



소프트웨어프로젝트 02분반

# project3 report

소프트웨어학부

20181269 박현우

## I. 소스 프로그램

### <Friend.java>

```
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>

```
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>

```
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.friendsListGenerator();

        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(":");
                }
            }
        }
    }
}
```

```

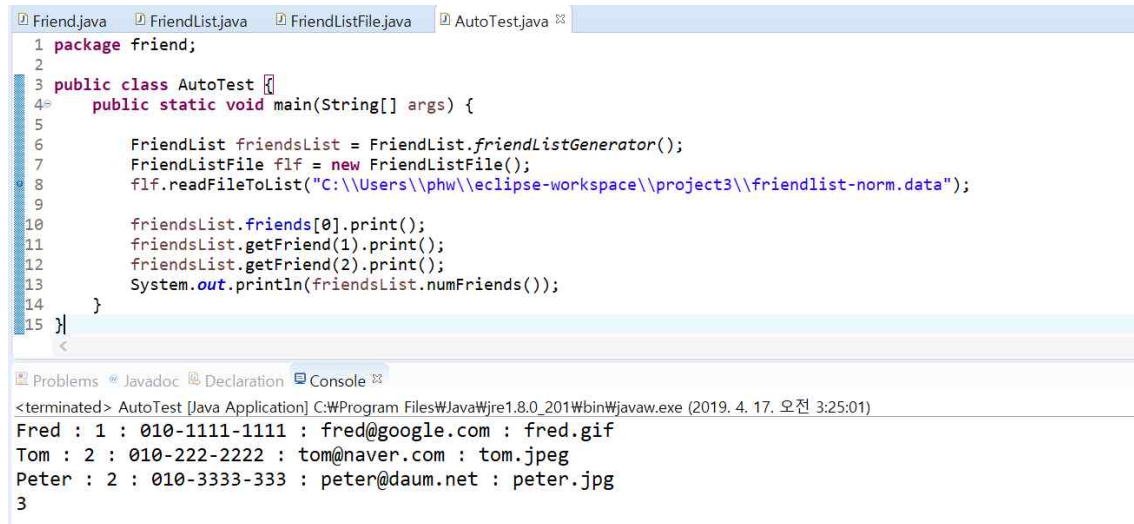
        for(int i=0; i<token.length; i++) {
            token[i] = token[i].trim();
        }
        String regex =
"^[_a-zA-Z0-9-\\.]+@[\\.a-zA-Z0-9-]+\\.([a-zA-Z]+)$";
        for (int i=0; i<idx; i++) {
            if
(token[0].equals(friendsList.friends[i].name))
                System.out.println("Name
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;
}
}

```

## II. 실행 화면

eclipse에서의 Test 화면



```
1 package friend;
2
3 public class AutoTest {
4     public static void main(String[] args) {
5
6         FriendList friendsList = FriendList.friendListGenerator();
7         FriendListFile flf = new FriendListFile();
8         flf.readFileToList("C:\\Users\\phw\\eclipse-workspace\\project3\\friendlist-norm.data");
9
10        friendsList.friends[0].print();
11        friendsList.getFriend(1).print();
12        friendsList.getFriend(2).print();
13        System.out.println(friendsList.numFriends());
14    }
15 }
```

<terminated> AutoTest [Java Application] C:\Program Files\Java\jre1.8.0\_201\bin\javaw.exe (2019. 4. 17. 오전 3:25:01)

Fred : 1 : 010-1111-1111 : fred@google.com : fred.gif  
Tom : 2 : 010-222-2222 : tom@naver.com : tom.jpeg  
Peter : 2 : 010-3333-333 : peter@daum.net : peter.jpg  
3

165.194.35.156에서의 AutoTest 화면

```
Wed Apr 17 03:26:14 KST 2019 by 20181269
Case1: Normal Input
Fred : 1 : 010-1111-1111 : fred@google.com : fred.gif
Tom : 2 : 010-222-2222 : tom@naver.com : tom.jpeg
Peter : 2 : 010-3333-333 : peter@daum.net : peter.jpg
End of Case1

Case2: Check String Trimming
The result should be the same as the normal case above.
Fred : 1 : 010-1111-1111 : fred@google.com : fred.gif
Tom : 2 : 010-222-2222 : tom@naver.com : tom.jpeg
Peter : 2 : 010-3333-333 : peter@daum.net : peter.jpg

End of Case2

*** From now on, Test Reactions on Abnormal Inputs ***
Your Answer (your program's reaction) should be similar to the Correct Answer

Case3-0: Check ID Conflicts
Correct Answer: Name Conflict
Your Answer : Name Conflict
Fred : 1 : 010-1111-1111 : fred@google.com : fred.gif
Tom : 2 : 010-222-2222 : tom@naver.com : tom.jpeg
Tom : 2 : 010-3333-333 : peter@daum.net : peter.jpg
End of Case3-0

Case3-1: Check Format Error ->
Correct Answer: Irregular input line ; Skip the input line : Tom:2: 010-222-2222: tom@naver.com
Your Answer : Irregular input line
Fred : 1 : 010-1111-1111 : fred@google.com : fred.gif
Tom : 2 : 010-222-2222 : tom@naver.com : tom.jpeg
Tom : 2 : 010-3333-333 : peter@daum.net : peter.jpg
End of Case3-1

Case3-2: Check Format Error ->
Correct Answer: Illegal value -- xxx ; Skip the input line : Fred: xxx:010-1111-1111: fred@google.com :fred.gif
Your Answer : Illegal value
Fred : 1 : 010-1111-1111 : fred@google.com : fred.gif
Tom : 2 : 010-222-2222 : tom@naver.com : tom.jpeg
Tom : 2 : 010-3333-333 : peter@daum.net : peter.jpg
End of Case3-2

Case3-3: Check Format Error ->
Correct Answer: Illegal email address ; Skip the input line : Peter: 2 : 010-3333-333 : peter#daum.net : peter.jpg
Your Answer : Illegal email address
Fred : 1 : 010-1111-1111 : fred@google.com : fred.gif
Tom : 2 : 010-222-2222 : tom@naver.com : tom.jpeg
Tom : 2 : 010-3333-333 : peter@daum.net : peter.jpg
End of Case3-3

Case4: Unknown Schedule File
Correct Answer: Unknown File
Your Answer : Unknown File
Fred : 1 : 010-1111-1111 : fred@google.com : fred.gif
Tom : 2 : 010-222-2222 : tom@naver.com : tom.jpeg
Tom : 2 : 010-3333-333 : peter@daum.net : peter.jpg
End of Case4
```

### III. 설계 노트

#### <하나의 친구를 나타내는 **Friend class**>

Friend class의 속성으로 이름을 나타내는 문자열 name, 그룹 번호를 나타내는 int 형 변수 group, 전화번호를 나타내는 문자열 phone, 이메일 주소를 나타내는 문자열 email, 프로필 사진을 나타내는 문자열 picture를 추가한다. Friend 객체를 만들어내기 위한 생성자를 선언한다. 생성자에서 각 항목에 값을 저장할 때 함수 getName(), getGroup(), getPhone(), getEmail(), getPicture()를 이용하여 저장하도록 한다. 하나의 친구에 대한 정보를 출력하는 메소드인 print()를 선언한다. 이 메소드는 System.out.println을 이용하여 친구 정보를 출력하도록 한다. 이 때 출력 형태는 " Fred : 1 : 010-1111-1111 : fred@google.com : fred.gif "와 같이 한다.

#### <친구들의 묶음을 나타내는 **FriendList class**>

배열의 이름이 friends이고 배열의 길이가 100인 Friend 객체 배열을 선언한다. FriendList 생성자를 만들 때 생성자 앞에 private 접근 제한자를 붙여준다. 생성자를 외부에서 호출할 수 없도록 하여 단 하나의 객체만을 만들도록 하기 위해서이다. 클래스 내부에서 FriendList 객체 friendsList를 생성한다. 그리고 외부에서 이를 호출하여 사용할 수 있도록 friendListGenerator() 메소드를 선언하여 FriendList 객체를 리턴해주도록 한다. 목록에 있는 친구 숫자를 알려주는 메소드인 numFriends()를 선언한다. 이 메소드는 friends 배열의 객체에 저장된 값이 존재하면 count값을 증가시켜 친구 숫자를 계산한다. 목록에 있는 i번째 친구 정보를 알려주는 메소드인 getFriend(int i)를 선언한다. 이 메소드는 friends 배열 중에 i를 인덱스로 하는 Friend 객체를 리턴한다.

#### <파일의 내용을 읽어 친구 정보와 친구 목록을 만드는 **FriendListFile class**>

import를 이용하여 여러 클래스들을 불러온다. 주어진 파일에서 친구 목록을 생성하는 메소드인 readFileToList(String fileName)을 선언한다. 이 메소드 내에서 friendListGenerator()를 이용하여 FriendList 생성자를 호출한다. try~catch 문을 이용하여 파일을 받아온다. try문에서 File로 파일을 받아오고 Scanner로 파일을 읽는다. hasNext()를 이용하여 다음 요소가 있는지 판별한 후 이를 while 문에 집어넣는다. nextLine()을 이용하여 한 줄씩 읽어온 후 line에 저장한다. 처음을 //로 시작하면 주석 처리를 해야 하기 때문에 startsWith로 이를 판별한 후 //로 시작하지 않는 경우에만 구분자로 쪼개어 저장하는 작업을 한다. split()을 이용하여 :를 구분자로 하여 line을 쪼갰 후 token 배열에 담는다. token 배열에 담긴 문자열을 trim()을 이용하여 앞뒤 공백을 제거해준다. 이메일 형식을 regex로 정의하고 token[3]에 저장된 문자열이 이와 맞지 않는 경우에 "Illegal email address"라는 메시지를 출력한다.

token에 새로 저장하려는 문자열이 기존에 저장해놓은 문자열과 일치할 경우 "Name Conflict"라는 메시지를 출력한다. 이 경우가 아니면 token 배열에 저장된 값을 friendList.friends 배열에 담는다. 이 때, 그룹 번호는 int형으로 담아야 하므로 String을 int형으로 변환해주는 Integer.parseInt를 사용한다. 이 작업을 while문의 조건을 만족하는 동안 계속 반복하고 close를 해준다. catch문은 파일을 찾을 수 없는 경우에 FileNotFoundException, 범위 밖의 배열 첨자를 지정하는 경우에 ArrayIndexOutOfBoundsException, 부적절한 문자열을 수치로 변환하고자 하는 경우에 NumberFormatException을 이용하여 작성한다. 더불어 각각 에러 메시지를 출력한다. 마지막으로 friendsList를 리턴한다.

#### <AutoTest 결과>

AutoTest를 돌려본 결과 Normal Input에 대한 반응인 Case1과 Case2는 결과가 제대로 출력되었다. 하지만 Abnormal Inputs에 대한 반응인 Case3, Case4는 에러 메시지는 제대로 출력되었으나 그 이후에 Friend 정보가 제대로 출력되지 않았다. 이것은 향후에 개선해야 할 부분이다.

#### IV. 자체평가표

평가 항목	학생 자체 평가 (리포트 해당 부분 표시 및 간단한 의견)	평가 (빈칸)	점수 (빈칸)
기본 동작 - autotest 결과화면	Normal Input에 대한 결과값은 올바르게 출력했지만, Abnormal Input에 대한 결과값은 올바르게 출력하지 못한 점이 아쉽다.		
설계 사항 - 설계 착안점 - 사용한 클래스 - 시행 착오 및 해결책	설계를 할 때 Java 스타일대로 코드를 짜기 위해 노력했다. Friend와 FriendList, FriendListFile 클래스를 기능에 맞게 사용했다. AutoTest 결과가 완벽하지 않은 점은 향후에 원인을 알아내어 해결해야 할 과제로 남았다.		
본인 인증	수업시간에 배운 개념들에 Java 책에서 보고 깨우친 것들을 더하여 직접 코드를 작성하였다.		
기타			
총평/계	클래스와 메소드를 적절히 사용하여 기본적인 구현은 해냈지만, 예외처리 부분에서 미흡한 부분이 있다.		