



소프트웨어프로젝트 02분반

project6 report

소프트웨어학부

20181269 박현우

I. 소스 프로그램

<Friend.java> - project 3

```
public class Friend {
    String name;
    int group;
    String phone;
    String email;
    String picture;

    public Friend() {
        name = getName();
        group = getGroup();
        phone = getPhone();
        email = getEmail();
        picture = getPicture();
    }

    public String getName() {
        return this.name;
    }

    public int getGroup() {
        return this.group;
    }

    public String getPhone() {
        return this.phone;
    }

    public String getEmail() {
        return this.email;
    }

    public String getPicture() {
        return this.picture;
    }

    public void print() {
        System.out.println(this.name + " : " + this.group + " : " + this.phone + " : "
+ this.email + " : " + this.picture);
    }
}
```

<FriendList.java> - project 3

```
public class FriendList {
    Friend[] friends = new Friend[100];

    private FriendList() {
        for(int i=0 ; i<100; i++)
            friends[i] = new Friend();
    }

    static FriendList friendsList = new FriendList();

    public static FriendList friendListGenerator() {
        return friendsList;
    }

    public int numFriends() {
        int count = 0;
        for (int i=0 ; i<100 ; i++)
            if (friends[i].name != null)
                count++;
        return count;
    }

    public Friend getFriend(int i) {
        return friends[i];
    }
}
```

```

    }
}

```

<FriendListFile.java> - project 3

```

import java.io.File;
import java.io.FileNotFoundException;
import java.util.Scanner;
import java.lang.String;
import java.lang.ArrayIndexOutOfBoundsException;
import java.lang.NumberFormatException;

public class FriendListFile {

    public FriendListFile() {
    }

    public FriendList readFileToList(String fileName) {
        FriendList friendsList = FriendList.friendListGenerator();

        try {
            File file = new File(fileName);
            Scanner input = new Scanner(file);
            int idx = 0;
            while (input.hasNext()) {
                String line = input.nextLine();
                if (!(line.startsWith("//"))) {
                    String[] token = line.split(":");
                    for(int i=0; i<token.length; i++) {
                        token[i] = token[i].trim();
                    }
                    boolean bool1=true, bool2=true;
                    String regex =
"^[_a-zA-Z0-9-\\.]+@[\\.a-zA-Z0-9-\\.]+\\.([a-zA-Z]+)$";
                    for (int i=0; i<idx; i++) {
                        if (token[0].equals(friendsList.friends[i].name)) {
                            System.out.println("Name Conflict");
                            bool1 = false;
                        }
                    }
                    if (!(token[3].matches(regex))) {
                        System.out.println("Illegal email
address");
                        bool2 = false;
                    }
                    if (bool1 && bool2) {
                        int token1 = Integer.parseInt(token[1]);
                        friendsList.friends[idx].name = token[0];
                        friendsList.friends[idx].group = token1;
                        friendsList.friends[idx].phone = token[2];
                        friendsList.friends[idx].email = token[3];
                        friendsList.friends[idx].picture = token[4];
                        idx++;
                    }
                }
                input.close();
            }
            catch(FileNotFoundException e) {
                System.out.println("Unknown File");
            }
            catch(ArrayIndexOutOfBoundsException e) {
                System.out.println("Irregular input line");
            }
            catch(NumberFormatException e) {
                System.out.println("Illegal value");
            }
            return friendsList;
        }
    }
}

```

<FriendListGUI.java>

```
import java.awt.*;
import javax.swing.*;
import java.awt.event.*;
import java.io.*;

public class FriendListGUI extends JFrame {
    public FriendListGUI(String fileName) {
        int[] count = {0};
        FriendList friendsList = FriendList.friendListGenerator();
        FriendListFile flf = new FriendListFile();
        flf.readFileToList(fileName);

        setTitle("친구 목록");
        setSize(800,400);

        JPanel panel = new JPanel();
        JPanel buttons = new JPanel();
        JPanel friends = new JPanel();

        add(panel, BorderLayout.CENTER);
        panel.setLayout(new BorderLayout());
        panel.add(friends, BorderLayout.NORTH);
        friends.add(new JLabel(" "));
        friends.add(new JLabel("이름"));
        friends.add(new JLabel("그룹"));
        friends.add(new JLabel("전화번호"));
        friends.add(new JLabel("Email"));
        friends.add(new JLabel("사진"));
        JCheckBox[] cbox = new JCheckBox[100];
        JLabel[] name = new JLabel[100];
        JTextField[][] info = new JTextField[100][4];
        for (int i=0; i<100; i++)
            cbox[i] = new JCheckBox();
        while ( friendsList.friends[count[0]].name != null ) {
            name[count[0]] = new JLabel(friendsList.friends[count[0]].name);
            info[count[0]][0] = new
JTextField(Integer.toString(friendsList.friends[count[0]].group));
            info[count[0]][1] = new JTextField(friendsList.friends[count[0]].phone);
            info[count[0]][2] = new JTextField(friendsList.friends[count[0]].email);
            info[count[0]][3] = new JTextField(friendsList.friends[count[0]].picture);
            friends.add(cbox[count[0]]);
            friends.add(name[count[0]]);
            friends.add(info[count[0]][0]);
            friends.add(info[count[0]][1]);
            friends.add(info[count[0]][2]);
            friends.add(info[count[0]][3]);
            count[0]++;
        }
        friends.setLayout(new GridLayout(count[0]+1,6));

        add(buttons, BorderLayout.EAST);
        buttons.setLayout(new GridLayout(4,1));
        JButton addbtn = new JButton("Add");
        JButton delbtn = new JButton("Delete");
        JButton modbtn = new JButton("Modify");
        JButton savebtn = new JButton("Save File");
        buttons.add(addbtn);
        buttons.add(delbtn);
        buttons.add(modbtn);
        buttons.add(savebtn);

        addbtn.addActionListener(new ActionListener() {
```

```

public void actionPerformed(ActionEvent e) {
    JFrame addFrame = new JFrame();
    addFrame.setTitle("추가될 친구 정보");
    addFrame.setSize(600,200);

    JPanel addFriend = new JPanel();

    JButton donebtn = new JButton("Done");
    addFrame.add(donebtn, BorderLayout.EAST);
    addFrame.add(addFriend, BorderLayout.CENTER);
    addFriend.setLayout(new GridLayout(2,5));
    addFriend.add(new JLabel("이름"));
    addFriend.add(new JLabel("그룹"));
    addFriend.add(new JLabel("전화번호"));
    addFriend.add(new JLabel("Email"));
    addFriend.add(new JLabel("사진"));
    JTextField[] frd = new JTextField[5];
    for (int i=0; i<5; i++) {
        frd[i] = new JTextField();
        addFriend.add(frd[i]);
    }

    donebtn.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            try {
                boolean bool1=true, bool2=true;
                String regex =
                    "^[_a-zA-Z0-9-\\.]+@[\\\\.a-zA-Z0-9-]+\\.[a-zA-Z]+$";
                for (int i=0; i<count[0]; i++) {
                    if
(frd[0].getText().equals(friendsList.friends[i].name)) {
JOptionPane.showMessageDialog(null, "Name Conflict");
bool1 = false;
                    }
                    if (!(frd[3].getText().matches(regex))) {
JOptionPane.showMessageDialog(null, "Illegal email address");
bool2 = false;
                    }
                    if (bool1 && bool2) {
                        friendsList.friends[count[0]].name =
frd[0].getText();
                        friendsList.friends[count[0]].group =
Integer.parseInt(frd[1].getText());
                        friendsList.friends[count[0]].phone =
frd[2].getText();
                        friendsList.friends[count[0]].email =
frd[3].getText();
                        friendsList.friends[count[0]].picture =
frd[4].getText();
                        name[count[0]] = new
JLabel(friendsList.friends[count[0]].name);
                        info[count[0]][0] = new
JTextField(Integer.toString(friendsList.friends[count[0]].group));
                        info[count[0]][1] = new
JTextField(friendsList.friends[count[0]].phone);
                        info[count[0]][2] = new
JTextField(friendsList.friends[count[0]].email);
                        info[count[0]][3] = new
JTextField(friendsList.friends[count[0]].picture);
                        friends.add(cbox[count[0]]);
                        friends.add(name[count[0]]);
                        friends.add(info[count[0]][0]);
                        friends.add(info[count[0]][1]);
                    }
                }
            }
        }
    });
}

```

```

        friends.add(info[count[0]][2]);
        friends.add(info[count[0]][3]);
        count[0]++;
        friends.setLayout(new GridLayout(count[0]+1,6));
        friends.revalidate();
        addFrame.setVisible(false);
    }
    catch (NumberFormatException ne) {

OptionPane.showMessageDialog(null, "Illegal value");
    }
    });

    addFrame.setVisible(true);
});

delbtn.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        for (int i=0; i<100; i++) {
            if (cbox[i].isSelected()) {
                for (int j=i; j<count[0]; j++) {
                    friendsList.friends[j].name =
friendsList.friends[j+1].name;
                    friendsList.friends[j].group =
friendsList.friends[j+1].group;
                    friendsList.friends[j].phone =
friendsList.friends[j+1].phone;
                    friendsList.friends[j].email =
friendsList.friends[j+1].email;
                    friendsList.friends[j].picture =
friendsList.friends[j+1].picture;
                }
                name[j].setText(friendsList.friends[j].name);
                info[j][0].setText(Integer.toString(friendsList.friends[j].group));
                info[j][1].setText(friendsList.friends[j].phone);
                info[j][2].setText(friendsList.friends[j].email);
                info[j][3].setText(friendsList.friends[j].picture);
            }
            friends.remove(cbox[count[0]-1]);
            friends.remove(name[count[0]-1]);
            friends.remove(info[count[0]-1][0]);
            friends.remove(info[count[0]-1][1]);

            friends.remove(info[count[0]-1][2]);
            friends.remove(info[count[0]-1][3]);
            count[0]--;
            friends.setLayout(new
GridLayout(count[0]+1,6));
            friends.revalidate();
            JOptionPane.showMessageDialog(null,
"Deletion Completed");
            break;
        }
    }
});

modbtn.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        for (int i=0; i<100; i++) {
            if (cbox[i].isSelected()) {
                try {

```

```

        String regex =
        "^[_a-zA-Z0-9-\\.]+@[\\_a-zA-Z0-9-]+\\.([a-zA-Z]+)$";
        if
        (!(info[i][2].getText().matches(regex)))
        JOptionPane.showMessageDialog(null, "Illegal email address");
        else {

        friendsList.friends[i].group = Integer.parseInt(info[i][0].getText());
        friendsList.friends[i].phone = info[i][1].getText();
        friendsList.friends[i].email = info[i][2].getText();
        friendsList.friends[i].picture = info[i][3].getText();
        JOptionPane.showMessageDialog(null, "Modification Completed");
        }
        catch(NumberFormatException ne) {

        JOptionPane.showMessageDialog(null, "Illegal value");
        }
        break;
    }
    });

    savebtn.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent ev) {
            try {
                FileOutputStream output = new FileOutputStream(fileName);
                PrintWriter pw = new PrintWriter(output);
                for (int i=0; i<count[0]; i++) {
                    String line = friendsList.friends[i].name + " : " +
                    friendsList.friends[i].group + " : " +
                    friendsList.friends[i].phone + " : " +
                    friendsList.friends[i].email + " : " +
                    friendsList.friends[i].picture;
                    pw.println(line);
                }

                pw.close();
                output.close();
                JOptionPane.showMessageDialog(null, "File Save Completed");
            }
            catch(FileNotFoundException e) {
                System.out.println("Unknown File");
            }
            catch(IOException e) {
                System.out.println("IO Error");
            }
        }
    });

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

```

II. 실행 화면

초기 GUI 화면

친구 목록

	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	
Delete						
Modify						
Save File						

Add - 친구 추가

추가될 친구 정보

이름	그룹	전화번호	Email	사진	
Harry	4	010-1010-1010	harry@naver.com	harry.jpg	Done

친구 목록

	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	
<input type="checkbox"/>	Harry	4	010-1010-1010	harry@naver.com	harry.jpg	Delete
<input type="checkbox"/>	Hugo	1	010-0101-0101	hugo@daum.net	hugo.jpeg	
<input type="checkbox"/>	Toby	2	010-4444-4444	toby@google.com	toby.jpg	
<input type="checkbox"/>	Danny	2	010-0303-0303	danny@naver.com	danny.gif	Modify
<input type="checkbox"/>	Eric	3	010-1515-1515	eric@daum.net	eric.jpg	
<input type="checkbox"/>	Dele	3	010-2020-2020	dele@naver.com	dele.jpeg	
<input type="checkbox"/>	Fernando	4	010-1818-1818	fernando@google.co	fernando.jpg	Save File
<input type="checkbox"/>	Victor	3	010-1212-1212	victor@naver.com	victor.gif	

예외처리 1 – 이미 같은 이름의 친구가 존재할 때

추가될 친구 정보

이름	그룹	전화번호	Email	사진
John	2	010-4444-4444	toby@google.com	toby.jpg

Done

메시지

Name Conflict

확인

예외처리 2 – 그룹에 숫자가 아닌 값이 들어갔을 때

추가될 친구 정보

이름	그룹	전화번호	Email	사진
Toby	A	010-4444-4444	toby@google.com	toby.jpg

Done

메시지

Illegal value

확인

예외처리 3 – 이메일 형식이 맞지 않을 때

추가될 친구 정보

이름	그룹	전화번호	Email	사진
Toby	2	010-4444-4444	@tobycom	toby.jpg

Done

메시지

Illegal email address

확인

Delete - 친구 Tom 제거

친구 목록

	이름	그룹	전화번호	Email	사진
<input type="checkbox"/>	John	1	010-1111-1111	john@google.com	john.gif
<input checked="" type="checkbox"/>	Peter	2	010-3333-3333	peter@daum.net	peter.jpg
<input type="checkbox"/>	Harry	4	010-1010-1010	harry@naver.com	harry.jpg
<input type="checkbox"/>	Hugo	1	010-0101-0101	hugo@daum.net	hugo.jpeg
<input type="checkbox"/>	Toby	2	010-4444-4444	toby@google.com	toby.jpg
<input type="checkbox"/>	Danny	2	010-0303-0303	danny@naver.com	danny.gif
<input type="checkbox"/>	Eric	3	010-1515-1515	eric@daum.net	eric.jpg
<input type="checkbox"/>	Dele	3	010-2020-2020	dele@naver.com	dele.jpeg
<input type="checkbox"/>	Fernando	4	010-1818-1818	nando@google.com	fernando.jpg
<input type="checkbox"/>	Victor	3	010-1212-1212	victor@naver.com	victor.gif

메시지

Deletion Completed

확인

Add

Delete

Modify

Save File

Modify - 친구 Peter 정보 수정

친구 목록

	이름	그룹	전화번호	Email	사진
<input type="checkbox"/>	John	1	010-1111-1111	john@google.com	john.gif
<input checked="" type="checkbox"/>	Peter	4	010-9999-9999	peter@naver.com	peter.jpeg
<input type="checkbox"/>	Harry	4	010-1010-1010	harry@naver.com	harry.jpg
<input type="checkbox"/>	Hugo	1	010-0101-0101	hugo@daum.net	hugo.jpeg
<input type="checkbox"/>	Toby	2	010-4444-4444	toby@google.com	toby.jpg
<input type="checkbox"/>	Danny	2	010-0303-0303	danny@naver.com	danny.gif
<input type="checkbox"/>	Eric	3	010-1515-1515	eric@daum.net	eric.jpg
<input type="checkbox"/>	Dele	3	010-2020-2020	dele@naver.com	dele.jpeg
<input type="checkbox"/>	Fernando	4	010-1818-1818	nando@google.com	fernando.jpg
<input type="checkbox"/>	Victor	3	010-1212-1212	victor@naver.com	victor.gif

메시지

Modification Completed

확인

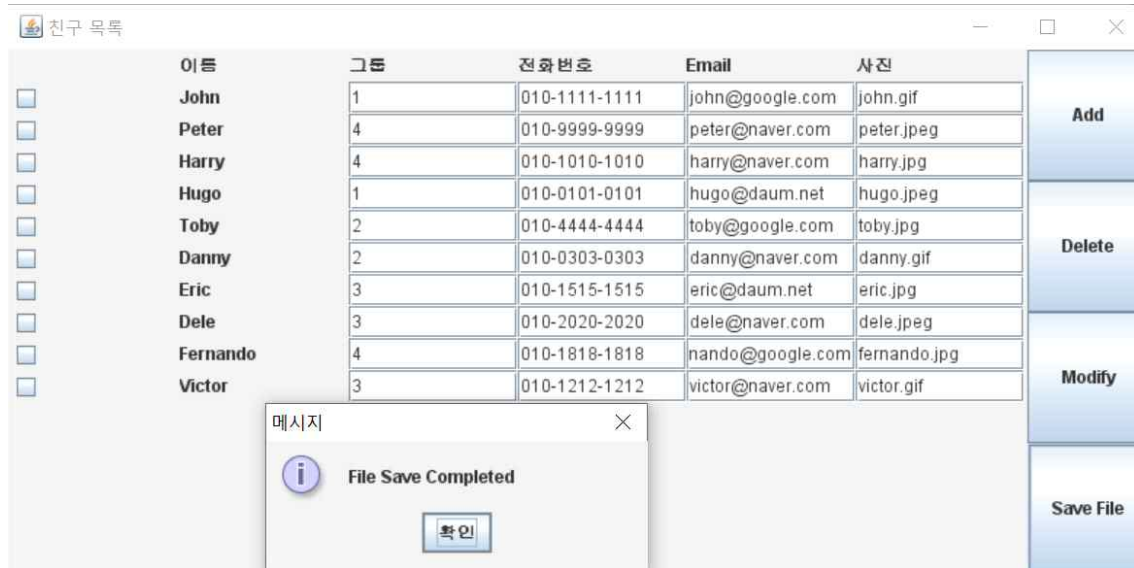
Add

Delete

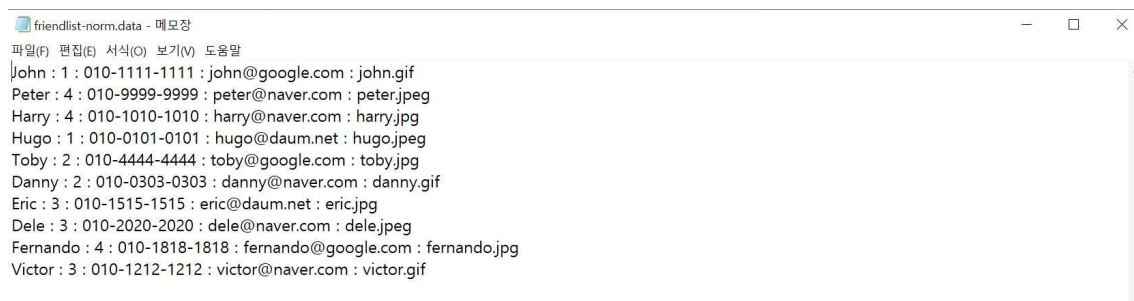
Modify

Save File

Save File – 파일에 저장



friendlist-norm.data에 저장된 결과



III. 설계 노트

<Friend, FriendList, FriendListFile class>

이 클래스들은 project3에서 수행했던 것이다.

<FriendListGUI class>

GUI 프로그램의 기본적인 프레임과 레이아웃 등은 project5에서 수행했던 것이므로 그 이후의 과정만 기술한다.

① 현재 저장된 친구의 숫자를 나타내는 변수 설정

원래 기존 project5까지의 프로그램에서는 저장된 친구의 수를 나타낼 때 *i* 라는 int형 변수를 사용하였다. 하지만 이 변수를 다른 클래스에서도 사용해야 하는데 변수를 공유하려 하니 "local variable defined in an enclosing scope must be final or effectively final"라는 예러 메시지가 출력되었다. 이 값은 계속해서 바뀌는 값이기 때문에 static final로 선언하는 것도 불가능했다. 그래서 이를 해결하기 위한 방법으로 count라는 int형 배열을 선언하여 count[0]에 값을 저장하고 이를 다른 클래스에서도 사용할 수 있도록 했다.

② CheckBox, 친구 정보 TextField의 배열 선언

친구 정보를 삭제하거나 수정할 때 체크박스의 체크여부를 확인하여 체크된 친구의 정보를 다루어야 하는데, 이 과정을 편리하게 하기 위해 크기가 100인 체크박스 배열을 선언하였고 친구 정보는 100*4인 2차원 배열을 선언하였다. 이렇게 하면 *i*번째 체크박스가 체크되었을 때 친구 정보는 info[i][0], info[i][1], info[i][2], info[i][3]을 가지고 삭제 혹은 수정을 하면 된다.

③ 친구 정보를 화면에 표시

친구 정보를 파일에서 읽어와 화면에 표시할 때에는 while 문을 이용하여 친구 정보가 null이 아닐 때까지 friendsList.friends[count[0]]의 정보를 info[count[0]]에 담는다. 담은 정보를 friends 패널에 add해주고 count[0]를 1 올려준다. 이 때 friends 패널의 레이아웃은 GridLayout을 이용하여 행 수는 친구 각각의 정보와 맨 윗줄을 표현해야 하기 때문에 count[0]+1로 하고, 열 수는 체크박스, 이름, 그룹, 전화번호, 이메일, 사진을 포함해야 하기 때문에 6으로 한다.

④ Add 버튼

ActionListener 클래스를 활용하여 Add 버튼이 클릭되었을 때 실행할 이벤트를 만든다. JFrame 객체 addFrame을 선언하여 Add 버튼 클릭 시 프레임이 뜨도록 한다. 프레임에는 친구 정보를 입력할 수 있도록 TextField를 넣는다. TextField에 정보를 입력하고 Done 버튼을 클릭하면 입력된 정보가 friendsList.friends[count[0]]에 저장되도록 한다. 이 때 getText() 메소드를 이용한다. 그리고 이를 화면에도 적용시켜줘야 하기 때문에 info[count[0]]에 정보를 담아 add해준다. 그리고 count[0]을 1 증가시키고 레이아웃 설정을 다시 한 번 해준 후에 revalidate()를 하여 화면을 갱신시킨다.

⑤ Delete 버튼

ActionListener 클래스를 활용하여 Delete 버튼이 클릭되었을 때 실행할 이벤트를 만든다. isSelected()를 이용하여 체크박스의 체크여부를 확인한 후에 체크되었으면 체크된 줄의 친구 정보를 제거하는 것이다. 여기서 문제가 발생했는데, 단순히 체크된 친구의 정보를 null로 초기화시키고 구성요소를 remove하면 친구 목록 배열의 중간에 빈자리가 생기게 된다. 예를 들어 친구가 5명 저장되어 있을 때 3번째 친구가 제거되면 count[0]이 4가 된다. 이후에 친구를 추가할 때 count[0]이 4이기 때문에 5번째 배열에 친구를 추가하려 하지만 이미 5번째 배열에는 친구가 저장되어 있는 상태이기 때문에 문제가 발생하는 것이다. 이를 해결하기 위해 체크된 친구의 정보 뒤쪽부터 한 칸 씩 정보를 앞으로 옮기도록 했다. 이렇게 하면 빈자리가 생기지 않아 문제를 해결할 수 있다. 하지만 이 방법의 문제점은 친구를 한 번에 한 명씩만 삭제할 수 있다는 것이다. 여러 친구를 삭제하려 하면 정보가 앞당겨지면서 체크된 것이 꼬여버려 문제가 발생하는 것이다. 따라서 어쩔 수 없이 첫 번째 체크된 체크박스의 친구 정보를 제거하고 나면 break를 사용하여 for문을 탈출하도록 했다. 친구 정보 제거를 완료하면 메시지 창이 뜨도록 하였다.

⑥ Modify 버튼

ActionListener 클래스를 활용하여 Modify 버튼이 클릭되었을 때 실행할 이벤트를 만든다. isSelected()를 이용하여 체크박스의 체크여부를 확인한 후에 체크되었으면 체크된 줄의 친구 정보를 화면의 내용으로 수정하는 것이다. getText() 메소드를 이용하여 friendsList.friends[i]의 정보를 수정하고 완료 시에 역시 메시지 창이 뜨도록 하였다. 이때도 역시 break를 사용하여 한 명의

친구 정보만을 수정하도록 했다.

⑦ Save File 버튼

ActionListener 클래스를 활용하여 Save File 버튼이 클릭되었을 때 실행할 이벤트를 만든다. FileOutputStream과 PrintWrier를 이용하여 파일에 문자열이 저장되도록 한다. for문을 활용하여 현재 저장된 친구 수(count[0])만큼 지정된 양식(ex) Fred : 1 : 010-1111-1111 : fred@google.com : fred.gif)으로 친구 정보를 파일에 넣는다. 파일 저장을 완료하면 close()를 해주었다. 이번에도 파일 저장 완료 시에 메시지 창이 뜨도록 했다.

⑧ 예외 처리

발생할 수 있는 오류에 대해 예외 처리를 해주었다. try~catch문을 활용하였다. 친구를 추가하거나 친구 정보를 수정하는 경우에 대해 여러 예외 처리를 했다. 먼저 equals() 메소드를 이용하여 새로 입력하는 친구의 이름이 이미 저장된 친구 중에 있을 경우 "Name Conflict"라는 메시지 창이 뜨도록 했다. 그리고 matches() 메소드를 이용하여 이메일 형식이 틀렸을 경우에 "Illegal email address"라는 메시지 창이 뜨도록 했다. 그리고 그룹은 숫자 형식으로 되어있어야 하는데 그게 아닐 경우에 "Illegal value"라는 메시지 창이 뜨도록 했다. 이 때 NumberFormatException을 이용하여 예외 처리를 해주었다. 이 경우들이 아닐 때에만 친구 추가 혹은 수정이 이루어지도록 했다. 파일을 저장하는 경우에는 FileNotFoundException과 IOException에 대해 예외 처리를 해주었다.

IV. 자체평가표

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"> - "초기" 동작 - "Add" 동작 - "Modify" 동작 - "Delete" 동작 - "Save" 동작 - 기타 비정상 동작 실험 (구현 한계 점검) 	<p>GUI 화면은 깔끔하게 잘 구성한 것 같다. 각 버튼의 기능도 ActionListener를 활용하여 알맞게 구현해냈다. 예외 처리도 잘 해주었지만 해주지 못한 경우들이 더 존재할 것 같다.</p>		
<p>설계 노트</p> <ul style="list-style-type: none"> - 주요 결정사항 및 근거 - 한계/문제점 - 해결 방안 - 필수내용: * 프로그램 구성(Class 구조) * member visibility 	<p>프로그램을 짜면서 발생한 여러 시행착오들을 그래도 나름 잘 해결한 것 같다. 클래스 간에 정보를 공유하는 부분을 제대로 하지 못해서 하나의 큰 클래스 안에 구성하게 된 점은 약간 아쉽다. 또한 Delete를 할 때 한 명씩만 삭제를 할 수 있도록 제한을 두게 된 것도 해결해야 할 부분이다.</p>		
<p>리포트</p> <ul style="list-style-type: none"> - 평가자 시각으로 리포트 검토 - 위의 평가 요소들이 명확하게 기술되었는가? 	<p>여러 경우의 실행화면을 통해 프로그램이 잘 작동한다는 것을 보여주고 있다. 프로그램을 설계할 때 각 부분에서 어떤 방식으로 설계를 했는지를 설계 노트에 자세하고 명확하게 기술했다.</p>		
총평/계	<p>약간의 아쉬운 부분이 있고 해결해야 할 부분도 있지만 일단 전체적으로 기능을 거의 완벽히 구현한 것 같아 만족스럽다.</p>		