Com S 207: Programming Assignment 6
350 Points
Due Data: April 16 23:59pm

In this assignment you will learn about array lists and reading from files. This assignment requires you to write a class named `WeaklyCalendar`. This class must be in a package named `pa6`.

Note that the description of a programming assignment is not a linear narrative and often requires multiple readings before things start to click. You are encouraged to consult instructors/Teaching Assistants for any questions/clarifications regarding programming assignments.

1 Calendar

You will be writing a program to implement a baby version of Google calendar/iCal to keep track of appointments. A Calendar stores all appointments. You can add an appointment to the calendar or delete an existing appointment. For our purposes, we work with weekly calendar (not monthly/yearly calendars).

Each appointment has two fields, day of the appointment and time of the appointment Suppose you have an appointment on Monday at 5PM. This information can be represented as string of the form "Monday@5PM". Similarly an appointment on Tuesday at 8AM can be represented with the string "Tuesday@8AM".

For the sake of this program we make some simplifying assumption. Every appointment starts at the top of the hour, and lasts for at most one hour. For example, we do not allow appointments that start at 5:30PM.

All the data about appointments is initially stored in a text file named `myCalendar.txt`. This file consists of words and each word describes an appointment. Here is a sample content of the file

```
Monday@5PM
Tuesday@2PM
Friday@6PM
```

Note that each appointment is described by a *single* word.

Your task is to write a program that reads the appointment data from the file named `myCalenda.txt`. Once it reads the data from this file, the program interacts with the user. The use can add an appointment, remove and appointment or print the calendar on the screen etc.

2 Your Task

Your task is to a write program named `WeeklyCalendar` that keeps track of all appointments. For this you will be writing several methods. Your program must have the following method.

`readFromFile.` This method read appointments from a file named `myCalendar.txt`. The format of the file is as described above. It returns an array list of Strings. Each word of the file

`myCalendar.txt` will be an element of the array list. For example if the contents of the file are as described above, then this method returns an array list (of Strings) of size 3. The string at index 0 is "Monday@5PM", String at index 1 is "Tuesday@2PM", and String at index 2 is "Friday@6PM". From now we will refer to this list as *appointment list*.

`hasAppointmentAt.` This method takes three parameters with following names and types.

- `appointmentList.` This is an array list of strings consisting of all appointments.
- `specifiedDay.` Weekday.
- `specifiedTime` A string representing time. This of form "hhAM" or "hhPM". For example, "9AM" or "10PM". You may assume that minutes are not specified.

The method searches the `appointmentList` for an appointment on `specifiedDay` at `specifiedTime`. If found, the method returns true. Otherwise, the method returns false.

`addAppointment.` This method takes three parameters with following names and types.

- `appointmentList.` This is an array list of strings consisting of all appointments.
- `specifiedDay.` Weekday.
- `specifiedTime` A string representing time. This of form "hhAM" or "hhPM". For example, "9AM" or "10PM". You may assume that minutes are not specified.

The method first searches whether there is an existing appointment at `specifiedTime` on `specifiedDay`. If so, then this method returns `false`. Otherwise, this method adds an appointment to the `appointmentList` and returns `true`

`removeAppoingtment.` This method takes three parameters with following names and types.

- `appointmentList.` This is an array list of strings consisting of all appointments.
- `specifiedDay.` Weekday.
- `specifiedTime` A string representing time. This of form "hhAM" or "hhPM". For example, "9AM" or "10PM". You may assume that minutes are not specified.

The method first searches whether there is an existing appointment at `specifiedTime` on `specifiedDay`. If so, then this method returns removes that appointment and returns `true`. Otherwise, this method returns `false`.

`appointmentsOnDay.` This method takes two parameters with following names and types.

- `appointmentList.` This is an array list of strings representing all appointments.

- `specifiedDay`. A string representing a week day.

This method returns an array list (of Strings) consisting of all appointments on the `specifiedDay`. `printAppointmentCalendar`. This method takes a variable named `appointmentList` (which is an array list of Strings) representing all appointments. This prints the appointment onto the screen. Each line of the output starts with weekday followed by appointments on that day. Since there are seven weekdays, the output has 7 lines. An example output is as follows:

```
Monday: 2PM  8AM 3PM
Tuesday:
Wednesday:  11AM  12PM
Thursday:  2PM 8AM 11AM
Friday:
Saturday: 9AM 10AM 12PM 5PM 8PM
Sunday:
```

`main`. This is the main method. The main method first reads the appointments from a file named `myCalendar.txt` and stores them in an array list. Then it repeatedly prompts the user to enter one of "add", "remove", "printDay", "printAll" or "quit". When the user enters "quit" the program terminates.

3 Coding Conventions

You must follow good coding conventions. Variable names must be meaningful, variable names (and method names) must start with lower case letters, code must be properly indented. Your code must have appropriate comments. Failure to follow good coding conventions will cause you to lose points (even if your solution is correct).

4 Suggestions

Do not attempt to write code for all the methods at once. Write one method at a time, test that the method works correctly then proceed to the next method.

5 Specifications

You must follow the specifications exactly. Your program must be named `WeeklyCalendar` and should be in a package named `pa6`. Please note that Java is case-sensitive. The input-output behavior of your program must be exactly as described. Failure to follow the specifications (even if your solution is correct) will cause you to lose points. Submit `WeeklyCalendar.java` via Canvas. Do not submit `WeeklyCalendar.class`.