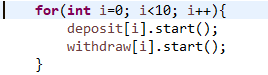
Task 1

Cause it is thread-unsafe. Both deposit and withdraw thread works on the same account and may interrupt the other thread randomly when the other thread is working,which haven’t ensure the consistency of account amount.///two critical section work at the same time

Task2

there will be 10 runs, in every run, one pair of deposit and withdraw threads will start to execution (from the first account to last account one by one)// but thread A won’t wait the thread B(the one start before it) end. While thread B is running, thread A will start.

For every thread, its lifetime is from \*.start() to their run() method end.

And their consistency is not preserved as TASK1 explained.

Task5

The advantage of method level：easy to compile，just add prefix

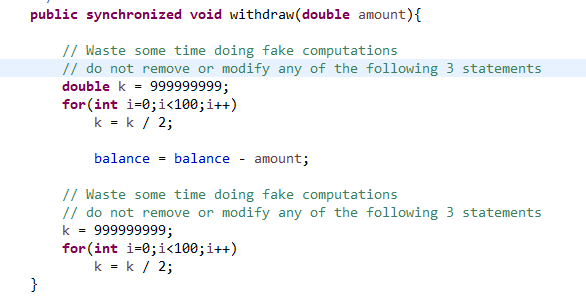
The advantage of block level:

More flexible.

It can help you determine when you want to allow other threads to interrupt while this thread is still working, with multiple thread, it can save time.

Take task3 and task 4 for example.

For task3, if you deposit/withdraw. You have to wait the whole deposit method end. which waste 2 times of fake computation



For task4,after we ensure that balance have been changed, we can unlock the thread, which waste 1 time of fake computation

