

Цель задания

Научиться работать с synchronized блоками и избегать ситуации deadlock.

Условие задания

Проект **Transactions** в директории модуля **Multithreading**.

В банке (класс Bank) есть счета (класс Account) с двумя полями — money и accNumber.

Все счета хранятся внутри банка. Множество клиентов банка могут одновременно переводить деньги между счетами и запрашивать баланс по своему счёту. Всё происходит в highly concurrent (многопоточной) среде.

При этом транзакции на суммы > 50000 отправляются на проверку в службу безопасности. Можно считать, что таких транзакций не более 5% от всех. За проверку отвечает отдельный и уже реализованный метод Bank.isFraud().

Служба безопасности не может обрабатывать более одной транзакции одновременно. Проверка занимает 1000 мс.

Если служба безопасности обнаружила мошенничество, необходимо заблокировать оба счёта, то есть запретить любые изменения остатков в дальнейшем.

Что нужно сделать

1. Создайте метод transfer() класса Bank, который переводит деньги с одного счёта на другой. Если сумма транзакции > 50000 — транзакция отправляется на проверку службе безопасности: вызывается метод isFraud(). Если возвращается true, то счета блокируются (как – на ваше усмотрение).
2. Создайте метод getBalance класса Bank, который возвращает остаток на счёте по переданной строке номера аккаунта.

Рекомендации

- Для решения задачи вы можете дорабатывать классы Account и Bank как угодно.
- Дополнительно создайте тест (или набор тестов) для эмуляции реальной работы этих двух классов и проверки системы.
- Проверяйте сумму на банковских счетах до запуска транзакций и после завершения — сумма в банке не должна измениться.

- Удостоверьтесь, что ваша программа работает в многопоточном режиме. Для этого можете использовать утилиту [visualVM](#).