

Practice 7

Deadline: Apr. 12th

We provided `Account.java`, `DepositThread.java` and `Test.java` on Sakai. Please read the code to understand its semantics. Then execute `Test.java` and observe the results. Is this program thread safe? If not, how should we fix it?

In this practice, we'll be fixing `Account.java`, `DepositThread.java` and `Test.java` so that the program works as expected.

Specifically, you should use the following two approaches:

1. Using the `synchronized` keyword
2. Using `ReentrantLock`

Please show us your fixes using these two approaches and demonstrate that your program now outputs the correct result, i.e., `balance` should be 1000.

Optional Task: Which approach is faster? You may use the `JMH` library we introduced in lab 2 to profile and compare these two approaches. A sample profiling run might look like this:

Benchmark	Mode	Cnt	Score	Error	Units
<code>Test.testAccountLock</code>	avgt	3	1897.608	± 228.313	ms/op
<code>Test.testAccountSync</code>	avgt	3	1860.767	± 272.979	ms/op

Evaluation

The practice will be checked by teachers or SAs. What will be tested:

1. That you understand every line of your own code, not just copy from somewhere
2. That your program compiles correctly (javac)
3. Correctness of the program logic
4. That the result is obtained in a reasonable time

Late submissions after the deadline will incur a 20% penalty, meaning that you can only get 80% of this practice's score.