



소프트웨어프로젝트 02분반

project5 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++;
    }
}
```

```

        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();
                    }
                    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");
                    }
                    if (!(token[3].matches(regex))) {
                        System.out.println("Illegal email
address");
                    }
                    else {
                        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.*;

public class FriendListGUI extends JFrame {
    public FriendListGUI(String fileName) {
        int i=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"));
        while ( friendsList.friends[i].name != null ) {
            friends.add(new JCheckBox());
            friends.add(new JLabel(friendsList.friends[i].name));
            friends.add(new
JTextField(Integer.toString(friendsList.friends[i].group)));
            friends.add(new JTextField(friendsList.friends[i].phone));
            friends.add(new JTextField(friendsList.friends[i].email));
            friends.add(new JTextField(friendsList.friends[i].picture));
            i++;
        }
        friends.setLayout(new GridLayout(i+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) {
                System.out.println("Add button clicked.");
            }
        });
        delbtn.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                System.out.println("Delete button clicked.");
            }
        });
    }
}
```

```

    });
    modbtn.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            System.out.println("Modify button clicked.");
        }
    });
    savebtn.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            System.out.println("Save File button clicked.");
        }
    });

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

```

<AddFriend.java>

```

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

public class AddFriend extends JFrame {
    public AddFriend() {

        setTitle("추가될 친구 정보");
        setSize(600,200);

        JPanel friends = new JPanel();

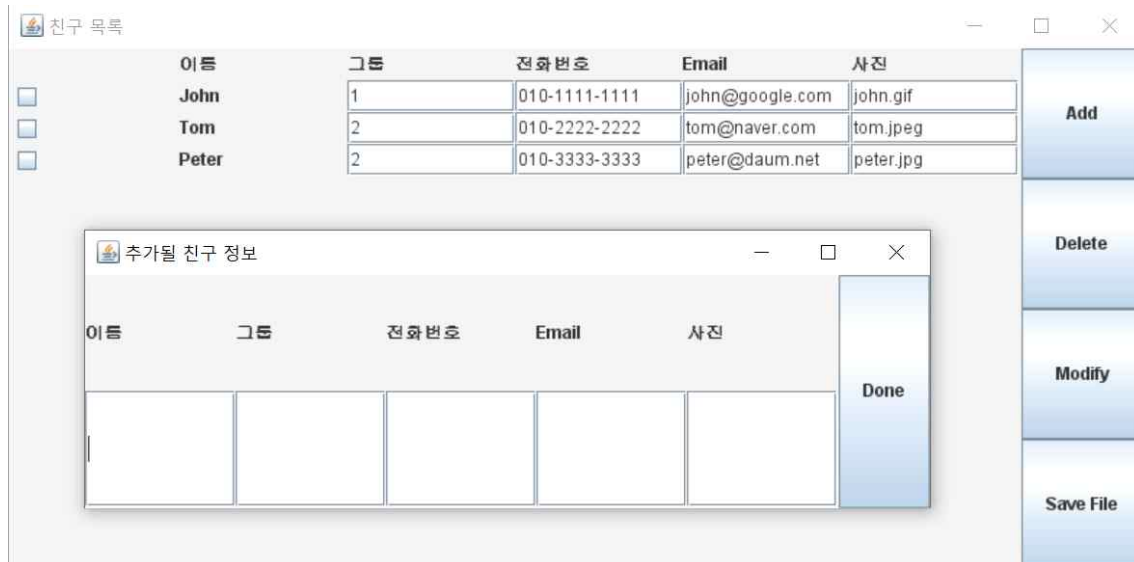
        JButton donebtn = new JButton("Done");
        add(donebtn, BorderLayout.EAST);
        donebtn.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                System.out.println("Done button clicked.");
            }
        });
        add(friends, BorderLayout.CENTER);
        friends.setLayout(new GridLayout(2,5));
        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"));
        for (int i=0; i<5; i++)
            friends.add(new JTextField());

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

```

II. 실행 화면

GUI 화면



ActionListener 테스트 화면



III. 설계 노트

<Friend, FriendList, FriendListFile class>

이 클래스들은 project3에서 수행했던 것이다.

<친구 목록 화면 GUI화면을 생성하는 FriendListGUI class>

project3에서 만들었던 friendListGenerator() 메소드를 이용하여 FriendList 객체를 불러오고 FriendListFile 클래스의 readFileToList 메소드로 파일을 읽어올 수 있도록 한다. setTitle를 이용하여 제목을 "친구 목록"으로 하고, setSize를 이용하여 크기를 800*400으로 한다. JPanel로 세 패널 panel, buttons, friends를 생성한다. BorderLayout으로 panel을 CENTER에 add하고 panel 안에서 BorderLayout을 다시 사용하여 panel 내에서 NORTH에 friends를 add한다. friends 패널에 JLabel로 여러 항목들을 추가한다. 그 밑줄에는 체크박스과 이름 라벨, 다른 정보들의 텍스트필드를 추가한다. 이 때, 친구 정보가 저장되어있는 만큼만 add해줘야 하기 때문에 while 문을 이용하여 i를 증가시키면서 friendsList.friends[i]가 비어있지 않을 때까지만 추가하도록 한다. friends 내에서의 레이아웃은 GridLayout을 사용하는데 이 때 열의 수는 6으로 하면 되지만 행의 수는 저장되어있는 친구 수에 따라 달라져야 하기 때문에 i+1로 해준다. buttons 패널을 BorderLayout으로 EAST에 add한다. 버튼 네 개를 세로로 배열할 것이기 때문에 GridLayout(4,1)로 배치한다. JButton으로 Add 버튼, Delete 버튼, Modify 버튼, Save File 버튼을 생성한다. 각각의 버튼이 클릭되면 클릭되었다는 사실이 출력되도록 ActionListener를 이용하여 버튼이 클릭됐다는 문구가 출력되도록 한다. 마지막으로 setDefaultCloseOperation으로 프로세스가 안정적으로 종료될 수 있도록 하고, setVisible(true)로 창을 화면에 나타내도록 한다.

<친구 추가 GUI 화면을 생성하는 AddFriend class>

setTitle를 이용하여 제목을 "추가될 친구 정보"로 하고, setSize를 이용하여 크기를 600*200으로 한다. JPanel로 패널 friends를 생성한다. JButton으로 Done 버튼을 생성하고 BorderLayout으로 EAST에 add한다. ActionListener를 이용하여 버튼 클릭 시 버튼이 클릭됐다는 문구가 출력되도록 한다. BorderLayout으로 friends 패널을 CENTER에 add하고 panel 안에 JLabel로 여러 항목들을 추가한다. 그 밑줄에는 텍스트필드 다섯 개를 추가한다. 이를 GridLayout(2,5)로 배치한다. 마지막으로 setDefaultCloseOperation으로 프로세스가 안정적으로 종료될 수 있도록 하고, setVisible(true)로 창을 화면에 나타내도록 한다.

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"> - GUI 화면 완성도 - 클릭 시 동작 	<p>GUI 화면을 과제 설명 예시에 있는 것과 똑같이 잘 구현해냈다. 또한 버튼 클릭 시에 문구가 출력되도록 ActionListener를 잘 사용하였다. 다만 이에 대한 이해가 아직은 부족한 것 같다.</p>		
<p>설계 노트</p> <ul style="list-style-type: none"> - 클래스 구조 - GUI 설계 시 착안 사항 - 시행 착오 내용 및 해결 방안 	<p>FriendListGUI와 AddFriend 클래스를 각각 만들었다. GUI를 설계할 때 각 요소별로 알맞은 위치에 배치하는 것을 중요하게 생각했다. 레이아웃을 설정할 때 어려움이 있었는데 패널 속에 패널을 배치함으로써 문제를 해결하였다.</p>		
<p>리포트</p> <ul style="list-style-type: none"> - 평가자 시각으로 리포트 검토 - 위의 평가 요소들이 명확하게 기술되었는가? - 기타 건의 사항 	<p>프로그램이 잘 동작된다는 것을 실행 화면을 통해 알 수 있고 설계 노트에 소스 코드에 대한 설명이 있어 코드를 이해하는 데에 도움이 될 수 있다.</p>		
총평/계	<p>제대로 된 방식으로 했는지는 모르겠으나 GUI를 완벽하게 구현해냈다는 것만큼은 만족스럽다.</p>		