2nd Year CC computer Engineering
University of Batna 2, 2023/2024

Programmation orientée objet 2 (TP-06) Multithreading

New Java Keywords :

Thread

Runnable

synchronized

Thread.sleep(500)

Thread.CurrentThread().getNumber()

Exercise °1 (Thread, Runnable) :

Create a Java program that implements a counter using threads. The program should have two versions:

Version 1: Using extends Thread:

1. Create a class called CounterThread that extends the Thread class.
2. Inside the CounterThread class, override the run() method. In this method, implement a loop that counts from 1 to 10, printing each number to the console.
3. Create two instances of the CounterThread class and start both threads.
4. Ensure that both threads are started and running concurrently.
5. Test and observe the output.

Version 2: Using implements Runnable:

1. Create a class called Counter that implements the Runnable interface.
2. Inside the Counter class, implement the run() method. This method should perform the same task as in Version 1: counting from 1 to 10 and printing each number to the console.
3. Create two instances of the Counter class.
4. Create two Thread objects, passing each Counter instance as a parameter to their constructors.
5. Start both threads.
6. Ensure that both threads are started and running concurrently.
7. Test and observe the output.

Exercise °2 (synchronized) :

Suppose you have a shared bank account represented by the BankAccount class. Multiple threads represent withdrawals and deposits to this account. Without proper synchronization, concurrent access to the account can lead to incorrect results, such as incorrect balances or race conditions.

Create a BankAccount class with methods for withdrawing and depositing funds. Then, create multiple threads to perform withdrawals and deposits simultaneously. Use synchronization to ensure that the balance is updated correctly and prevent race conditions.

Class BankAccount

synchronized void withdraw(double amount)

synchronized void deposit(double amount)

class BankTransaction implements Runnable

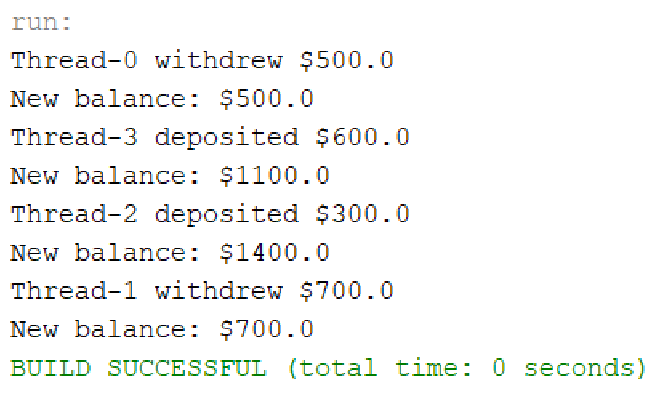
BankAccount account;

boolean isWithdrawal;

double amount;

void run()

At the end the code should give similar ouput :



```
run:
Thread-0 withdrew $500.0
New balance: $500.0
Thread-3 deposited $600.0
New balance: $1100.0
Thread-2 deposited $300.0
New balance: $1400.0
Thread-1 withdrew $700.0
New balance: $700.0
BUILD SUCCESSFUL (total time: 0 seconds)
```

Exercise °2 (Thread, mySQL and files) :

We will see it during the session