

№ 7 Многопоточные приложения

Задание

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

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

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.

Процесс записи информации в файл. Организуйте процесс записи в файл тремя конкурирующими потоками.

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

Ресторан. Доступно n столов. За столом может сидеть два человека. Если все места заняты, то посетитель ожидает пока не освободится место.

Склад. Машины заезжают на склад для разгрузки/загрузки контейнеров. Число контейнеров, находящихся в текущий момент на складе и в машине, должно быть неотрицательным. Причем количество контейнеров на складе - N большое значение, но не превышающее вместимость склада V . На складе работает M сотрудников. Один сотрудник может обслуживать только одну машину. Машина имеет вместимость G . Машина может загружаться (если она пуста), разгружаться (если есть контейнеры) или выполнять последовательно оба действия разгрузка-загрузка.

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).

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

На 9-10 надо решить все задачи.

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 Semaphore?