CSCI 345 - Object Oriented Design

Assignment 02

Design Patterns – Listener Pattern – Timer

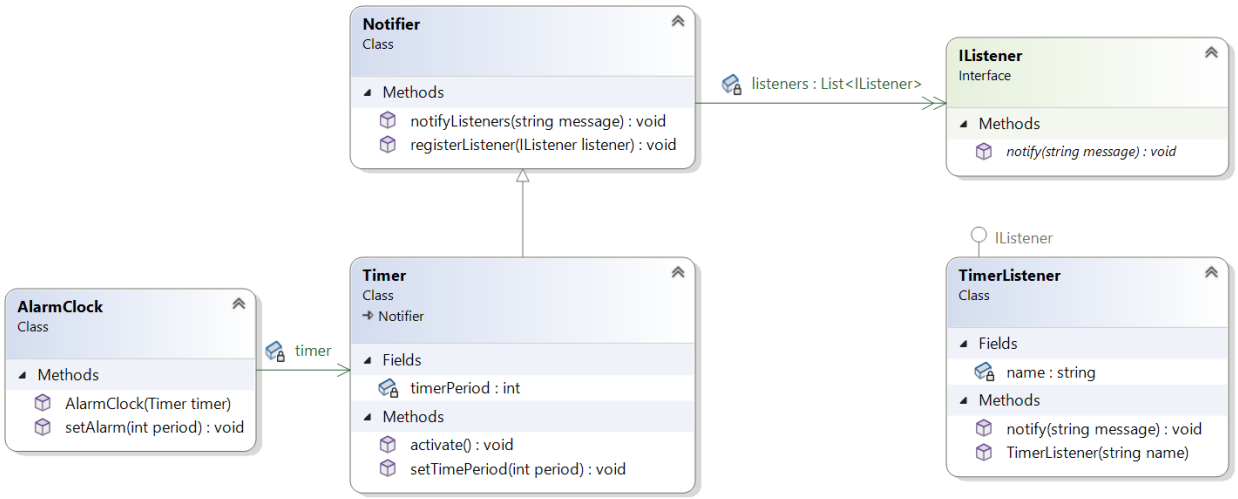
Overview

This assignment is designed to give you help understanding the Listener design pattern . You can work with one other student on this assignment, or you can elect to complete the assignment on your own.

Program Specification

Mr. Pumphrey, the renowned project manager of all quirky projects, has tasked you, the newly hired junior software developer, to implement the following UML Class Diagram:

UML Class Diagram



1. The program is designed to model the interaction between an AlarmClock object, a Timer object and a TimerListener object.

2. Create a class named AlarmClock. Define a constructor that receives a reference to a Timer object. The AlarmClock.setAlarm method should call the Timer.setTimePeriod and the Timer.activate method.
3. Create a class named TimerListener that implements the IListener interface. Define a constructor that receives a string and assigns the string parameter to a private string field named name that represents the TimerListener's name. The notify method should simply output a message to the console that outputs the value in the name field and a message. For example, if the name field is storing the string "Sally", and the message "wake up!" is passed to the notify method, the output should be "Sally: wake up!". The string "wake up!" should be hard-coded in the Timer.activate method.
4. Create a class named Timer that subclasses Notifier. The Timer class should have a private integer field named timerPeriod that represents the amount of time in seconds the timer should be set for.
5. Create a method in the Timer class named setTimePeriod that receives an integer parameter. The method should use the parameter to set the timerPeriod field.
6. Create a method in the Timer class named activate that receives no parameters and has a void return type. The activate method should activate the timing process and call the IListener notify method when the timer goes off. Specifically, the activate method should call the Thread class's static sleep method to suspend the thread's execution as shown here:

   Java: Thread.sleep(timerPeriod * MILLISECONDS_PER_SECOND);
   - The amount of time the thread is suspended is based on the value in the Timer class's timerPeriod field.
   - Output the message "Activating timer…" when the Timer class's Activate method is executed.

You can download a test program file from Canvas->Files-> Assignments-> Assignment_02.

Program Execution

Sample Program Execution

```
Running the Backpack application...

Activating timer...
Mickey: Wake up!
Donald: Wake up!
```

Comments

At the top of all source code files, add the following commenting:

```
/*
#  Name:
#  Date:
#  Description:
*/
```

Submission

Upload all source code files to Canvas. For this assignment, Canvas has been configured to permit only files that end with the .java file extension.