

Task 1. Parallel algorithm and shared state

Перечисление средств

У Вас есть класс Bank с методом transfer() для перевода денег с одного счета на другой в пределах банка.

```
public class Bank {  
    public void transfer(Account from, Account to,  
int amount) {  
        atomic {  
            from.withdraw(amount);  
            to.deposit(amount);  
        }  
    }  
}
```

К сожалению, в Java отсутствует конструкция типа atomic для выполнения транзакций.

Напишите свою реализация тела метода transfer(), которая могла бы работать в многопоточной среде.

Во время перевода денег со счета на счет, данные счета должны блокироваться. Подумайте, как при этом избежать deadlock-ов.

Выполните тестирование метода transfer(). Для этого:

- 1 Создайте несколько десятков (сотен) счетов и положите на них случайное количество денег
- 2 Подсчитайте сколько денег есть всего в банке (сумма денег на всех счетах)
- 3 Запустите в нескольких тысячах потоков одновременные перевод случайных сумм денег со случайного счета на случайный счет (сумма на счету не может быть отрицательной)

Дождитесь окончания перевод и подсчитайте сколько денег есть всего в банке (сумма денег на всех счетах). Сумма денег в банке до переводов и после, должны совпадать.

Task 2. Producer-consumer task

Завдання. Реалізувати потокобезпечний кільцевий буфер та виконання наступних вимог:

- 1) П'ять потоків генерують рядки в кільцевий буфер. Рядок має формат: «Потік № ... згенерував повідомлення ...»
- 2) Інші два потоки перекладають рядки з першого кільцевого у другий кільцевий буфер. У другий кільцевий буфер записується рядок наступного формату: «Потік № ... переклав повідомлення ...».
- 3) Основний потік програми вичитує та роздруковує 100 повідомлень із другого буфера.

Потоки, описані у вимогах 1)-2) повинні бути потоками-демонами, інакше програма не завершить своє виконання.

Кільцевий буфер – це зв'язний список, який замкнений у кільце. Кільцевий буфер зберігає значення за допомогою двох індексів: головного та кінцевого. При занесенні даних в буфер збільшується кінцевий індекс. При видобуванні даних із буфера збільшується головний індекс. При виході за границю списку індекси автоматично корегуються, переходячи до першого елементу списку. Ситуація, коли головний елемент співпадає із кінцевим, інтерпретується як порожній буфер.

При роботі з кільцевим буфером врахувати наступне:

- 1) якщо кільцевий буфер порожній, то потік, що видобуває дані, повинен зачекати, поки у буфері з'являться дані;
- 2) якщо кільцевий буфер повний, то потік, що додає дані, повинен зачекати, поки у буфері не з'явиться місце.