Lab7 Report

Author: 11910507陈梓涵

Use synchronized keyword

In the Test, it create 100 thread each to store about 10 dollars. The only method restoring dollar and changing the balance is `deposit`. So Add `synchronized` key before the `deposit`.

```
public synchronized void deposit(double money) {
    try {
        double newBalance = balance + money;
        try {
            Thread.sleep( millis: 10);   // Simulati
        }
        catch(InterruptedException ex) {
            ex.printStackTrace();
        }
        balance = newBalance;
    } catch (Exception e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }

}
```

The Result is 1000.0 dollars.

```
C:\Users\dream\.jdks\openjdk-19.0.2
Balance: 1000.0

Process finished with exit code 0
```

Use ReentrantLock

Similarly, use the `ReentrantLock` need us to create a lock.

```
private final ReentrantLock lock = new ReentrantLock();
```

Then Add lock on the deposit method.

```
public void deposit(double money) {
    lock.lock();
    try {
        double newBalance = balance + money;
        try {
            Thread.sleep( millis: 10);    // Sim
        }
        catch(InterruptedException ex) {
            ex.printStackTrace();
        }
        balance = newBalance;
    } catch (Exception e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    } finally {
        lock.unlock();
    }

}
```

The Result is also 1000.0 dollars.