Object-Oriented Programming

Exercise session 10

Exercise 1

A programmer decides to write a simulator of a referendum. In his simulation, voters are each modeled by a separate thread while polling stations (i.e., the places where voters will go to record their vote) act as shared resources. Each voter is assigned to a given polling station. Then, at its assigned station, each virtual voter spends a few minutes to register to vote (this process is modeled in the simulation by a short delay), this amount varying from one voter to another, then gives his vote for the referendum. As several voters will go to a same given polling station, the programmer used synchronization mechanisms to simulate the voters queueing at said station.

Open the source code of the simulator, which you will find in a `referendum/` subfolder provided along with this document, and have a look at its various classes. In particular, review the `Voter` and `PollingStation` classes and observe how they communicate and the mechanisms they use for synchronization. Then, compile the simulator, run it a few times and answer the following questions.

— What is the practical effect of the `synchronized` keyword in the signature of the `welcome()` method in `PollingStation`? Why is it relevant in this simulation?

— Remove this keyword, re-compile the simulator and run it several times. What happens now? How does it affect the outcome?

— Suppose that the operations of the `welcome()` method are split into two separate methods, one for registering the voter and another one to get his or her decision. Would it change the outcome of the referendum? Why?

Exercise 2

In order to increase the realism of the simulator, the programmer wants you to expand it by modeling in it a foreign agent who will try to influence voters while they are queueing. The agent will temper with their Facebook feed in order to insert a story about workers from Park county (Colorado, USA) who claim immigrants took their jobs. Upon reading this news story, a voter has 10% more chances of voting *Leave.*

In the simulator, the foreign agent will be modeled by one additional thread class `ForeignAgent`, which will be instantiated once. This thread will have references to all `Voter` objects in order to be able to influence them. However, you should also take account of the fact that a voter can only be influenced if he or she consults his or her Facebook feed while queueing to vote. The chances for this event to occur are of 10% for retired people, 25% for working people and 40% for students. After implementing it, include the `ForeignAgent` thread to the simulation and run it.

Tips :
— Modify the `Voter` class so the foreign agent is able to influence all voters.
— To check from what part of the population a voter is from, use the `instanceof` operator.

Exercise 3

In modern jogging events, such as the *15km de Liège Métropole* or the *Maasmarathon*, each participant wears a bib [1] which is designed to notify the starting and the finish lines when (s)he has crossed either of them. Thanks to this bib, a runner can be timed accurately, but it also allows the organizers to automatically display his/her name on a monitor (along with the time) as soon as (s)he crosses the finish line.

Your task is to simulate such a monitor with the `wait()` and `notify()` methods. To help you, you will find a `Runner` class in the subfolder `jogging` provided along with this statement, as well as a `TimeReporting` interface. Have a look at them first.

Then, create a `FinishLine` class implementing the `TimeReporting` interface as well as a `Monitor` thread class which receives an amount of runners to monitor and which the task consists in consulting the `FinishLine` until all runners have crossed it. Finally, create a test program to ensure your implement works.

Tips :
— You can assume a `Runner` crosses the starting line as soon as its thread starts.

— You can take inspiration from the example program shown in the last slide of the theoretical course. Don't forget to also carefully review the previous slides as well.

---

1. FR : un dossard