

## № 7 Многопоточные приложения

### Задание

1. Разработать многопоточное консольное приложение. **Можно использовать `synchronized`.**

**Ремонт дороги.** В Минске на улице Белорусской идет ремонт дороги и работает только одна полоса. В обоих направлениях собралось много машин. Обеспечьте безопасное движение машин (информируйте о прибытии машины и направлении движения, ожидании, проезде и о количестве машин на каждой стороне). Режим работы такой: проезжают три (или менее) с одной стороны, затем три соответственно с другой и т.д..

**Студенческая столовая.** Доступно  $n$  столов (за столом может сидеть только один человек). Если все места заняты, то студент ожидает пока не освободится место.

2. Используя возможности, предоставляемые пакетом `java.util.concurrent` (**не использовать `synchronized`**) вспомогательные классы управления потоками

**Прокат лыж.** В лыжном прокате имеется  $n$  пар гоных лыж (лыж на всех клиентов не хватит). Работник проката может обслуживать только одного клиента, остальные должны ждать своей очереди. Если в текущий момент в прокате нет лыж клиент может ждать или уйти, если превышено время ожидания. Пенсионеры обслуживаются вне очереди. Промоделировать работу проката для  $m$  клиентов ( $m > n$ ).

**Тендер.** На тендер выставляется несколько заявок на закупки. Участники тендера делают заявки. Заявку можно корректировать в сторону уменьшения несколько раз за торги. Тендер определяет победителя в случае если поступило не менее двух предложений (побеждает тот, кто предложил наименьшую цену).

**Общее пояснение:** Все объекты, желающие получить доступ к ресурсу, должны быть отдельными потоками. Написать демонстрацию. Все выполняемые действия и результаты должны быть выведены в лог (консоль).

*P.S. Для оценки 4-5 необходимо решить одну задачу из п.1 и одну из п.2 на выбор.*

### Вопросы:

1. Каким образом можно создать поток?
2. Можно ли поток запустить дважды?
3. Поясните `run` и `start` методы.
4. Что такое монитор?
5. Как работают методы `wait` и `notify/notifyAll`?
6. Чем отличается работа метода `wait` с параметром и без параметра?

7. Как работает метод `Thread.yield()`? Чем отличаются методы `Thread.sleep()` и `Thread.yield()`?
8. Как работает метод `Thread.join()`?
9. Что такое `dead lock`?
10. Что значит приоритет потока?
11. Что такое потоки - демоны в Java?
12. Что значит усыпить поток?
13. В каких состояниях может быть поток в Java? Как вообще работает поток?
14. Можем ли мы остановить поток? В каких случаях?
15. Что такое синхронизация?
16. Различия между `CyclicBarrier` и `CountDownLatch`?
17. Как работает `Semaphor`?