UNIVERSIDAD DE MÁLAGA
Dpto. Lenguajes y Ciencias de la Computación

Systems Programming and Concurrency

ANDALUCÍA TECH
Campus de Excelencia Internacional
Campus of International Excellence

Exam of Java, June 2021

Description of the system

A new take-away café has been open in Teatinos. This café is small and has a limited capacity (**CAPACITY**). There is a single **Barista** in the café who is in charge of preparing coffee and charge to the clients. A **Client** may come into the café if the capacity is not exceeded, then waits for the Barista to prepare his/her coffee, pays the coffee and exits from the café. On the other hand, the Barista waits for some Client to be in the café, prepares his/her coffee and waits to be paid. Let's model the system with the next classes:

- **Barista**. A class that inherits from Thread and takes care of preparing coffee and charge the Clients of the Café (this class is already implemented).
- **Client**. A class that inherits from Thread and models a Client that comes into the Café and, once inside, asks for a coffee and, when s/he receives the coffee, pays it and exits (this class is already implemented).
- **Café**. This class models the shared resource. Inside the Café may be, at most, CAPACITY Clients. This class has the next methods that are used by the Barista and the Clients:
  - `public void enterCafe(int id)`: a client uses this method to come into the Café. He may enter iif the CAPACITY is not exceeded.
  - `public void waitCoffee(int id)`: a client waits in this method until the Barista prepares to him a cup of coffee.
  - `public void payAndExitCafe(int id)`: a Client uses this method to pay his coffee and to exit from the Café.
  - `public void prepareCoffee()`: the Barista waits for, at least, one Client inside the Café, then prepares a coffee for him and waits for he to pay. Each time this method is executed, the Barista prepares a single coffee and, hence, a single Client will be served.

In addition, there is class Driver that creates a Café instance, one Barista and several Clients. As an example, the execution of the program may lead to the next execution:

Client 1 comes in → Client 2 comes in → Coffee served, waiting payment → Client 2 takes coffe → Client 3 comes in → Client 2 pays and exits → Coffee served, waiting payment → Client 1 takes coffe → Client 1 pays and exits → …

You have to develop two implementations of the class Café:

1. Binary semaphores (5 points) or general semaphores (3 points).
2. Monitors (4 points) or Locks (5 points).