Marked Lab #3

The lab must be prepared in pair and submitted on Campus (see section « Marked Lab 3 - Instructions »).
Ce TP doit être réalisé en binôme et rendu sur Campus (voir la section « Marked Lab 3 - Instructions *»).*

Description - *Sujet*

You have to develop an application for a bank in Java. The application uses a relational database to store its data.

Il s'agit de développer en Java une application pour une banque. L'application s'appuie sur une base de données relationnelle pour stocker ses données.

The application template is defined in the Eclipse project provided with the lab. The main class is `BankManagerImpl`; it implements the `BankManager` interface. A sample test program is provided in the `SimpleTest` class. You have to implement the `BankManagerImpl` class and complete the `SimpleTest` class. **Please read carefully the javadoc** before you start coding.

Le squelette de l'application est défini dans le projet Eclipse fourni avec le TP. La classe principale est `BankManagerImpl` ; elle implémente l'interface `BankManager`. Un exemple de programme de test est fourni dans la classe `SimpleTest`. Vous devez implémenter la classe `BankManagerImpl` et compléter la classe `SimpleTest`. ***Lire svp attentivement le javadoc*** *avant de coder.*

The database must implement the following integrity constraint: the balance of an account cannot be negative (see the `addBalance()` method). The database must also automatically log the operations performed on the accounts (see the `getOperations()`method). These functions must not be implemented by the Java application; they must be implemented at the database level, using triggers.

La base de données doit implémenter la contrainte d'intégrité suivante : le solde d'un compte bancaire ne doit pas être négatif (voir la méthode `addBalance()`). Elle doit aussi automatiquement journaliser les opérations effectuées sur les comptes (voir la méthode `getOperations()`). Ces fonctions ne doivent pas être implémentées par l'application Java ; elles doivent être implémentées au niveau de la base de données, au moyen de déclencheurs.

The application must handle concurrent queries and updates made by multiple customers, using transactions.

L'application doit gérer les requêtes et mises à jour concurrentes faites par plusieurs utilisateurs, en utilisant des transactions.

I will test your `BankManagerImpl` class with a Java program I developed. The program will first run single-user tests. Then it will run multi-users tests: each user/customer will be emulated by a thread that

connects to the database using a dedicated `BankManagerImpl` object. However, you must develop your own test program and include it in you submission.

Je testerai votre classe `BankManagerImpl` en utilisant un programme Java que j'ai développé. Ce programme effectuera d'abord des tests mono-utilisateur. Il effectuera ensuite des tests multi-utilisateurs : chaque utilisateur/client sera simulé par un thread qui se connecte à la base de données au travers d'un objet `BankManagerImpl` propre. Vous devez cependant écrire votre propre programme de test et l'inclure dans votre rendu.

Beware! - *Attention !*

Please **do not alter** the `BankManager` interface, the prototype of the `BankManagerImpl` constructor, or the `Operation` class. If you do so, my test program will not compile.

***Ne pas modifier** svp l'interface `BankManager`, le prototype du constructeur de `BankManagerImpl`, ou la classe `Operation`. Si vous le faites, mon programme de test ne compilera pas.*

Do not specify any database name in your SQL statements, since I will run the tests against my own database. For instance, don't write « `create table 'durand'.'MYTABLE'` etc. »; write « `create table MYTABLE` etc. » instead.

***Ne pas spécifier de nom de base de données** dans vos ordres SQL, car j'exécuterai les tests sur ma propre base. Par exemple, n'écrivez pas « `create table 'durand'.'MYTABLE'` etc. », mais plutôt « `create table MYTABLE` etc. ».*

Do not hard-code anything in the url, user and password strings. I will typically call your constructor with the following values: "jdbc:mysql://localhost:3311/test", "root", "root".

Ne rien fixer en dur dans les chaînes url, user et passwd. J'appellerai typiquement votre constructeur avec les valeurs suivantes : "jdbc:mysql://localhost:3311/test", "root", "root".

Submission - *Remise*

Submit a ZIP archive named LASTNAME1.FirstName1.LASTNAME2.FirstName2.zip. The archive must be submitted once per pair. It must contain (flat, no sub-directory):

- the `BankManagerImpl.java` file
- the `SimpleTest.java` file

Remettre une archive ZIP nommée NOM1.Prenom1.NOM2.Prenom2.zip. L'archive ne doit être déposée qu'une fois par binôme. Elle doit contenir (à plat, sans sous-répertoire) :

- *le fichier `BankManagerImpl.java`*
- *le fichier `SimpleTest.java`*