Instructions for Appointment Hierarchy and Tester Class:

A modified version of Project 9 in Big Java, Chapter 9:

Implement an **abstract** superclass called Appointment and subclasses Onetime, Daily, and Monthly.  In the superclass, define a **description** and a person’s **last name** as private attributes, with corresponding public getters and setters. In the superclass, also define a private int military time attribute of **hour** (00 – 23) and a private int military time attribute of **minutes** (00 – 59). Then, create an abstract public boolean method, **occursOn**, with the following signature: **public boolean  occursOn**(int year, int month, int day);  
  
The hierarchy of classes should be as follows:

Appointment is the superclass.   
Daily is a subclass of Appointment.  
Monthly is a subclass of Appointment.  
OneTime is a subclass of Appointment.

Decide which subclasses should include the attributes of hour, minutes, day, month, and year, or any combination of these. Remember not to include any of the attributes that are not needed for that specific type of appointment. Also, take advantage of the hierarchy, and carefully select what attribute(s) can be defined at a superclass level so that its subclasses can inherit the attribute. Hint: Go from general in the superclasses, to specific in the subclasses. All subclasses need to know the hour and minutes, but not all subclasses need to know the day, month, and year, since some appointments are daily, or monthly.

The boolean public method **occursOn**(int year, int month, int day) should be defined in each subclass, as each subclass implements it in its own way. Think about what is required to see if an appointment occursOn a given date when the appointment is a one-time appointment, a monthly appointment, or a daily appointment.

In the driver class:

Code 2 methods:

 1.  Make Appointments - Fills an array of Appointment objects, with a mixture of appointments and dates an times.

 2.  Check Appointments - Asks the user to enter a specific date, and print out all the appointments that occur on that date.

Make sure you implement the polymorphic method occursOn(…) that can be called by any of the objects in the Appointment array.

A sample of the output of the program is as follows:

How many appointments do you wish to make? 3  
  
Please make a selection:  
1. One Time Appointment  
2. Daily Appointment   
3. Monthly Appointment   
  
1  
What is the description of your appointment? Dentist Checkup

What is your person’s name? Dr. Smith

What year is your appointment? (2017 - 2018) 2017

What month is your appointment? (1 - 12) 03

What day is your appointment? (1 - 31) 15

What is the hour of your appointment? (00 – 23) 13

What is the minutes of your appointment? (00 – 59) 30

**Appointment added with Dr.Smith on 03/15/2017 at 13:30**

Please make a selection:  
  
1. One Time Appointment  
2. Daily Appointment   
3. Monthly Appointment   
-----------------------------------------------

2  
What is the description of your appointment? Athletic Training

What is your person’s name? Ms. Jones

What is the hour of your appointment? (00 -23) 10

What is the minutes of your appointment? (00 -59) 00

**Appointment added with Ms. Jones Daily at 10:00**

Please make a selection:  
  
1. One Time Appointment  
2. Daily Appointment   
3. Monthly Appointment 

------------------------------------------------

3  
What is the description of your appointment? Piano Lessons

What is your person’s name? Ms. Katie

What day of the month is your appointment? (1 - 31) 15

What is the hour of your appointment? (00 -23) 17

What is the minutes of your appointment? (00 -59) 30

**Appointment added with Ms. Katie Monthly on day 15 at 17:30**

\*\*\*\*\*\*\*\*\*\*\*\*\*Thank you for making all of your appointments.\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
  
Loop with the following:  
  
***What is the date you wish to look up in your Appointments' Calendar?***   
Enter the month: 03  
Enter the day: 15  
Enter the year: 2017  
  
**On 3 / 15 / 2017 you have the following appointments:**   
Dentist appointment with Dr. Smith at 13:30  
Piano Lessons with Ms. Katie at 17:30  
Athletic Training with Ms. Jones at 10:00  
  
  
Do you wish to look up another date?

(If they answer NO, exit the program with message)  
 “**Thank you** for using your appointment calendar.”  
  
 (If they answer YES, continue to ask for another date to look up).  
 ***“What is the date you wish to look up in your Appointments' Calendar?***”  
 Etc.

HINT: Try to use the hierarchy of .toString(), so that each level returns a specific string of values available at that level only.

Extra Credit:

Validate that the **hour of time and date** of a proposed appointment do not conflict with an appointment that is already in the array of appointments. If there is a conflict, as user to select another date or time, and validate again.