

(고급프로그래밍 과제3)

10/02/2021 (MM/DD/YYYY)

32212488 안태현 (고급프로그래밍 1분반)

Sections

1. Description	3
2. Solution Modification	4
3. Class Structure	5
4. Exception Handling	7
5. Data File	8
6. Program Execution	17
Execution Sequence 1	17
Execution Sequence 2	18

Section 1. Description

다음과 같은 형식을 따르는 파일이 있다.

```
3,3
Algebra,Calculus,Geometry
Student1,76,25,14
Student2,12,95,72
Student3,64,64,18
```

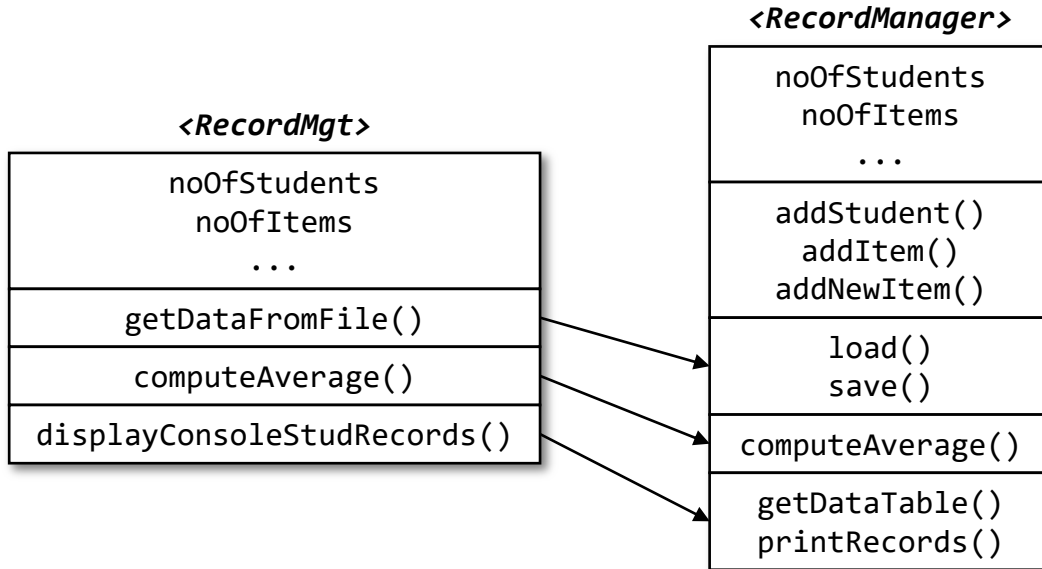
성적 정보를 담는 파일은 텍스트 파일로 첫 번째 라인은 평가 항목의 수와 학생 수, 두 번째 라인은 평가 항목 명칭의 리스트를 나타낸다. 세 번째 라인부터는 학생 이름과 각 항목들에 대한 점수들로 구성된다. 각각의 값들은 콤마(,)로 구분된다.

이때, 주어진 파일에 대하여 다음의 내용들을 구현하시오.
(단, 파일은 위에서 언급된 형식을 완벽하게 따른다고 가정하고, 코드 작성은 제시된 *solution(RecordMgt.java)*을 기반으로 작성할 것.)

1. 위에서 언급한 형식의 데이터를 읽고 저장한다.
2. 현재의 데이터를 사용자들이 읽기 쉽게 화면에 출력하는 기능을 작성한다. 마지막 행에서는 각 항목의 평균 값을 출력한다.
3. 학생 정보를 추가하는 기능을 작성한다. 이때 학생 이름과 관련 과목의 성적을 모두 입력한다.
4. 평가 항목을 추가하는 기능을 작성한다. 이때 모든 학생에 대해 추가된 항목의 성적을 입력한다.
5. 현재의 정보를 파일에 저장하는 기능을 구현한다.
6. 위의 기능들을 메뉴 기반으로 구현한다.

Section 2. Solution Modification

주어진 코드(RecordMgt)의 메소드를 몇 개 변경하였다.



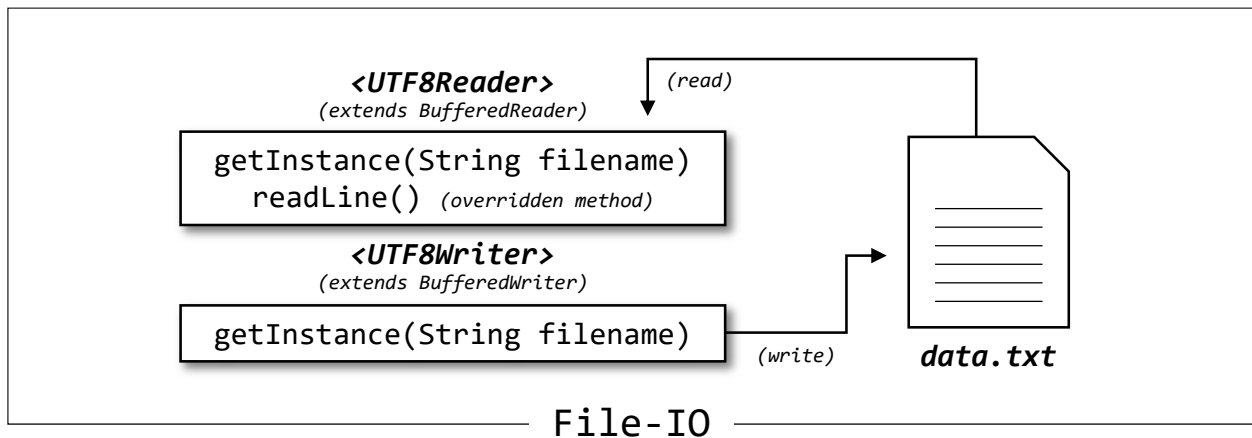
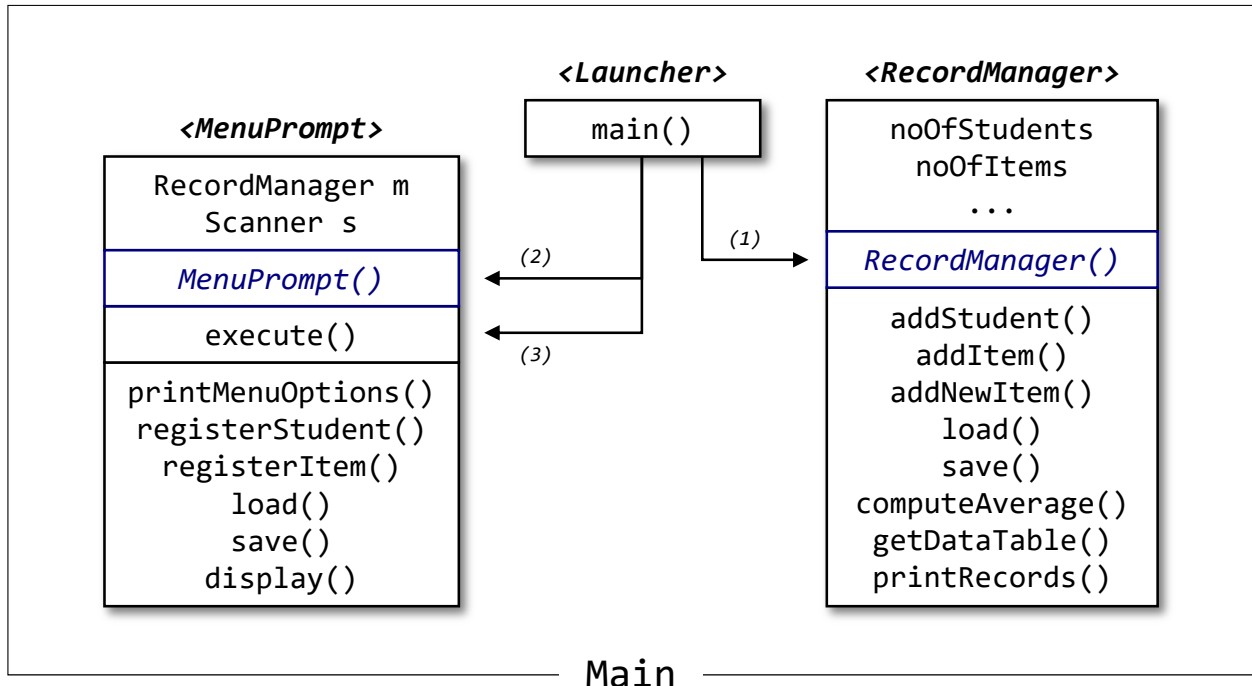
getDataFromFile()을 load()와 save()로 변경하고, displayConsoleStudRecords()는 좀 더 일반화된 테이블 형식으로 출력하기 위해서 getDataTable()과 printRecords()로 변경하였다.

또한, 코드의 중복을 방지하고 메소드의 재사용성을 높이기 위하여 addStudent(), addItem(), addNewItem() 메소드를 추가하였다.

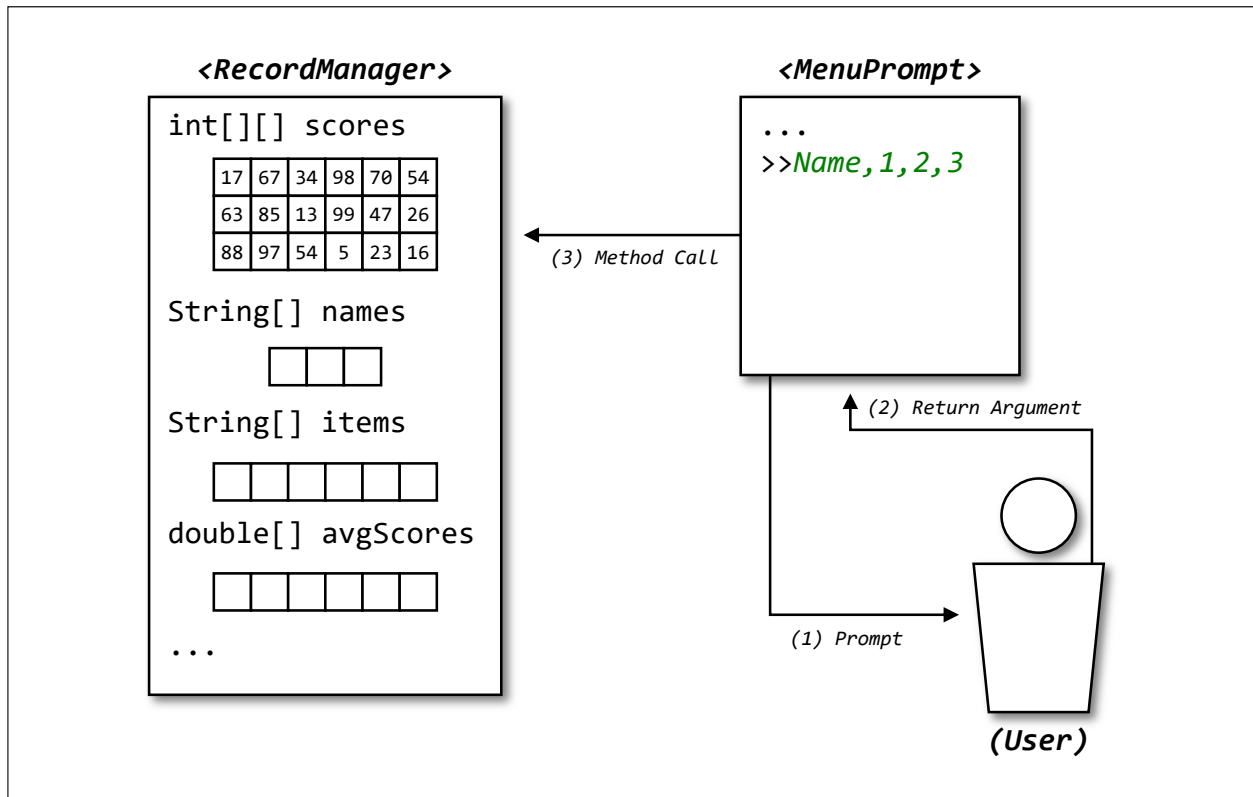
Section 3. Class Structure

앞서 언급한 RecordManager의 메소드들을 사용자가 사용할 수 있도록 프로그램을 메뉴형 인터페이스로 구현해야 한다. 따라서 MenuPrompt 클래스를 도입하였다.

클래스 간의 대략적인 구조는 아래와 같다.



MenuPrompt는 사용자와 RecordManager의 매개체 역할을 한다. 각각의 관계를 그림으로 나타내면 아래와 같다.



(1) MenuPrompt에서 사용자에게 어떤 값을 입력할 것을 요구한다.

(2) 사용자는 그에 따른 값을 입력한다.

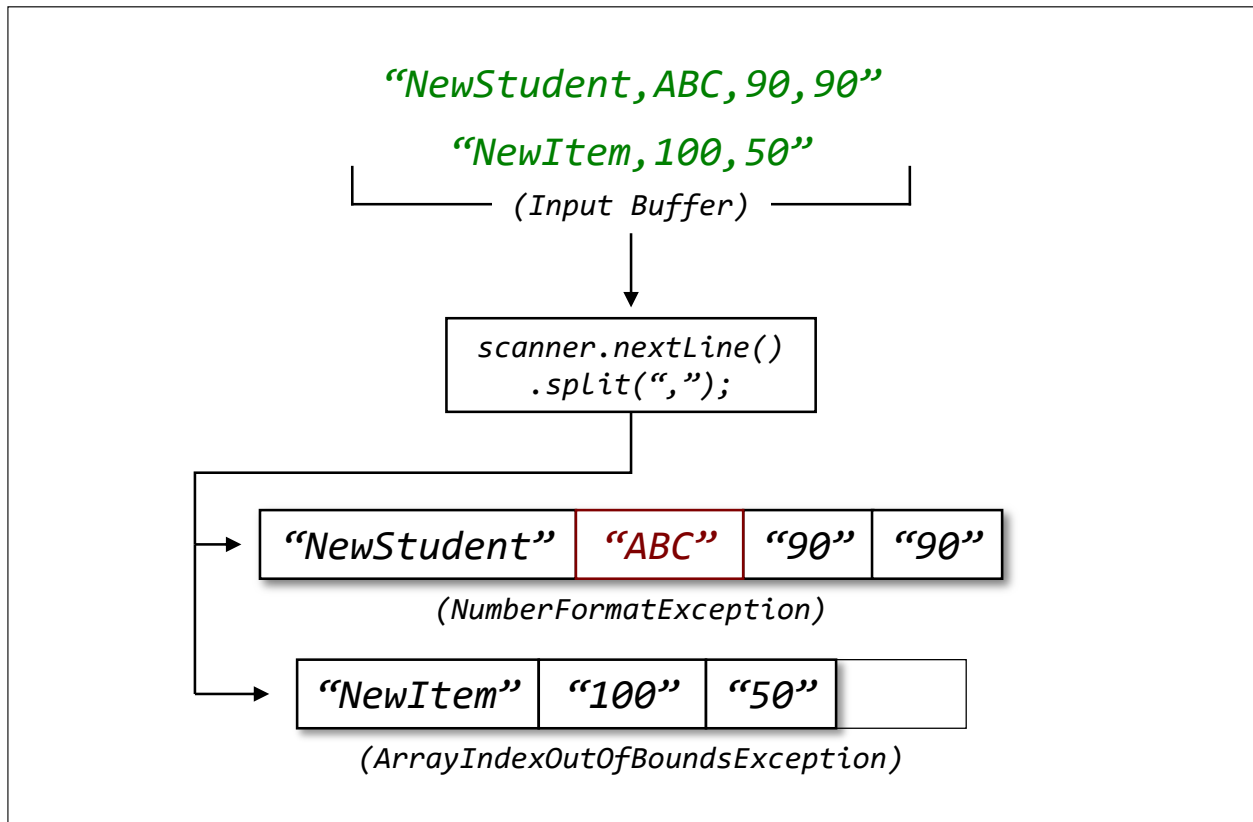
(3) MenuPrompt는 사용자의 입력값에 따라서 RecordManager의 특정한 메소드를 호출한다.

(4) RecordManager의 필드의 데이터 조작이 이루어진다.

Section 4. Exception Handling

파일은 주어진 형식을 완벽하게 따른다고 가정하였다. 그러나 사용자가 학생 또는 항목을 추가할 경우, 주어진 형식에 어긋나는 형태의 값을 전달할 가능성이 있다. 따라서 이에 대한 예외 처리가 필요하다.

예외 발생의 경우는 2가지가 있다. 아래 그림을 참고하자.



이와 같은 경우에 대응하기 위하여 RecordManager의 ‘add’ 계열 메소드들은 다음과 같이 예외를 처리한다.

```
catch (Exception e) {
    if (index+1 < array.length)
        System.err.printf("Invalid argument: unexpected '%s' at position %d\n",
array[index+1], index+1);
    else
        System.err.printf("Invalid argument: reached end of line at position %d\n",
index+1);
    return false;
}
```

Section 5. Data File

data.txt [file] (UTF-8 encoded)

```
3,4
Algebra,Calculus,Geometry
Student1,72,12,78
Student2,73,20,92
Student3,82,30,78
Student4,92,12,61
```


Section 6. Program Execution

Execution Sequence 1:

- *Load*
- *Register new item*
- *Register new student*
- *Display*

Record Manager Prompt: Version 2

```
<MAIN OPTIONS>
DISPLAY CURRENT RECORDS ----- A
REGISTER NEW STUDENT ----- B
REGISTER NEW ITEM ----- C
LOAD FROM FILE ----- D
SAVE TO FILE ----- E
EXIT ----- F
>>D

<LOAD FROM FILE>
Enter a valid filename: (default filename will be used if the input is blank)
>>
Load from "data.txt" complete

<MAIN OPTIONS>
DISPLAY CURRENT RECORDS ----- A
REGISTER NEW STUDENT ----- B
REGISTER NEW ITEM ----- C
LOAD FROM FILE ----- D
SAVE TO FILE ----- E
EXIT ----- F
>>C

<REGISTER NEW ITEM>
Required argument format: "name,score,score,score,..."
Required minimum: 5 arguments (1 + 4)
>>Statistics,90,90,90,90

<MAIN OPTIONS>
DISPLAY CURRENT RECORDS ----- A
REGISTER NEW STUDENT ----- B
REGISTER NEW ITEM ----- C
LOAD FROM FILE ----- D
SAVE TO FILE ----- E
EXIT ----- F
>>B

<REGISTER NEW STUDENT>
Required argument format: "name,score,score,score,..."
Required minimum: 5 arguments (1 + 4)
>>Anonymous Student,100,100,100,100

<MAIN OPTIONS>
```

```

DISPLAY CURRENT RECORDS ----- A
REGISTER NEW STUDENT ----- B
REGISTER NEW ITEM ----- C
LOAD FROM FILE ----- D
SAVE TO FILE ----- E
EXIT ----- F
>>A
    <DISPLAY CURRENT RECORDS>
Statistics of 5 student(s)
(Data Table) | Algebra | Calculus | Geometry | Statistics
Student1    | 72      | 12       | 78       | 90
Student2    | 73      | 20       | 92       | 90
Student3    | 82      | 30       | 78       | 90
Student4    | 92      | 12       | 61       | 90
Anonymous Student | 100     | 100      | 100      | 100
Mean        | 83.8000 | 34.8000  | 81.8000  | 92.0000

    <MAIN OPTIONS>
DISPLAY CURRENT RECORDS ----- A
REGISTER NEW STUDENT ----- B
REGISTER NEW ITEM ----- C
LOAD FROM FILE ----- D
SAVE TO FILE ----- E
EXIT ----- F
>>F

```

Execution Sequence 2:

- *Register 2 new items*
- *Register 2 new students*
- *Display*
- *Save*

Record Manager Prompt: Version 2

```

    <MAIN OPTIONS>
DISPLAY CURRENT RECORDS ----- A
REGISTER NEW STUDENT ----- B
REGISTER NEW ITEM ----- C
LOAD FROM FILE ----- D
SAVE TO FILE ----- E
EXIT ----- F
>>C
    <REGISTER NEW ITEM>
Required argument format: "name,score,score,score,..."
Required minimum: 1 arguments (1 + 0)
>>Differential Geometry

    <MAIN OPTIONS>
DISPLAY CURRENT RECORDS ----- A
REGISTER NEW STUDENT ----- B

```

```

REGISTER NEW ITEM ----- C
LOAD FROM FILE ----- D
SAVE TO FILE ----- E
EXIT ----- F
>>C

<REGISTER NEW ITEM>
Required argument format: "name,score,score,score,..."
Required minimum: 1 arguments (1 + 0)
>>Probability & Statistics

<MAIN OPTIONS>
DISPLAY CURRENT RECORDS ----- A
REGISTER NEW STUDENT ----- B
REGISTER NEW ITEM ----- C
LOAD FROM FILE ----- D
SAVE TO FILE ----- E
EXIT ----- F
>>B

<REGISTER NEW STUDENT>
Required argument format: "name,score,score,score,..."
Required minimum: 3 arguments (1 + 2)
>>Leonhard Euler,100,100

<MAIN OPTIONS>
DISPLAY CURRENT RECORDS ----- A
REGISTER NEW STUDENT ----- B
REGISTER NEW ITEM ----- C
LOAD FROM FILE ----- D
SAVE TO FILE ----- E
EXIT ----- F
>>B

<REGISTER NEW STUDENT>
Required argument format: "name,score,score,score,..."
Required minimum: 3 arguments (1 + 2)
>>Friedrich Gauss,100,100

<MAIN OPTIONS>
DISPLAY CURRENT RECORDS ----- A
REGISTER NEW STUDENT ----- B
REGISTER NEW ITEM ----- C
LOAD FROM FILE ----- D
SAVE TO FILE ----- E
EXIT ----- F
>>A

<DISPLAY CURRENT RECORDS>
Statistics of 2 student(s)
(Data Table) | Differential Geometry | Probability & Statistics
Leonhard Euler | 100 | 100
Friedrich Gauss | 100 | 100
Mean | 100.0000 | 100.0000

```

```

    <MAIN OPTIONS>
DISPLAY CURRENT RECORDS ----- A
REGISTER NEW STUDENT ----- B
REGISTER NEW ITEM ----- C
LOAD FROM FILE ----- D
SAVE TO FILE ----- E
EXIT ----- F
>>E

    <SAVE TO FILE>
Enter a valid filename: (default filename will be used if the input is blank)
>>
Save to "data.txt" complete

    <MAIN OPTIONS>
DISPLAY CURRENT RECORDS ----- A
REGISTER NEW STUDENT ----- B
REGISTER NEW ITEM ----- C
LOAD FROM FILE ----- D
SAVE TO FILE ----- E
EXIT ----- F
>>F

```

Contents of “data.txt” After the Execution

```

2,2
Differential Geometry,Probability & Statistics
Leonhard Euler,100,100
Friedrich Gauss,100,100

```