) {  
12        this.nameOfContact = nameOfContact;  
13    }  
14    public String getName() {  
15        return nameOfContact;  
16    }  
17    public void setGroup(int groupOfContact) {  
18        this.groupOfContact = groupOfContact;  
19    }  
20    public int getGroup() {  
21        return groupOfContact;  
22    }  
23    public void setPhoneNumber(String phoneNumberOfContact) {  
24        this.phoneNumberOfContact = phoneNumberOfContact;  
25    }  
26    public String getPhoneNumber() {  
27        return phoneNumberOfContact;  
28    }  
29    public void setEmailAddress(String emailAddressOfContact) {  
30        this.emailAddressOfContact = emailAddressOfContact;  
31    }  
32    public String getEmailAddress() {  
33        return emailAddressOfContact;  
34    }  
35    public void setPicture(String pictureOfContact) {  
36        this.pictureOfContact = pictureOfContact;  
37    }  
38    public String getPicture() {  
39        return pictureOfContact;  
40    }  
41  
42    // make a form of information of a contact  
43    public String formOfInformation() {  
44        return (nameOfContact + ":" + groupOfContact + ":" + phoneNumberOfContact  
45                + ":" + emailAddressOfContact + ":" + pictureOfContact);  
46    }  
47    public void print() {  
48        System.out.println(this.formOfInformation());  
49    }  
50 }
```

cs

▲ Fig. 6. Source code of the class Friend 1

3.2.3.1. Analyzation of the source code

1-50 : Friend class

- 4-8 : private 항목 변수 선언
- 11-40 : private 변수의 setter, getter 생성
- 43-46 : public formOfInformation method 정의
 - 44-45 : 구분자(“:”)를 포함하여 형식을 맞춤
- 47-49 : public print method 정의
 - 48 : formOfInformation method로 정리된 형식 출력

3.2.3.2. Point of view

1. 구분자(“:"), 5항목으로 contact 정보 private으로 구성

- 1.1. 이름
- 1.2. Group
- 1.3. 전화번호
- 1.4. Email
- 1.5. 사진

3.2.2. FriendList class

```
1 public class FriendList {  
2  
3     private int maxOfContactLists = 100;  
4     private int numberofContactLists = 0;  
5     private Friend[] contactInformation = new Friend[maxOfContactLists];  
6  
7     public int numFriends() {  
8  
9         return numberofContactLists;  
10    }  
11  
12    public Friend getFriend(int i) {  
13  
14        if (isRightIndexNumber(i)) {  
15            return contactInformation[i];  
16        }  
17        else {  
18            return null;  
19        }  
20    }  
21  
22    public void addFriendInformation(Friend parsedFriendInformation) {  
23        contactInformation[numberofContactLists++] = parsedFriendInformation;  
24    }  
25  
26    private boolean isRightIndexNumber(int index) {  
27        if (index >= 0 && index <= 100 && index <= numberofContactLists) {  
28            return true;  
29        }  
30        else {  
31            return false;  
32        }  
33    }  
34  
35    public boolean isConflictedName(String nameOfContact) {  
36        for (int i = 0; i < numberofContactLists; i++) {  
37            if (nameOfContact.equals(contactInformation[i].getNameOfContact())) {  
38                System.out.println("The name of the contact is conflicted.");  
39                return true;  
40            }  
41        }  
42        return false;  
43    }  
44 }
```

cs

▲ Fig. 7. Source code of the class FriendList 1

3.2.2.1. Analyzation of the source code

1-48 : FriendList class

- 5 : contactInformation 배열 객체 생성
- 7-10 : public numFriends method 정의
- 12-20 : public getFriend method 정의
 - 14 : index의 형식이 옳은지 판단

- 22-24 : public addFriendInformation method 정의
- 23 : 파싱된 contact 정보를 contactInformation 배열 객체에 저장
- 26-33 : private isRightIndexNumber method 정의
 - 27 : index 범위 판단
- 35-43 : public isConflictedName method 정의
 - 36-41 : 이전에 contactInformation 배열 객체에 저장된 이름에 같은 이름이 있는지 판단

3.2.2.2. Point of view

1. Friend class를 배열 객체로 생성하였다.
 - 1.1. contact 정보를 배열 객체에 저장하였다.
2. 배열 index 범위의 maximum을 100으로 설정하였다.
 - 2.1. 배열의 index 값 예외처리를 하였다.
3. 이름충돌 예외처리를 객체 지향 개념으로 구현하였다.

3.2.3. FriendListFile class

```
1 public class Friend {  
2  
3     // private variables of informations of a contact  
4     private String nameOfContact;  
5     private int groupOfContact;  
6     private String phoneNumberOfContact;  
7     private String emailAddressOfContact;  
8     private String pictureOfContact;  
9  
10    // make setters, getters of private variables of informations of a contact  
11    public void setNameOfContact(String nameOfContact) {  
12        this.nameOfContact = nameOfContact;  
13    }  
14    public String getNameOfContact() {  
15        return nameOfContact;  
16    }  
17    public void setGroupOfContact(int groupOfContact) {  
18        this.groupOfContact = groupOfContact;  
19    }  
20    public int getGroupOfContact() {  
21        return groupOfContact;  
22    }  
23    public void setPhoneNumberOfContact(String phoneNumberOfContact) {  
24        this.phoneNumberOfContact = phoneNumberOfContact;  
25    }  
26    public String getPhoneNumberOfContact() {  
27        return phoneNumberOfContact;  
28    }  
29    public void setEmailAddressOfContact(String emailAddressOfContact) {  
30        this.emailAddressOfContact = emailAddressOfContact;  
31    }  
32    public String getEmailAddressOfContact() {  
33        return emailAddressOfContact;  
34    }  
35    public void setPictureOfContact(String pictureOfContact) {  
36        this.pictureOfContact = pictureOfContact;  
37    }  
38    public String getPictureOfContact() {  
39        return pictureOfContact;  
40    }  
41  
42    // make a form of information of a contact  
43    public String formOfInformation() {  
44        return (nameOfContact + ":" + groupOfContact + ":" + phoneNumberOfContact  
45                + ":" + emailAddressOfContact + ":" + pictureOfContact);  
46    }  
47    public void print() {  
48        System.out.println(this.formOfInformation());  
49    }  
50 }
```

CS

▲ Fig. 8. Source code of the class Friend 1

3.2.3.1. Analyzation of the source code

1-50 : Friend class

- 4-8 : private 항목 변수 선언
- 11-40 : private 변수의 setter, getter 생성
- 43-46 : public formOfInformation method 정의
 - 44-45 : 구분자(“:”)를 포함하여 형식을 맞춤
- 47-49 : public print method 정의
 - 48 : formOfInformation method로 정리된 형식 출력

3.2.3.2. Point of view

1. 구분자(“:"), 5항목으로 contact 정보 private으로 구성
 - 1.1. 이름
 - 1.2. Group
 - 1.3. 전화번호
 - 1.4. Email
 - 1.5. 사진

3.2.4. FriendListFrame class

```
1 import java.awt.*;
2 import java.awt.event.*;
3 import java.util.ArrayList;
4 import javax.swing.*;
5
6 public class FriendListFrame extends JFrame {
7
8     private FriendList friendList;
9
10    private JPanel menuPanel = new JPanel();
11    private JPanel labelPanel = new JPanel();
12    private JPanel checkBoxPanel = new JPanel();
13    private JPanel friendListInformationPanel = new JPanel();
14
15    private JButton add = new JButton("Add");
16    private JButton delete = new JButton("Delete");
17    private JButton modify = new JButton("Modify");
18    private JButton save = new JButton("Save");
19
20    private ActionListener actionPerformed = new ActionListener() {
21
22        @Override
23        public void actionPerformed(ActionEvent e) {
24            if (e.getSource().equals(add))
25                System.out.println("You pushed the Add button.");
26            else if (e.getSource().equals(delete))
27                System.out.println("You pushed the delete button.");
28            else if (e.getSource().equals(modify))
29                System.out.println("You pushed the Modify button.");
30            else if (e.getSource().equals(save))
31                System.out.println("You pushed the Save button.");
32        }
33    };
34
35    public FriendListFrame(FriendList friendList) {
36
37        this.friendList = friendList;
38        this.setLayout(new BorderLayout());
39        this.setSize(1180, 410);
40        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
41        this.setTitle("친구 목록");
42        this.addFriendListFrameButtons();
43        this.addFriendListFrameLabels();
```

```

44     this.addFriendListInformations();
45     this.setVisible(true);
46 }
47
48 private void addFriendListInformations() {
49
50     ArrayList<FriendListInformationPanel> friend = new ArrayList<FriendListInformationPanel>();
51     for (int i = 0; i < friendList.numFriends(); i++) {
52         Friend friendInformation = friendList.getFriend(i);
53         friendListInformationPanel.add(new FriendListInformationPanel(friendInformation));
54     }
55
56     this.add(friendListInformationPanel, BorderLayout.CENTER);
57
58 }
59
60 private void addFriendListFrameButtons() {
61
62     menuPanel.setLayout(new FlowLayout());
63     menuPanel.setPreferredSize(new Dimension(60, 240));
64
65     menuPanel.add(add, BorderLayout.EAST);
66     add.setPreferredSize(new Dimension(60, 60));
67     add.addActionListener(actionListener);
68
69     menuPanel.add(delete, BorderLayout.EAST);
70     delete.setPreferredSize(new Dimension(60, 60));
71     delete.addActionListener(actionListener);
72
73     menuPanel.add(modify, BorderLayout.EAST);
74     modify.setPreferredSize(new Dimension(60, 60));
75     modify.addActionListener(actionListener);
76
77     menuPanel.add(save, BorderLayout.EAST);
78     save.setPreferredSize(new Dimension(60, 60));
79     save.addActionListener(actionListener);
80
81     this.add(menuPanel, BorderLayout.EAST);
82 }
83
84 public void addFriendListFrameLabels() {
85
86     labelPanel.setLayout(new FlowLayout(FlowLayout.RIGHT, 130, 5));
87     //labelPanel.setPreferredSize(new Dimension(960, 30));

```

```

88     JLabel name = new JLabel("이름");
89     labelPanel.add(name);
90     name.setPreferredSize(new Dimension(23, 15));
91     JLabel group = new JLabel("그룹");
92     labelPanel.add(group);
93     group.setPreferredSize(new Dimension(48, 15));
94     JLabel phoneNumber = new JLabel("전화번호");
95     labelPanel.add(phoneNumber);
96     phoneNumber.setPreferredSize(new Dimension(140, 15));
97     JLabel emailAddress = new JLabel("Email");
98     labelPanel.add(emailAddress);
99     emailAddress.setPreferredSize(new Dimension(90, 15));
100    JLabel picture = new JLabel("사진");
101    labelPanel.add(picture);
102    picture.setPreferredSize(new Dimension(65, 15));
103
104    this.add(labelPanel, BorderLayout.NORTH);
105
106 }

```

CS

▲ Fig. 9. Source code of FriendListFrame class

2.2.4.1. Analyzation of the source code

1 - 4 : 외부 class 추가

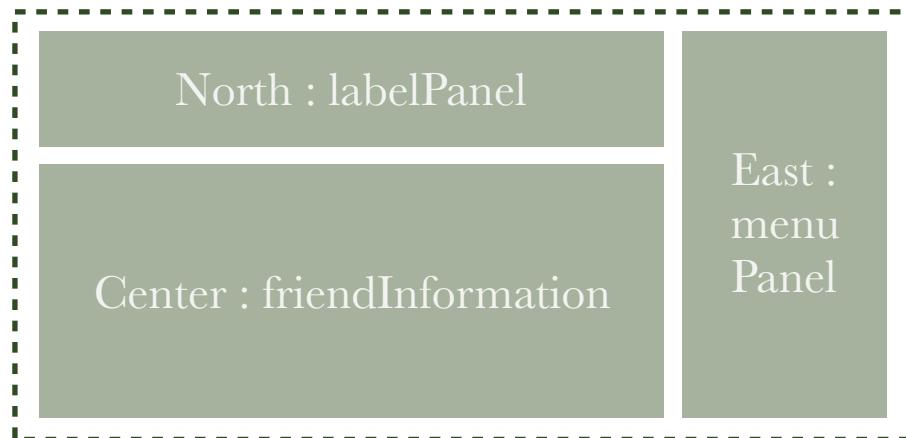
6 - 120 : FriendListFrame class (JFrame class 상속)

- 8 : FriendList 객체 선언 (addFriendListInformations() method에서 객체 반환)
- 10 - 13 : JPanel 객체 생성
- 15 - 18 : JButton 객체 생성
- 20 - 33 : ActionListener 객체 생성
 - 22 : overriding
 - 24 - 25 : “Add” button을 누르면 콘솔창에 메세지 출력
 - 26 - 27 : “Delete” button을 누르면 콘솔창에 메세지 출력
 - 28 - 29 : “Modify” button을 누르면 콘솔창에 메세지 출력
 - 30 - 31 : “save” button을 누르면 콘솔창에 메세지 출력
- 35 - 46 : FriendListFrame 객체 생성자 (파라미터 : friendList)
 - 37 : 생성자의 파라미터(friendList)를 FriendListFrame의 friendList에 저장
 - 38 : 레이아웃 설정 (border)
 - 39 : 화면 크기 설정
 - 40 : 화면 닫기 설정
 - 41 : 화면 title 설정
 - 42 : addFriendListButtons() method (FriendListFrame의 buttons 생성) 호출
 - 43 : addFriendListLabels() method (FriendListFrame의 labels 생성) 호출
 - 44 : addFriendListInformations() method (FriendListFrame에 정보 생성) 호출
- 48 - 60 : addFriendListInformations() method 생성
 - 50 : FriendListInformationPanel class에서 friend를 받아와 arrayList에 저장

- 52 - 58 : arrayList의 friend를 friendListFrame에 생성 (위치는 border의 Center)
- 62 - 82 : addFriendListFrameButtons() method 생성
 - 64 : menuPanel 레이아웃 설정 (flow)
 - 65 : menuPanel 크기 설정
 - 67 - 69 : menuPanel에 add button 생성
 - 70 - 72 : menuPanel에 delete button 생성
 - 73 - 75 : menuPanel에 modify button 생성
 - 76 - 78 : menuPanel에 save button 생성
- 80 : menuPanel을 friendListFrame에 생성 (위치는 border의 East)
- 84 - 106 : addFriendListFrameLabels() method 생성
 - 86 : labelPanel 레이아웃 설정 (flow)
 - 88 - 90 : labelPanel에 name label 생성
 - 91 - 93 : labelPanel에 group label 생성
 - 94 - 96 : labelPanel에 phoneNumber label 생성
 - 97 - 99 : labelPanel에 emailAddress label 생성
 - 100 - 102 : labelPanel에 picture label 생성
- 104 : labelPanel을 friendListFrame에 생성 (위치는 border의 North)

2.2.4.2. Point of view

1. friendListFrame의 레이아웃은 다음과 같다.



▲ Fig. 10. Layout of FriendListFrame

3.2.5. FriendListInformationPanel class

```
1 import java.awt.*;
2 import javax.swing.*;
3 import javax.swing.border.Border;
4
5 public class FriendListInformationPanel extends JPanel {
6
7     private JLabel name = new JLabel();
8     private JTextField group = new JTextField();
9     private JTextField phoneNumber = new JTextField();
10    private JTextField emailAddress = new JTextField();
11    private JLabel picture = new JLabel();
12    private JCheckBox checkBox = new JCheckBox();
13
14    private JPanel checkBoxPanel = new JPanel();
15    private JPanel informationPanel = new JPanel();
16
17    public FriendListInformationPanel(Friend friend) {
18
19        this.setLayout(new BorderLayout());
20        checkBoxPanel.setLayout(new FlowLayout(FlowLayout.LEFT));
21        informationPanel.setLayout(new FlowLayout(FlowLayout.LEFT, 95, 5));
22        //informationPanel.setPreferredSize(new Dimension(930, 30));
23
24        //checkBox.setPreferredSize(new Dimension(25, 20));
25        checkBoxPanel.add(checkBox);
26
27        name.setText(friend.getName());
28        informationPanel.add(name);
29        name.setPreferredSize(new Dimension(80, 20));
30        name.setHorizontalAlignment(JTextField.CENTER);
31        group.setText(friend.getGroup() + "''");
32        informationPanel.add(group);
33        group.setPreferredSize(new Dimension(30, 20));
34        group.setHorizontalAlignment(JTextField.CENTER);
35        phoneNumber.setText(friend.getPhoneNumber());
36        informationPanel.add(phoneNumber);
37        phoneNumber.setPreferredSize(new Dimension(160, 20));
38        phoneNumber.setHorizontalAlignment(JTextField.CENTER);
39        emailAddress.setText(friend.getEmailAddress());
40        informationPanel.add(emailAddress);
41        emailAddress.setPreferredSize(new Dimension(180, 20));
42        emailAddress.setHorizontalAlignment(JTextField.CENTER);
43        picture.setText(friend.getPicture());
```

```

44     informationPanel.add(picture);
45     picture.setPreferredSize(new Dimension(60, 20));
46     picture.setHorizontalAlignment(JTextField.CENTER);
47
48     this.add(checkBoxPanel, BorderLayout.WEST);
49     this.add(informationPanel, BorderLayout.CENTER);
50
51 }
52 }
```

CS

▲ Fig. 11. Source code of FriendInformationPanel class

3.2.5.1. Analyzation of the source code

1 - 3 : 외부 class 추가

5 - 52 : FriendListInformationPanel class (JPanel class 상속)

7 : JLabel 객체 생성

8 - 10 : JTextField 객체 생성

11 : JLabel 객체 생성

12 : JCheckBox 객체 생성

14 - 15 : JPanel 객체 생성

17 - 52 : FriendListInformationPanel 객체 생성자(파라미터 : friend)

- 19 : 전체 레이아웃 설정 (border)

- 20 : checkBoxPanel 레이아웃 설정 (flow)

- 21 : informationPanel 레이아웃 설정 (flow)

- 25 : checkBoxPanel에 checkBox 생성

- 27 - 30 : informationPanel에 name textField 생성 (textField Center 정렬)

- 31 - 34 : informationPanel에 group textField 생성 (textField Center 정렬)

- 35 - 38 : informationPanel에 phoneNumber textField 생성 (textField Center 정렬)

- 39 - 42 : informationPanel에 emailAddress textField 생성 (textField Center 정렬)

- 43 - 46 : informationPanel에 picture textField 생성 (textField Center 정렬)

- 48 : checkBoxPanel을 friendListInformationPanel에 생성 (위치는 border의 West)

- 49 : informationPanel을 friendListInformationPanel에 생성 (위치는 border의 Center)

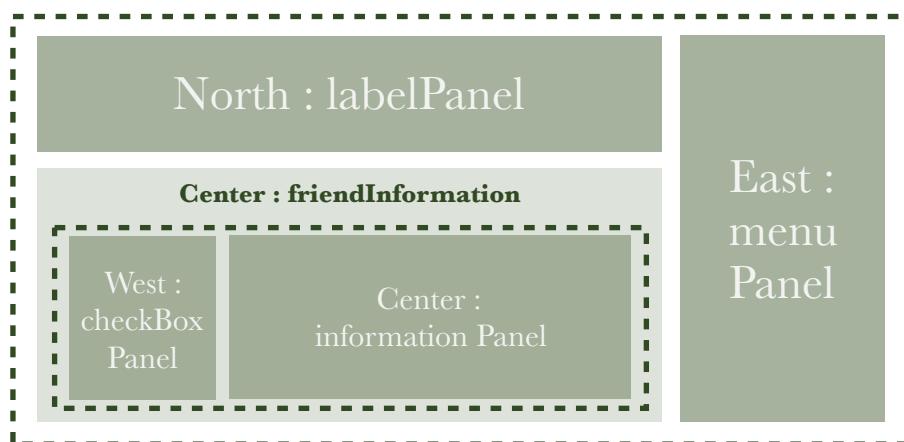
2.2.5.2. Point of view

1. **FriendListInformationPanel**의 레이아웃은 다음과 같다.



▲ Fig. 12. Layout of FriendListInformationPanel

2. 이를 **friendListFrame**에 적용하면 다음과 같다.



▲ Fig. 13. Layout of FriendListFrame with FriendListInformationPanel

3.2.6. AddFriendListInformationFrame class

```
1 import java.awt.*;
2 import java.awt.event.*;
3 import javax.swing.*;
4
5 public class AddFriendInformationFrame extends JFrame {
6
7     private JPanel menuPanel = new JPanel();
8     private JPanel labelPanel = new JPanel();
9     private JPanel inputTextFieldsPanel = new JPanel();
10
11    private JTextField name = new JTextField();
12    private JTextField group = new JTextField();
13    private JTextField phoneNumber = new JTextField();
14    private JTextField emailAddress = new JTextField();
15    private JTextField picture = new JTextField();
16
17    private JButton done = new JButton("Done");
18
19    private ActionListener actionListener = new ActionListener() {
20
21        @Override
22        public void actionPerformed(ActionEvent e) {
23            if (e.getSource().equals(done))
24                System.out.println("You pushed the Done button.");
25        }
26    };
27
28    public AddFriendInformationFrame() {
29
30        this.setLayout(new BorderLayout());
31        this.setSize(1020, 108);
32        this.setTitle("추가될 친구 정보");
33        this.addFriendInformationFrameButtons();
34        this.addFriendInformationFrameLabels();
35        this.addInputTextFields();
36        this.setVisible(true);
37    }
38
39    private void addInputTextFields() {
40
41        inputTextFieldsPanel.setLayout(new FlowLayout(FlowLayout.LEFT, 5, 20));
42        inputTextFieldsPanel.setPreferredSize(new Dimension(960, 60));
43    }
}
```

```
44         inputTextFieldsPanel.add(name);
45         name.setPreferredSize(new Dimension(185, 20));
46         name.setHorizontalAlignment(JTextField.CENTER);
47         inputTextFieldsPanel.add(group);
48         group.setPreferredSize(new Dimension(185, 20));
49         group.setHorizontalAlignment(JTextField.CENTER);
50         inputTextFieldsPanel.add(phoneNumber);
51         phoneNumber.setPreferredSize(new Dimension(185, 20));
52         phoneNumber.setHorizontalAlignment(JTextField.CENTER);
53         inputTextFieldsPanel.add(emailAddress);
54         emailAddress.setPreferredSize(new Dimension(185, 20));
55         emailAddress.setHorizontalAlignment(JTextField.CENTER);
56         inputTextFieldsPanel.add(picture);
57         picture.setPreferredSize(new Dimension(185, 20));
58         picture.setHorizontalAlignment(JTextField.CENTER);
59
60         this.add(inputTextFieldsPanel, BorderLayout.CENTER);
61
62     }
63
64
65     private void addFriendInformationFrameButtons() {
66
67         menuPanel.setLayout(new FlowLayout(FlowLayout.CENTER));
68         menuPanel.setPreferredSize(new Dimension(65,60));
69
70         menuPanel.add(done);
71         done.setPreferredSize(new Dimension(60,60));
72         done.addActionListener(actionListener);
73
74         this.add(menuPanel, BorderLayout.EAST);
75     }
76
77
78     private void addFriendInformationFrameLabels() {
79
80         labelPanel.setLayout(new FlowLayout(FlowLayout.RIGHT,100,5));
81         labelPanel.setPreferredSize(new Dimension(960, 20));
82
83         JLabel name = new JLabel("이름");
84         name.setPreferredSize(new Dimension(90, 15));
85         labelPanel.add(name);
86         JLabel group = new JLabel("그룹");
87         group.setPreferredSize(new Dimension(85, 15));
88         labelPanel.add(group);
```

cs

```

88     JLabel phoneNumber = new JLabel("전화번호");
89     phoneNumber.setPreferredSize(new Dimension(95, 15));
90     labelPanel.add(phoneNumber);
91     JLabel emailAddress = new JLabel("Email");
92     emailAddress.setPreferredSize(new Dimension(90, 15));
93     labelPanel.add(emailAddress);
94     JLabel picture = new JLabel("사진");
95     picture.setPreferredSize(new Dimension(75, 15));
96     labelPanel.add(picture);
97
98     this.add(labelPanel, BorderLayout.NORTH);
99 }
100 }
```

cs

▲ Fig. 14. Source code of AddFriendListInformationFrame class

3.2.6.1. Analyzation of the source code

1 - 3 : 외부 class 추가

5 - 100 : AddFriendInformationFrame class (JPanel 상속)

- 7 - 9 : JPanel 객체 생성

- 11 - 15 : JTextField 객체 생성

- 17 : JButton 객체 생성

- 19 - 26 : ActionListener 객체 생성

 - 21 : overriding

 - 22 - 25 : “Done” button을 누르면 콘솔창에 메세지 출력

- 28 - 37 : AddFriendInformationFrame() 객체 생성자

 - 30 : 레이아웃 설정 (border)

 - 31 : 화면 크기 설정

 - 32 : 화면 title 설정

 - 33 : addFriendInformationFrameButtons() method (FriendInformationFrame의 buttons 생성)

성) 호출

- 34 : addFriendInformationFrameLabels() method (FriendInformationFrame의 labels 생성)

호출

- 35 : addInputTextFields() method (FriendInformationFrame의 textFields 생성) 호출

- 39 - 63 : addInputTextFields() method

 - 41 : inputTextFieldsPanel 레이아웃 설정 (flow)

 - 45 - 47 : inputTextFieldsPanel의 name textPanel 생성 (textField Center 정렬)

 - 48 - 50 : inputTextFieldsPanel의 group textPanel 생성 (textField Center 정렬)

 - 51 - 53 : inputTextFieldsPanel의 phoneNumber textPanel 생성 (textField Center 정렬)

 - 54 - 56 : inputTextFieldsPanel의 emailAddress textPanel 생성 (textField Center 정렬)

 - 57 - 59 : inputTextFieldsPanel의 picture textPanel 생성 (textField Center 정렬)

 - 61 : inputTextFieldsPanel을 AddFriendInformationFrame에 생성 (위치는 border의 Center)

- 65 - 75 : addFriendInformationFrameButtons() method

 - 67 : menuPanel 레이아웃 설정 (flow)

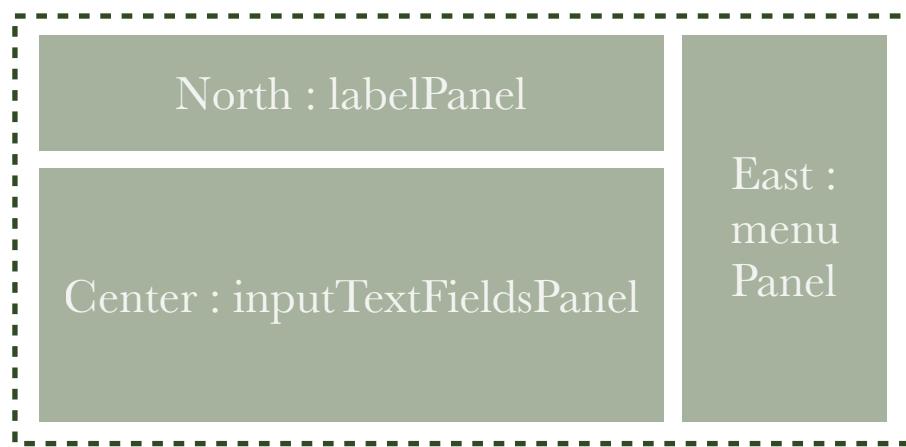
 - 70 - 72 : menuPanel에 done button 생성

 - 74 : menuPanel을 addFriendInformationFrame에 생성 (위치는 border의 East)

- 77 - 99 : addFriendInformationFrameLabel() method
- 79 : labelPanel 레이아웃 설정 (flow)
- 82 - 84 : labelPanel에 name label 생성
- 85 - 87 : labelPanel에 group label 생성
- 88 - 90 : labelPanel에 phoneNumber label 생성
- 91 - 93 : labelPanel에 emailAddress label 생성
- 94 - 96 : labelPanel에 picture label 생성
- 98 : labelPanel을 AddFriendInformationFrame에 생성 (위치는 border의 North)

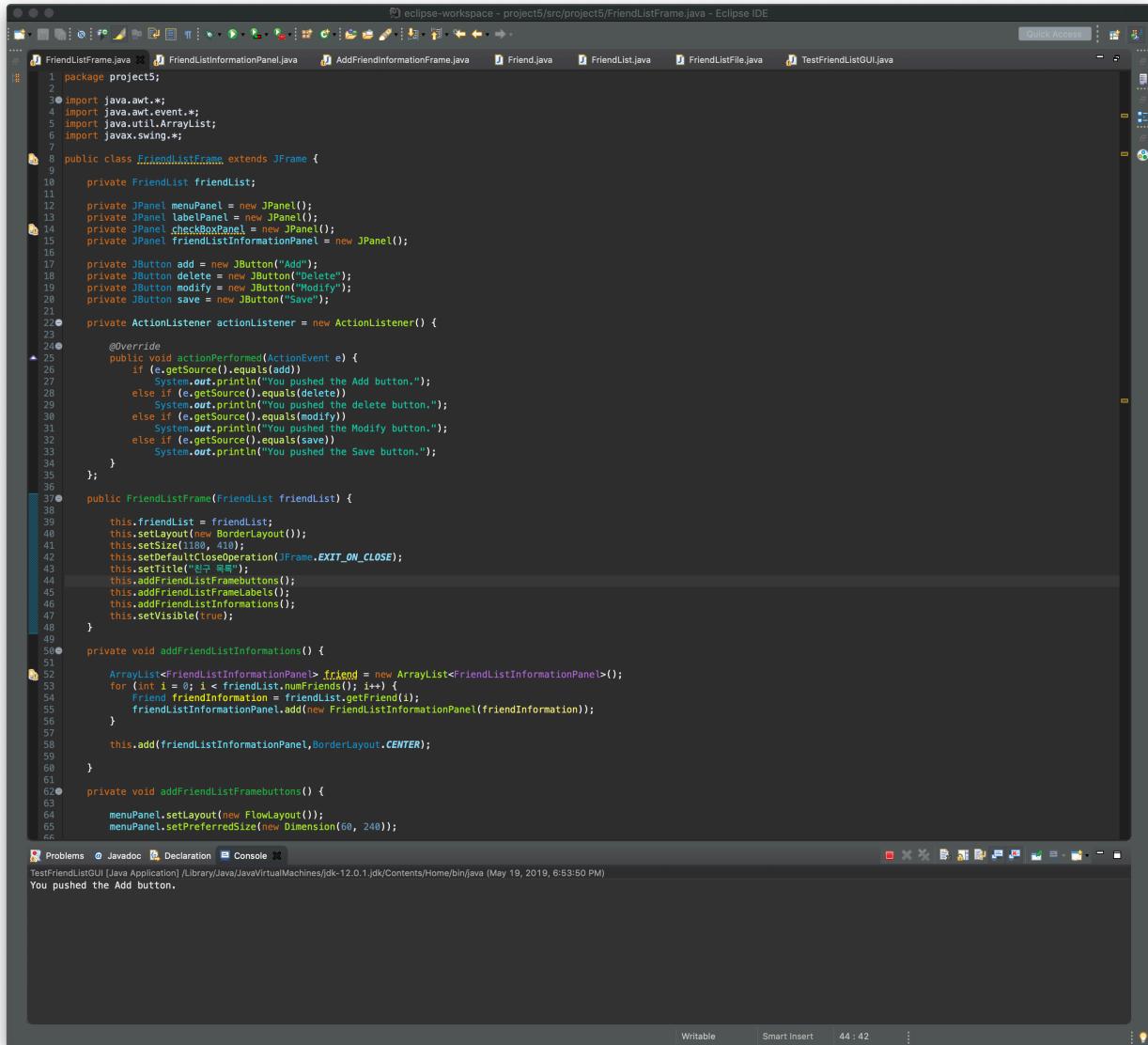
3.2.6.2. Point of view

1. AddFriendListInformationFrame의 레이아웃은 다음과 같다.



▲ Fig. 15. Layout of AddFriendListInformationFrame

4. Program Results



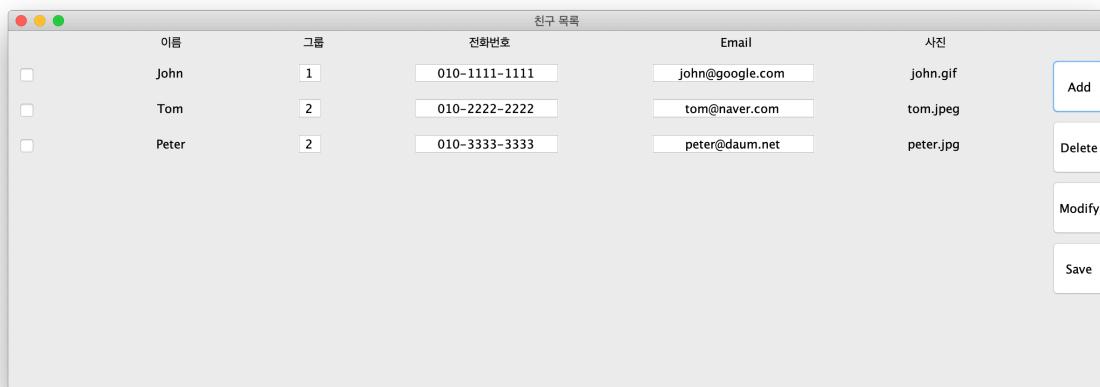
```

1 package project5;
2
3 import java.awt.*;
4 import java.awt.event.*;
5 import java.util.ArrayList;
6 import javax.swing.*;
7
8 public class FriendListFrame extends JFrame {
9
10    private FriendList friendList;
11
12    private JPanel menuPanel = new JPanel();
13    private JPanel labelPanel = new JPanel();
14    private JPanel checkBoxPanel = new JPanel();
15    private JPanel friendListInformationPanel = new JPanel();
16
17    private JButton add = new JButton("Add");
18    private JButton delete = new JButton("Delete");
19    private JButton modify = new JButton("Modify");
20    private JButton save = new JButton("Save");
21
22    private ActionListener actionListener = new ActionListener() {
23
24        @Override
25        public void actionPerformed(ActionEvent e) {
26            if (e.getSource().equals(add))
27                System.out.println("You pushed the Add button.");
28            else if (e.getSource().equals(delete))
29                System.out.println("You pushed the delete button.");
30            else if (e.getSource().equals(modify))
31                System.out.println("You pushed the Modify button.");
32            else if (e.getSource().equals(save))
33                System.out.println("You pushed the Save button.");
34        }
35    };
36
37    public FriendListFrame(FriendList friendList) {
38
39        this.friendList = friendList;
40        this.setLayout(new BorderLayout());
41        this.setSize(180, 410);
42        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
43        this.setTitle("FriendListFrame");
44        this.addFriendListFrameButtons();
45        this.addFriendListFrameLabels();
46        this.addFriendListInformations();
47        this.setVisible(true);
48    }
49
50    private void addFriendListInformations() {
51
52        ArrayList<FriendListInformationPanel> friend = new ArrayList<FriendListInformationPanel>();
53        for (int i = 0; i < friendList.numFriends(); i++) {
54            friendInformation = friendList.getFriend(i);
55            friendListInformationPanel.add(new FriendListInformationPanel(friendInformation));
56        }
57
58        this.add(friendListInformationPanel, BorderLayout.CENTER);
59    }
60
61    private void addFriendListFrameButtons() {
62
63        menuPanel.setLayout(new FlowLayout());
64        menuPanel.setPreferredSize(new Dimension(60, 240));
65

```

The screenshot shows the Eclipse IDE interface with the code editor open. The code is for a Java Swing application named 'FriendListFrame'. It includes imports for Java AWT, Event, Util, and Swing packages. The class defines a menu panel, label panel, check box panel, and friend list information panel. It contains four buttons: 'Add', 'Delete', 'Modify', and 'Save'. An 'actionListener' is defined to handle button events, printing the source of each button push to the console. The constructor initializes the frame with a title, sets its layout to BorderLayout, and adds the four panels and buttons. The 'addFriendListInformations' method adds three friend information panels to the center of the frame. The 'addFriendListFrameButtons' method sets up the menu panel with a flow layout and a preferred size of 60x240.

▲ Fig. 16. Test in Eclipse 1



	이름	그룹	전화번호	Email	사진	
<input type="checkbox"/>	John	1	010-1111-1111	john@google.com	john.gif	Add
<input type="checkbox"/>	Tom	2	010-2222-2222	tom@naver.com	tom.jpeg	
<input type="checkbox"/>	Peter	2	010-3333-3333	peter@daum.net	peter.jpg	Modify

The screenshot shows a Java application window titled '친구 목록' (Friend List). It displays a table of friends with columns: 이름 (Name), 그룹 (Group), 전화번호 (Phone Number), Email, and 사진 (Photo). There are three rows of data: John (Group 1, Phone 010-1111-1111, Email john@google.com, Photo john.gif), Tom (Group 2, Phone 010-2222-2222, Email tom@naver.com, Photo tom.jpeg), and Peter (Group 2, Phone 010-3333-3333, Email peter@daum.net, Photo peter.jpg). On the right side of the table, there are four buttons: 'Add', 'Delete', 'Modify', and 'Save'. The 'Add' button is currently highlighted.

▲ Fig. 17. Test in Eclipse 2



▲ Fig. 18. Test in Eclipse 3

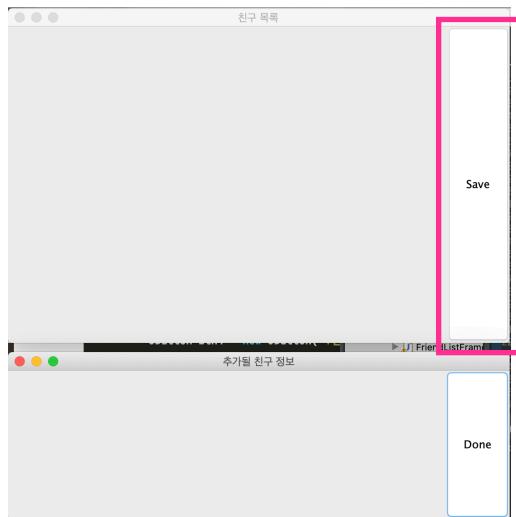
```

1 public class TestFriendListGUI {
2
3     public static void main(String[] args) {
4         // TODO Auto-generated method stub
5
6         FriendListFile friendListFile = new FriendListFile();
7         FriendList friendList = friendListFile.readFileToList("friendlist-norm.data");
8
9         FriendListFrame friendListFrame = new FriendListFrame(friendList);
10        AddFriendInformationFrame addfriendinformationframe = new AddFriendInformationF
11        rame();
12
13    }
14
15 }
```

▲ Fig. 19. Source code of the class project4Test

TestFriendListGUI main class를 작성하여 FriendListFrame과 AddFriendInforamtionFrame을 테스트 했다.
이때의 실행 창은 다음과 같다.

1. Button들이 겹쳐서 보이지 않았다.



▲ Fig. 17. Trial and error 1

이 시행착오는 button을 새로운 panel 위에 올려두지 않고 전체 panel의 east에 배치하려고 해서 겹쳐진 것이었다. 따라서 오른쪽 button을 위한 새로운 menuPanel을 만들어 그 위에 배치하였다. 그 결과로, Fig. 17. 와 같이 button들이 겹치지 않고 배치되었다.

2. 레이아웃의 충돌이 일어나 실행되지 않았다.

```

private void addInputTextField() {
    inputTextFieldsPanel.setLayout(new FlowLayout(FlowLayout.LEFT, 95, 5));
    inputTextFieldsPanel.setPreferredSize(new Dimension(560, 60));
    inputTextFieldsPanel.add(name);
    name.setPreferredSize(new Dimension(185, 20));
    name.setHorizontalTextPosition(FlowLayout.CENTER);
    inputTextFieldsPanel.add(phoneNumber);
    phoneNumber.setPreferredSize(new Dimension(185, 20));
    phoneNumber.setHorizontalTextPosition(FlowLayout.CENTER);
    inputTextFieldsPanel.add(emailAddress);
    emailAddress.setPreferredSize(new Dimension(185, 20));
    emailAddress.setHorizontalTextPosition(FlowLayout.CENTER);
    inputTextFieldsPanel.add(picture);
    picture.setPreferredSize(new Dimension(185, 20));
    picture.setHorizontalTextPosition(FlowLayout.CENTER);
    this.add(inputTextFieldsPanel, FlowLayout.LEFT);
}

private void addFriendInformationFrameButton() {
    menuPanel.setLayout(new FlowLayout(FlowLayout.RIGHT));
    menuPanel.setPreferredSize(new Dimension(60, 60));
    menuPanel.add(done);
    done.setPreferredSize(new Dimension(60, 60));
    done.addActionListener(actionListener);
    this.add(menuPanel, FlowLayout.RIGHT);
}

private void addFriendInformationFrameLabels() {
    labelPanel.setLayout(new FlowLayout(FlowLayout.LEFT, 120, 5));
    labelPanel.setPreferredSize(new Dimension(560, 30));
    JLabel name = new JLabel("이름");
    labelPanel.add(name);
    name.setPreferredSize(new Dimension(60, 15));
    JLabel group = new JLabel("그룹");
    labelPanel.add(group);
    group.setPreferredSize(new Dimension(60, 15));
    JLabel phoneNumber = new JLabel("전화번호");
    labelPanel.add(phoneNumber);
    phoneNumber.setPreferredSize(new Dimension(60, 15));
    JLabel emailLabel = new JLabel("Email");
    labelPanel.add(emailAddress);
    emailAddress.setPreferredSize(new Dimension(60, 15));
    JLabel picture = new JLabel("사진");
    labelPanel.add(picture);
    picture.setPreferredSize(new Dimension(60, 15));
    this.add(labelPanel, FlowLayout.LEFT);
}

```

```

Exception in thread "main" java.lang.IllegalComponentStateException: illegal component position
        at java.desktop/java.awt.Container.addImpl(Container.java:1011)
        at java.desktop/java.awt.Container.add(Container.java:1029)
        at java.desktop/java.awt.Container.addImpl(Container.java:553)
        at java.desktop/java.awt.Container.add(Container.java:489)
        at projects.AddFriendInformationFrame.addFriendInformationFrameButtons(AddFriendInformationFrame.java:75)
        at projects.AddFriendInformationFrame.main(AddFriendInformationFrame.java:35)
        at projects.TestFriendListGUI.main(TestFriendListGUI.java:12)

```

▲ Fig. 20. Trial and error 1

이 시행착오는 기본적인 실수이다. panel을 생성할 때 설정했던 layout과 같은 형태로 그 panel을 배치해야하는데 위에서 전체 panel을 border로 설정하고 세부 panel들을 그 위에 배치할 때 flow로 설정하여 충돌이 일어났던 것이다. 이를 Fig. 15. 와 같이 border로 설정했더니 정상적으로 작동하였다.

5. Conclusion

이 프로젝트에서는 프로젝트 3에서 수행했던, FriendList의 GUI 버전을 만들었다. 이는 GUI 화면을 만드는 프로그램으로서, 아래 2가지 화면을 만드는 코드를 작성했다.

프로젝트의 추진 방법은 다음과 같다.

1. GUI 기본 이해

- 1.1. 대표적 GUI 컴포넌트의 이름, 용도, 기능 이해
- 1.2. JPanel을 이용한 grouping 이해
- 1.3. LayoutManager 이해 및 적합한 용도 파악
- 1.4. JFrame 이해 및 실제 사용 방법 공부
- 1.5. ActionListener의 개념/용도 정도만 이해

2. 프로그래밍 및 점검

이 프로젝트를 진행하면서 panel 생성과 레이아웃 설정과 관련한 시행착오를 통해 GUI programming의 기본을 이해할 수 있었다.

따라서 GUI programming에 있어서 고찰점은 다음과 같다.

1. panel을 생성하여 배치해야 한다.

2. 레이아웃의 층들에 유의해야 한다.

6. References

[1] 남궁 성(2008). Java의 정석. 도우출판

7. Evaluation

평가 항목	학생 자체 평가 (리포트 해당 부분 표시 및 간단한 의견)	평가 (빈칸)	점수 (빈칸)
완성도 (동작 여부) - GUI 화면 완성도 - 클릭 시 동작	- 작성한 GUI 화면이 요구사항을 충족하였다. - Button 클릭 시 콘솔창에 메세지가 출력된다.		
설계 노트 - 클래스 구조 - GUI 설계 시 착안 사항 - 시행착오 내용 및 해결방안	- Class diagram 사진을 첨부하였다. - 설계 시 착안 사항을 모두 만족했다. - 시행착오 내용을 사진을 첨부하여 설명하고 해결방안을 서술하였다.		
리포트 - 평가자 시작으로 리포트 검토 - 위의 평가 요소들이 명확하게 기술되었는가? - 기타 건의 사항	- 평가자 시작에서 소스코드와 구조, 분석이 명확하게 작성하였다. - 위의 평가 요소들이 명확하게 기술되었다. - github 로그 (https://github.com/AllyHyeongKim/softwareProjectJava/tree/master/project5)		
총평/계	평가 항목을 모두 만족하였으며, 직접 프로그래밍하였음을 사진을 통해 입증하였다.		