

№ 5 Многопоточные приложения

Задание

Общее задание: Использовать возможности, предоставляемые пакетом `java.util.concurrent`. Все объекты, желающие получить доступ к ресурсу, должны быть отдельными потоками. Написать демонстрацию, решающую задачу. Все выполняемые действия, состояния и результаты должны быть выведены в лог (консоль).

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

Решить одну задачу на выбор. Преподаватель определяет вариант. (можно больше одной).

Вход в университет. В БГТУ есть две двери, через которые студенты могут входить и выходить. По обоим концам дверей собралось много студентов. Обеспечить безопасное движение в обоих направлениях. Студента можно перенаправить из одной двери в другую при превышении заданного времени ожидания.

Душ. В хостеле есть один общий душ. Одновременно в нем могут мыться m человек. Но есть ограничение: мужчины не могут мыться одновременно с женщинами. Утром все жильцы принимают душ и время у каждого разное. Организовать этот процесс оптимальным образом. Приоритеты у всех равны.

Прокат лыж. В лыжном прокате имеется n пар гонных 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 ($m \gg n$) и t работников проката ($t \ll m$).

CallCenter. В организации работает n операторов. Оператор может обслуживать только одного клиента, остальные должны ждать своей очереди. Клиент может положить трубку и перезвонить еще раз через некоторое время.

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

Решить одну задачу на выбор. Преподаватель определяет вариант. (можно больше одной).

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. Если очереди нет, то машины проезжают по мере прибытия.

Аэропорт. Посадка/высадка пассажиров может осуществляться через конечное число терминалов и конечное число трапов. Самолеты бывают разной вместимости. Организовать функционирование аэропорта, если m – вместимость самолета, t – количество терминалов < 3 , r – количество трапов (к самолету подается один трап, пассажиры проходят через него последовательно), s – количество самолетов.

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

Автостоянка. Есть две автостоянки на которых доступно n и m машиномест соответственно. На одном месте может находиться только один автомобиль. Если все места заняты, то автомобиль не станет ждать больше определенного времени и уедет на другую стоянку (по кругу). Организовать парковку k автомашин.

На 9-10 надо решить не менее двух задач из каждой категории.

Вопросы:

1. Каким образом можно создать поток?
2. В каких состояниях может пребывать поток?
3. Можно ли поток запустить дважды?
4. Поясните для чего используются `run` и `start` методы.
5. Чем отличаются методы `interrupt`, `interrupted`, `isInterrupted`?
6. Что такое монитор объекта? Как работают методы `wait` и `notify/notifyAll`?
7. Чем отличается работа метода `wait` с параметром и без параметра?
8. Как работает метод `Thread.yield()`? Чем отличаются методы `Thread.sleep()` и `Thread.yield()`?
9. Как работает метод `Thread.join()`?
10. Что такое `dead lock`?
11. Что значит приоритет потока?
12. Что такое потоки - демоны в Java?
13. Что значит усыпить поток?
14. Что такое `ThreadGroup` и зачем он нужен?
15. В каких состояниях может быть поток в Java? Как вообще работает поток?
16. Можем ли мы остановить поток? В каких случаях?
17. Что такое синхронизация?
18. Что такое «атомарные типы» в Java?
19. В чем разница между блокирующими и неблокирующими очередями?
20. Различия между `CyclicBarrier` и `CountDownLatch`?
21. Как работает `Semaphor`?

22.Что такое Executor?

23.Что такое ExecutorService?