# Collections & File I/O

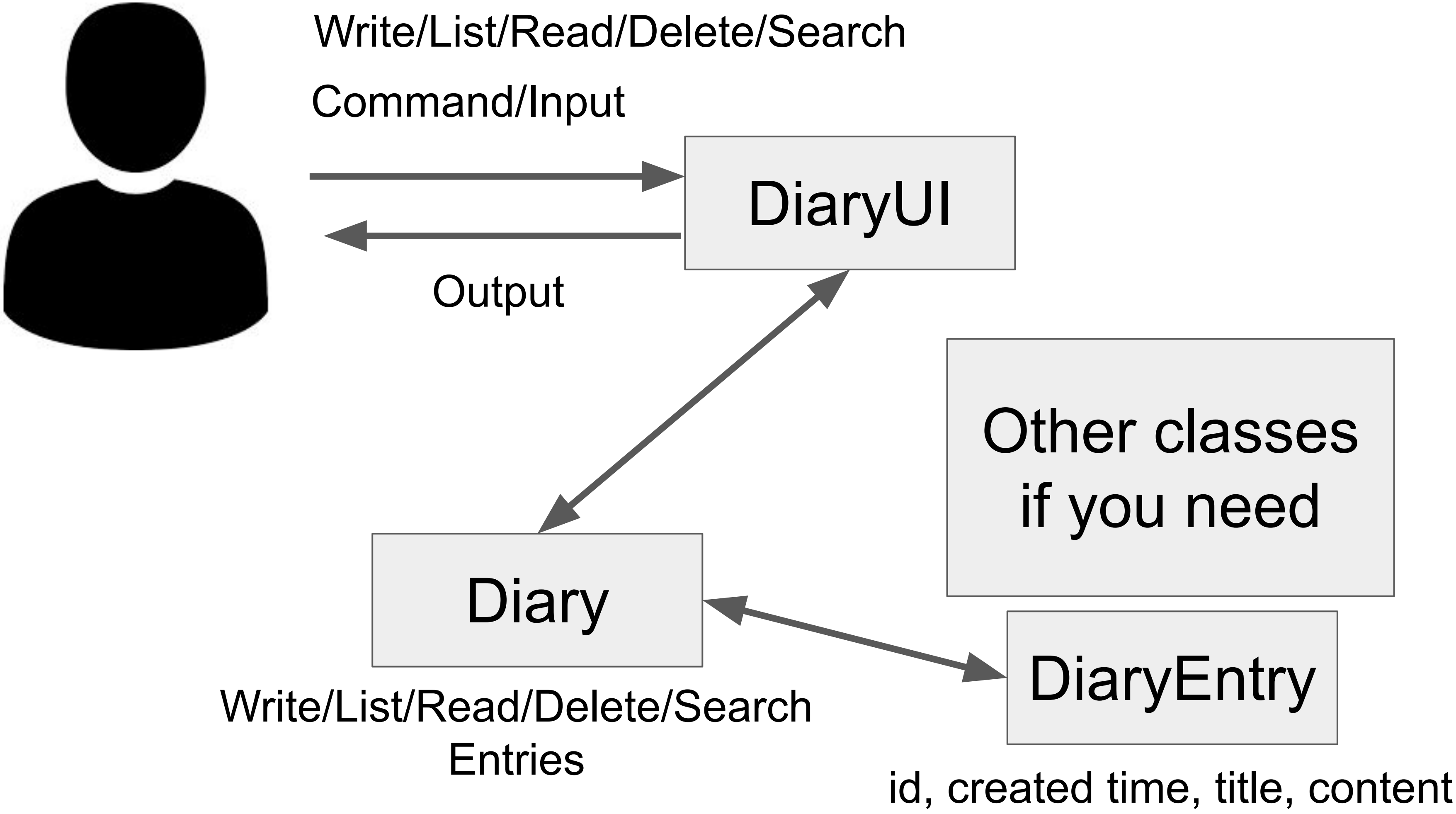
Lab 7

TA : Changmin Jeon, Minji Kim, Hyunwoo Jung,
Hyunseok Oh, Jingyu Lee, Seungwoo Jo

# Today’s Goal

- To properly use Java Collections
- To practice Java File I/O

# Practice 1 - Simple Diary Application (1)



Write/List/Read/Delete/Search

Command/Input

DiaryUI

Output

Other classes if you need

Diary

DiaryEntry

Write/List/Read/Delete/Search
Entries

id, created time, title, content

# Diary Application

- Implement a diary application using collections.
- A user should be able to write/list/read/delete/search diary entries.
- DiaryUI class is already implemented. It contains methods to get user commands/inputs, print messages, and handle some errors.
- You don’t have to care about exceptions not shown in this ppt.
- Implement createEntry, listEntries, readEntry, deleteEntry and searchEntry methods of Diary class.

# Output - Create Entries

- Create a diary entry with title and content.
- Each entries should have its own unique id when created.
- Assume that title input contains only alphanumeric characters and spaces(‘ ‘).

```
Type a command
   create: Create a diary entry
   list: List diary entries
   read <id>: Read a diary entry with <id>
   delete <id>: Delete a diary entry with <id>
   search <keyword>: List diary entries whose contents
contain <keyword>
Command: create
   title: First Entry
   content: Dear Diary, Life is beautiful.
   The entry is saved.
```

# Output - List Entries

- List (id, created time, title) of the diary entries you created before.
- The listed entries should be sorted in created time, by descending order. Print nothing if the list is empty.

```
Type a command
   create: Create a diary entry
   list: List diary entries
   read <id>: Read a diary entry with <id>
   delete <id>: Delete a diary entry with <id>
   search <keyword>: List diary entries whose contents contain
<keyword>
Command: list
   id: 3, created at: 2020/10/21 11:55:30, title: Third Entry
   id: 2, created at: 2020/10/21 11:48:30, title: Self Reflection
   id: 1, created at: 2020/10/21 11:47:28, title: First Entry
```

# Output - Read Entries

- Show (id, created time, title, content) of the diary entry selected by the id.

```
Type a command
   create: Create a diary entry
   list: List diary entries
   read <id>: Read a diary entry with <id>
   delete <id>: Delete a diary entry with <id>
   search <keyword>: List diary entries whose contents
contain <keyword>
Command: read 1
   id: 1
   created at: 2020/10/21 11:47:28
   title: First Entry
   content: Dear Diary, Life is beautiful.
```

# Output - Read Entries

- If there is no entry that has the input id, print an error message.

```
Type a command
   create: Create a diary entry
   list: List diary entries
   read <id>: Read a diary entry with <id>
   delete <id>: Delete a diary entry with <id>
   search <keyword>: List diary entries whose contents
contain <keyword>
Command: read 5
   There is no entry with id 5.
```

# Output - Delete Entries

- Delete a diary entry by id.

```
Type a command
    (...)
Command: delete 1
    Entry 1 is removed.

Type a command
    (...)
Command: list
    id: 3, created at: 2020/10/21 11:55:30, title: Third Entry
    id: 2, created at: 2020/10/21 11:48:30, title: Self Reflection
```

# Output - Delete Entries

- If there is no entry that has the input id, print an error message.

```
Type a command
    (...)
Command: delete 5
    There is no entry with id 5.

Type a command
    (...)
Command: list
    id: 3, created at: 2020/10/21 11:55:30, title: Third Entry
    id: 2, created at: 2020/10/21 11:48:30, title: Self Reflection
```

# Search Entries

- The user should be able to search entries which contain a given keyword exactly in their titles or contents. The search should be case-insensitive.
  - ex) title: First Entry  
    keyword: First ⇒ First Entry (O)  
    keyword: first ⇒ First Entry (O)  
    keyword: entr ⇒ First Entry (X)
- Use `split(“\\s”)` to split keywords.
- Show (id, created time, title, content) of the diary entries searched by the keyword. The entries don’t need to be sorted. Just check whether all the results are printed. Print a blank line between the entries.
- If there is no entry that contains keyword in the title or content, print an error message.

# Output - Search Entries

```
Type a command
   (…)
Command: search Entry
   id: 1
   created at: 2020/10/21 11:47:28
   title: First Entry
   content: I want to become a great engineer!

   id: 3
   created at: 2020/10/21 11:55:30
   title: Third Entry
   content: I want to become a great engineer!
```

# Output - Search Entries

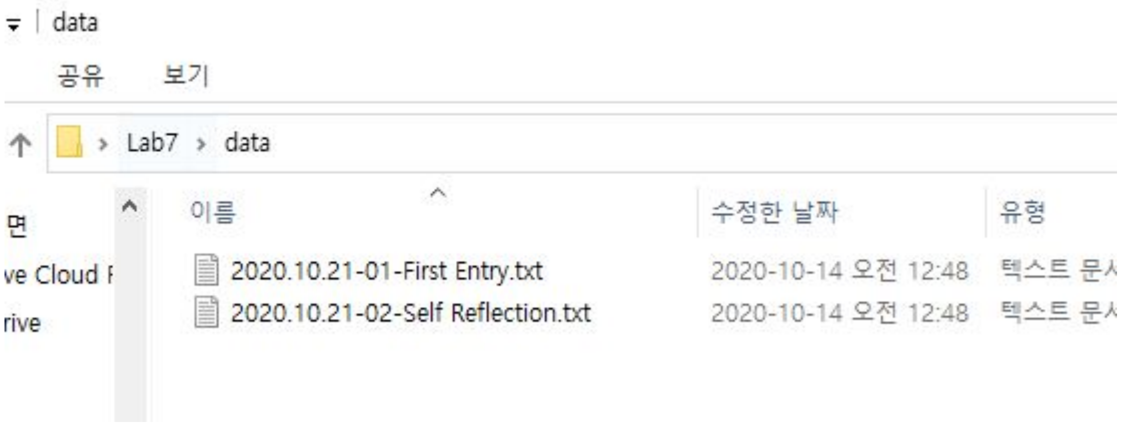
```
Type a command
   (…)
Command: search Third
   id: 3
   created at: 2020/10/21 11:55:30
   title: Third Entry
   content: I want to become a great engineer!



Type a command
   (…)
Command: search Thi
   There is no entry that contains “Thi”.
```

# Practice 2 - Simple Diary Application (2)

- If you turn off your diary application, the entries will be erased. Upgrade your application using file I/O.
- When you create an entry, save it in a file.
  - Create a file named “{YYYY.MM.DD}-{id}-{title}.txt”.
    - For example, 2020.10.21-01-First Entry.txt
  - Save the file in the “data” directory.
  - You can use your own format to properly save the entry.



| 이름 | 수정한 날짜 | 유형 |
|---|---|---|
| 2020.10.21-01-First Entry.txt | 2020-10-14 오전 12:48 | 텍스트 문 |
| 2020.10.21-02-Self Reflection.txt | 2020-10-14 오전 12:48 | 텍스트 문 |

# Practice 2 - Simple Diary Application (2)

- When you read an entry, load the file and read the content.
- When you delete an entry, delete the whole file.
  - You don’t need to care about the case where the file name doesn’t exist when reading or deleting
- All the commands should operate as same as before.
- You may have to load all the entries when you turn on the application, and keep track them to properly operate the commands.

# Submission

- Compress your final `src` directory into a zip file.
  - After unzipping, the ‘src’ directory must appear.
- Rename your zip file as 20XX-XXXXX_{name}.zip - for example, 2020-12345_KimMinji.zip
- Upload it to eTL - Lab 7 assignment.