

Inner classes

Mini-project 1

Let consider a semaphore system which manages up to 100 semaphores.

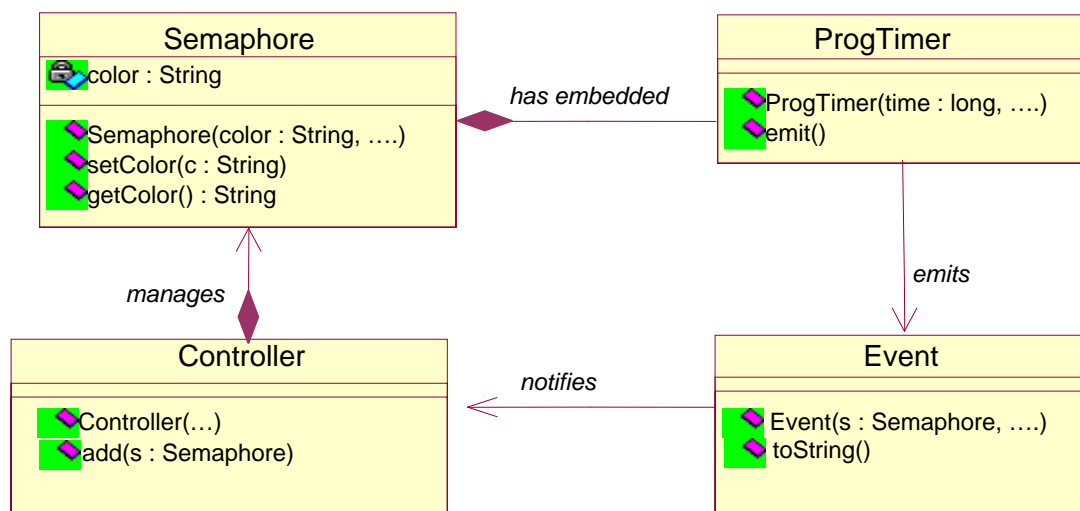
A semaphore can have three color lights: red, yellow, and green and its behaviour is based on the following rules:

1. The only transitions of the color light are: red->yellow, yellow->green, green->red.
2. The semaphore remains with the color red for RED_LAPSE seconds (i.e. 30 seconds), the color yellow for YELLOW_LAPSE seconds (i.e. 10 seconds), and the color green for YELLOW_LAPSE seconds (i.e. 35 seconds).
3. When the semaphore starts its activity the initial color is red.

To implement its behaviour the semaphore uses an embedded programmable timer which provides the semaphore with periodic events. Using these events the semaphore changes its current state, that is its color, at programmed lapses of time. The lapses are set up at the semaphore creation time.

Each time a semaphore changes the color, the controller prints the event's information: the semaphore's identifier (a unique integer), the semaphore's current color and the event time (hh:mm:ss).

A draft of the class model of the semaphore system is presented in the following diagram:



Requirements.

1. Complete the system diagram with missing information in order to support the above behaviour.
2. Write the program which implements the semaphore system.
3. Write the test program for the semaphore system. The program creates at least three semaphores with different color lapses of time and starts them.

Notes:

1. The class of the programmable timer is an inner class of the class Semaphore.
2. The current time is obtained with the static method `currentTimeMillis()` of the class `System`.