Spring 2015 CS151 Programming Assignment 2
Calendar
Instructor: Dr. Kim
Due: Friday March 13 11:59 pm

- Study java.util.GregorianCalendar to learn how to manipulate day, week and month of calendar. Also, here is an example I wrote for you to reference.
- In your implementation, ignore cases to recognize a user request and make sure to follow the given example format to enter data.

In this assignment, you will design and implement a calendar similar to one you can find in your phone. The calendar is going to be implemented as a console application.

The initial screen shows the current month looking like this. It also highlights today's date, for example, using a pair of brackets.

```
  February 2015
Su Mo Tu We Th Fr Sa
 1  2  3  4  5  6  7
 8  9 10 11 12 13 14
15 16 17 18 19 20 21
22 23 [24] 25 26 27 28
```

The initial screen comes with a main menu with following options: View by, Create, Go to, Event list, Delete, and Quit. After the function of an option is done, the main menu is displayed again for the user to choose the next option.

```
Select one of the following options:
[L]oad   [V]iew by  [C]reate, [G]o to [E]vent list [D]elete  [Q]uit
```

The user may enter one of the letter highlighted with a pair of bracket to choose an option. For example,

```
V
```

will choose the View by option.

- [L]oad
  The system loads events.txt to populate the calendar. If there is no such file because it is the first run, the load function prompts a message to the user indicating this is the first run. You may use Java serialization this function.
- [V]iew by
  User can choose a Day or a Month view. If a Day view is chosen, the calendar displays the current date. If there is an event(s) scheduled on that day, display them in the order of start time of the event. With a Month view, it displays the current month and highlights day(s) if any event scheduled on that day. After a view is displayed, the calendar gives the user three options: P, N, and M, where P, N, and M stand for previous, next, and main menu, respectively. The previous and next options allow the user to navigate the calendar back and forth by day if the calendar is in a day view or by month if it is in a month view. If the user selects m, navigation is done, and the user gets to access the main menu again.

  ```
  [D]ay view or [M]view ?
  ```

  If the user selects D, then

  ```
  Tuesday, Feb 24, 2015
  Dr. Kim 13:15 - 14:00
  ```

  ```
  [P]revious or [N]ext or [M]ain menu ?
  ```

  If the user selects M, then

```
 February 2015
Su Mo Tu We Th Fr Sa
 1  2  3  4  5  6  7
 8  9 10 11 12 13 14
15 16 17 18 19 20 21
22 23 24 25 26 27 28
```

```
[P]revious or [N]ext or [M]ain menu ?
```

- [C]reate
  This option allows the user to schedule an event. The calendar asks the user to enter the title, date, starting time, and ending time of an event. For simplicity, we consider one day event only. Also, let's assume there is no conflict between events that user entered, and therefore your program doesn't have to check if a new event is conflict with existing events. Please stick to the following format to enter data:
  - Title: a string (doesn't have to be one word)
  - date: MM/DD/YYYY
  - Starting time and ending time: 24 hour clock such as 06:00 for 6 AM and 15:30 for 3:30 PM. The user may not enter ending time if an ending time doesn't make sense for the event (e.g. leaving for Korea event may have a starting time but no ending time.)
- [G]o to
  With this option, the user is asked to enter a date in the form of MM/DD/YYYY and then the calendar displays the Day view of the requested date including any event scheduled on that day in the order of starting time.
- [E]vent list
  The user can browse scheduled events. The calendar displays all the events scheduled in the calendar in the order of starting date and starting time. An example presentation of events is as follows:

```
2015
  Wednesday, February 25 13:15 - 14:00 Dentist
  Thursday March 12      15:00 - 16:00 Job Interview
2016
  Friday June 3 17:00 Leave for Korea
```

- [D]elete
  User can delete an event from the Calendar. There are two different ways to delete an event: Selected and All. Other type of deletion will not be considered for simplicity.
  - [S]elected: all the events scheduled on the selected date will be deleted.
  - [A]ll: all the events scheduled on this calendar will be deleted.

```
[S]elected or [A]ll ?
```

  If the user enters s, then the calendar asks for the date as shown below.

```
Enter the date.
```

```
03/12/2015
```

- [Q]uit saves all the events scheduled in a text file called "events.txt" in the order of starting date and starting time. You may use Java serialization if you don't want to persist data in a text file.

The main menu will be displayed after each option is done. It is crucial to have a user friendly interface for the user to enter input. For example, if the calendar needs a date from the user, suggest a specific format of the date for the user to use. Our class grader will be the user to operate your calendar, and you don't want to frustrate the user with a confusing interface.

Deliverable

Softcopy of your implementation through the course web page:

- All source programs you wrote (.java) required to run the application.
  - Name the class with main method as MyCalendar

    - Put javadoc comments in the source codes.
    - Submit .java files only.
    - Do NOT use packages.
    - Put all .java files in a directory called **hw2**
    - Put input.txt (described below) in the **hw2** directory as well for input redirection.
    - Put output.txt as a result of output redirection. You may produce output.txt as shown below.

      ```
      C:\>java MyCalendarTester  <  input.txt  > output.txt
      ```

  - In the **hw2** directory, run

    ```
    javadoc -d ./doc *.java
    ```

    That will automatically generate all the javadoc pages and put them in the **doc** directory.
  - Zip **hw2** to **hw2.zip** and submit it.

- events.txt (get this result from an exemplary run of your MyCalendarTester)
- input.txt for input redirection as shown below.