

Java: Производитель-потребитель

CarParkControl - condition synchronization

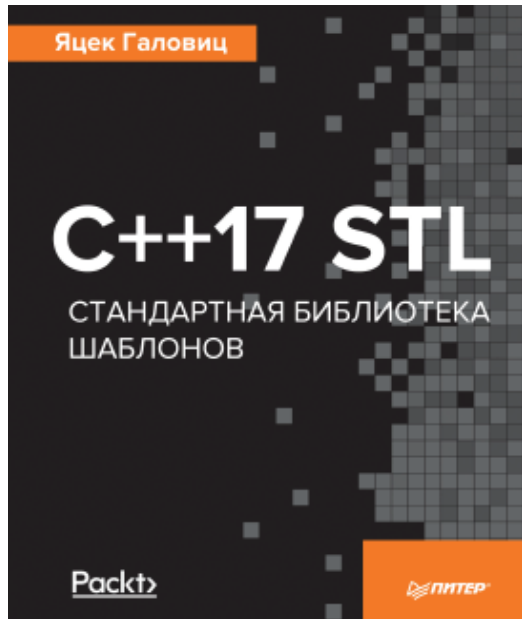
```
class CarParkControl {
    protected int spaces;
    protected int capacity;

    CarParkControl(int n)
        {capacity = spaces = n;}

    synchronized void arrive() throws InterruptedException {
        while (spaces==0) wait();           block if full
        --spaces;
        notifyAll();
    }

    synchronized void depart() throws InterruptedException {
        while (spaces==capacity) wait();    block if empty
        ++spaces;
        notifyAll();
    }
}
```

*Is it safe to use **notify()**
rather than **notifyAll()**?*



9 Параллелизм и конкурентность

- Синхронизация конкурентного использования `std::cout`
- Реализация идиомы «производитель/потребитель» с использованием `std::condition_variable`

Работа над ошибками

Самостоятельная работа, Неделя 10

Задание 2. Я.Галовиц. Реализовать, отладить, протестировать, отчет на почту

Задание 3. Реализовать, отладить, протестировать, отчет на почту. Аргументированно ответить на вопрос