UNIVERSIDAD DE MÁLAGA | uma.es

Dpto. Lenguajes y
Ciencias de la Computación

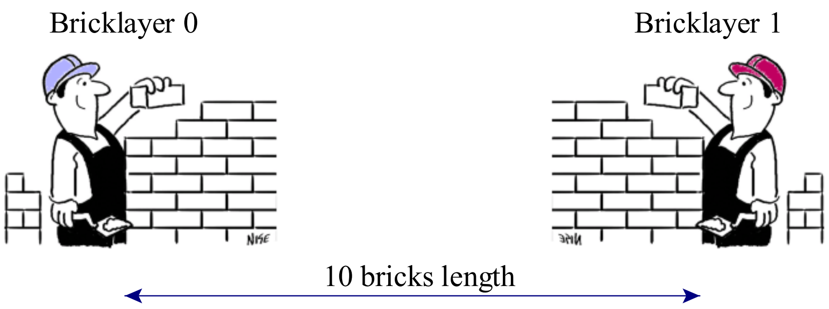
Systems Programming and Concurrency Control 30/5/2019

SURNAME____NAME____________

DNI/PASSPORT________________ COMPUTER ID _______________

Let's model a contest between two bricklayers. The winner is the bricklayer who firstly wins three sets. A set is a race to build a single brick wall with the next rules:

- The length of the wall is measured in linear bricks.
- Bricklayer 0 starts from the left and bricklayer 1 starts from the right.
- The set finishes when the brick wall is finished, i.e., the length of the wall is 10 bricks.
- Each bricklayer takes a random time to put each brick (between 10 and 100 milliseconds).
- The winner of the set is the first bricklayer who puts 6 or more bricks (10/2+1).
- If both bricklayers put 5 bricks then the set is null and there is no winner.
- The contest finishes when a bricklayer wins three sets.



You have to resolve this exercise with Semaphores (4 pts.), and Monitors/Locks (4 pts.). To do so you are already given:

- A class `Driver` that creates the shared resource (the Contest circuit) and the two bricklayers.
- A class `Bricklayer` that inherits from `Thread` that should simulate the behaviour of the players in its `run()` method (2 pts.).
- A class `Contest` that represents the shared resource, i.e. the contest. This class should provide the next functions:
    - `public void readySteadyGo(int id)`. This function synchronizes the two bricklayers through a handshake so they start at the same time.
    - `public void putBrick(int id)`. This function simulates that a bricklayer puts a brick and check if the wall has been finished; if so, then checks if there is any winner. In addition, the loser must check if the set has finished before putting additional bricks.
    - `public boolean isSetFinished()`. Returns true if the brick wall has been built completely.
    - `public boolean isContestFinished()`. Returns true if the contest has finished and there is a winner with three sets.

Notes:

1. The only documentation you can use is the Java API inside Eclipse.
2. Start with the already given source files.
3. Please, fulfil each source file with your name and surname.
4. To provide each of the two solutions, please use the corresponding Contest subclass.
5. When finished, pack the project with the solutions (.zip, .rar) and upload it to the Virtual Campus (there is a task for it).