**P12.9** Write a program to design an appointment calendar. An appointment includes the date, starting time, ending time, and a description; for example,

Dentist­2012/10/1­17:30­18:30 CS1­class­2012/10/2­08:30­10:00

Supply a user interface to add appointments, remove canceled appointments, and print out a list of appointments for a particular day. Follow the design process that was described in this chapter. Your solution should include a class Appointment- Calendar that is not coupled with the Scanner or PrintStream classes.

This programming assignment is modified somewhat, so be sure to read the following!

Each appointment will appear as an input file line like this:

OneTime,4,10,2016,12:00 AM,Leash's Birthday

Appointments will be categorized as OneTime, Monthly, or Daily.

You will want to add a good user interface that would allow the user to get the appointments for, say:

1. Today
2. Tomorrow
3. Seven days (including today)
4. A specific date

If a 1, 2 or 3 is entered, the appointments are listed. (It does **not** ask for today’s date …)

If a 4 is entered, then the user is prompted for the desired date, and the appointments are then listed.

In addition, you will want to allow for new appointments to be entered and existing appointments to be cancelled. These changes would be used to create a new, updated input file that would be written each time the user quits the program. The interface, then, would be extended:

1. Cancel an appointment
2. Add an appointment
3. Close this program

It would be good if your program included a check for valid dates. In other words, the user could not enter 2/29/2017, since February does not have 29 days next year. And as appointments are added, the validity of the date and time should be checked.

Use this program to show off everything that you can do with Java! (For example, maybe you would want to add an option that would also the user to search for all appointments containing “dentist”.)

Your program should have a getData method. getData should open the file, gather the data, close the file, and then use a return statement to send it back to the main program. You will also have a putData method that will be called at the end of the program so that the updated information resulting from additions and cancellations is written to the file.

Also note that you should have a superclass for Appointment, and then subclasses for the variations (e.g. Daily, OneTime, Monthly).

Again you will have a file with the main function, one [or more] with the classes, one for getData, and one for putData.