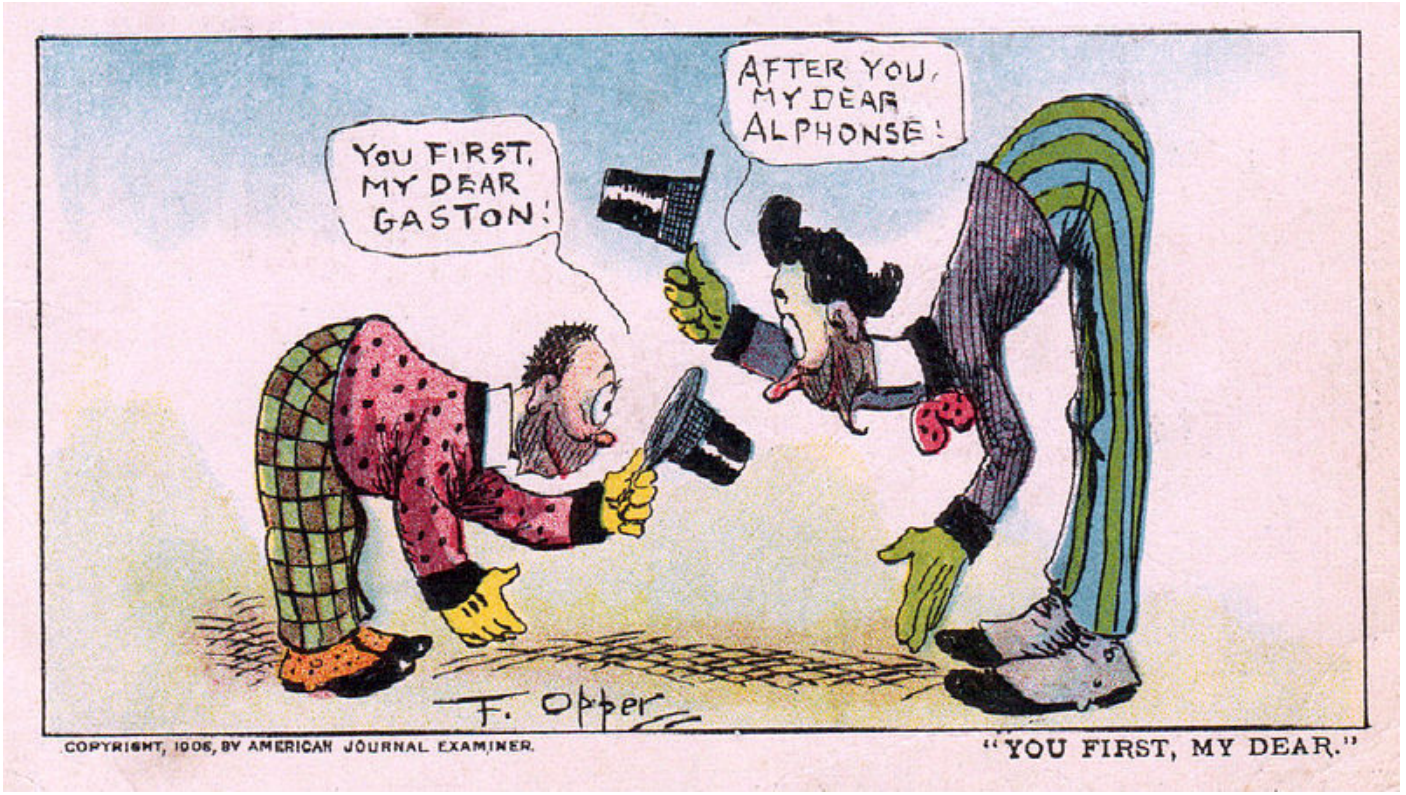
Параллельное программирование: Лекция/Практика 4

Учебный курс

Кензин Максим Юрьевич
Кафедра Информационных технологий
ИМИТ ИГУ / Институт динамики
систем и теории управления СО РАН

ИМИТ ИГУ, 2022

Пример Deadlock.



Пример Deadlock.

1 Вывести блокирующий метод из-под синхронизации (полу-решение)

```
public void bow(Friend bower) {
    synchronized (this)
                {System.out.format("%s: %s" + "  has bowed to me!%n", this.name, bower.getName());}
    bower.bowBack(this);
}
```

Пример Deadlock.

1 Вывести блокирующий метод из-под синхронизации (полу-решение)

```
public void bow(Friend bower) {
    synchronized (this)
                    {System.out.format("%s: %s" + "  has bowed to me!%n", this.name, bower.getName());}
    bower.bowBack(this);
}
```

2 Добавить сверху глобальный лок / транзакцию

```
private static final Object globalLock = new Object();                    ⟵ Bo Friend / B Main

public void bow(Friend bower) {
    synchronized (globalLock){
        System.out.format("%s: %s" + "  has bowed to me!%n", this.name, bower.getName());
        bower.bowBack(this);
    }
}
```

Пример Deadlock.

1 Вывести блокирующий метод из-под синхронизации (полу-решение)

```
public void bow(Friend bower) {
    synchronized (this)
                    {System.out.format("%s: %s" + "  has bowed to me!%n", this.name, bower.getName());}
    bower.bowBack(this);
}
```

2 Добавить сверху глобальный лок / транзакцию

```
private static final Object globalLock = new Object();                    ← Bo Friend / B Main

public void bow(Friend bower) {
    synchronized (globalLock){
        System.out.format("%s: %s" + "  has bowed to me!%n", this.name, bower.getName());
        bower.bowBack(this);
    }
}
```

Или

```
public void bow(Friend bower, Object globalLock) {
    synchronized (globalLock){
        System.out.format("%s: %s" + "  has bowed to me!%n", this.name, bower.getName());
        bower.bowBack(this, globalLock);
    }
}
```

Пример Deadlock.

1 Вывести блокирующий метод из-под синхронизации (полу-решение)

```
public void bow(Friend bower) {
    synchronized (this)
                {System.out.format("%s: %s" + "  has bowed to me!%n", this.name, bower.getName());}
    bower.bowBack(this);
}
```

2 Добавить сверху глобальный лок / транзакцию

```
private static final Object globalLock = new Object();                    ← Bo Friend / B Main

public void bow(Friend bower) {
    synchronized (globalLock){
        System.out.format("%s: %s" + "  has bowed to me!%n", this.name, bower.getName());
        bower.bowBack(this);
    }
}
```

Или

```
public void bow(Friend bower, Object globalLock) {
    synchronized (globalLock){
        System.out.format("%s: %s" + "  has bowed to me!%n", this.name, bower.getName());
        bower.bowBack(this, globalLock);
    }
}
```

3 Сравнение неизменных характеристик объектов для определения порядка выполнения

System.identityHashCode(this) ? System.identityHashCode(this):

```
sync (this){sync (bower) {bow()}}                                    sync (bower){sync (this) {bow()}}
```

Оповещение потоков* #1

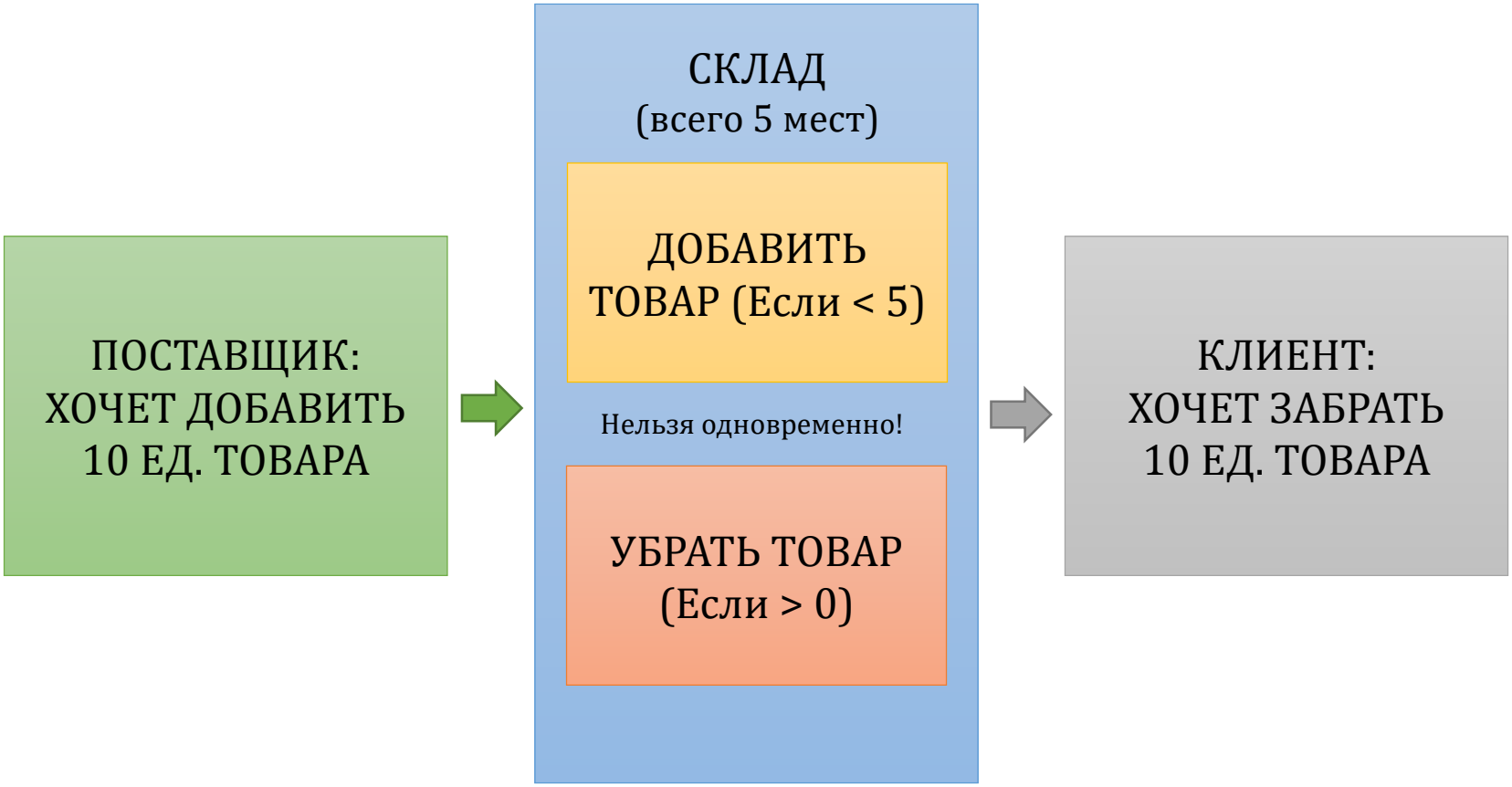
- **wait()/wait(long t)**: освобождает монитор и переводит вызывающий поток в состояние ожидания до тех пор, пока другой поток не вызовет метод notify() в том же объекте, где был вызван wait();

- **notify():** продолжает работу одного потока (случайного), у которого ранее в этом же объекте был вызван метод wait();

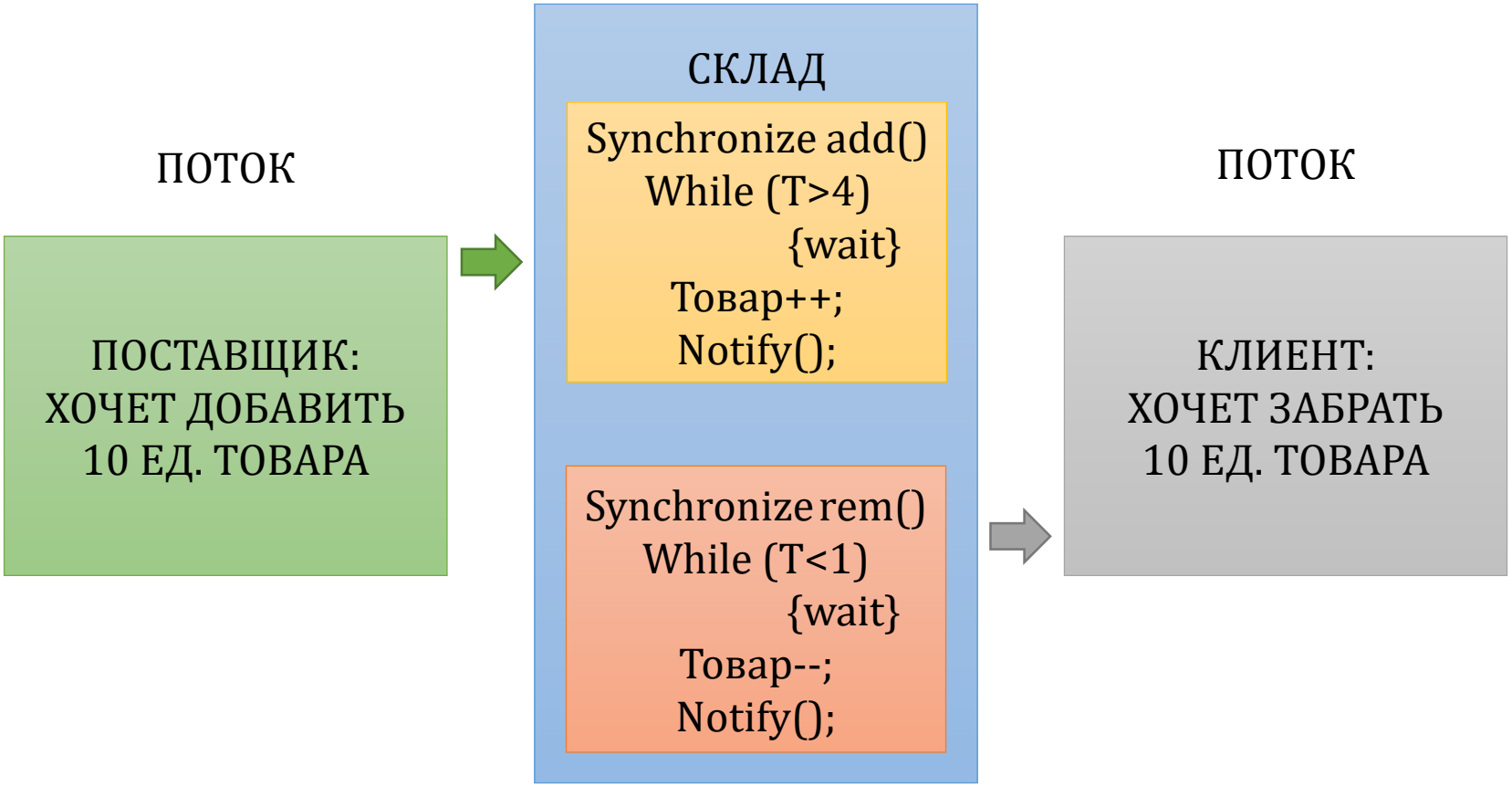
- **notifyAll():** возобновляет работу всех потоков, у которых ранее был вызван метод wait() в рамках общего объекта.

* Вызывать **только** из синхронизированного блока/метода.

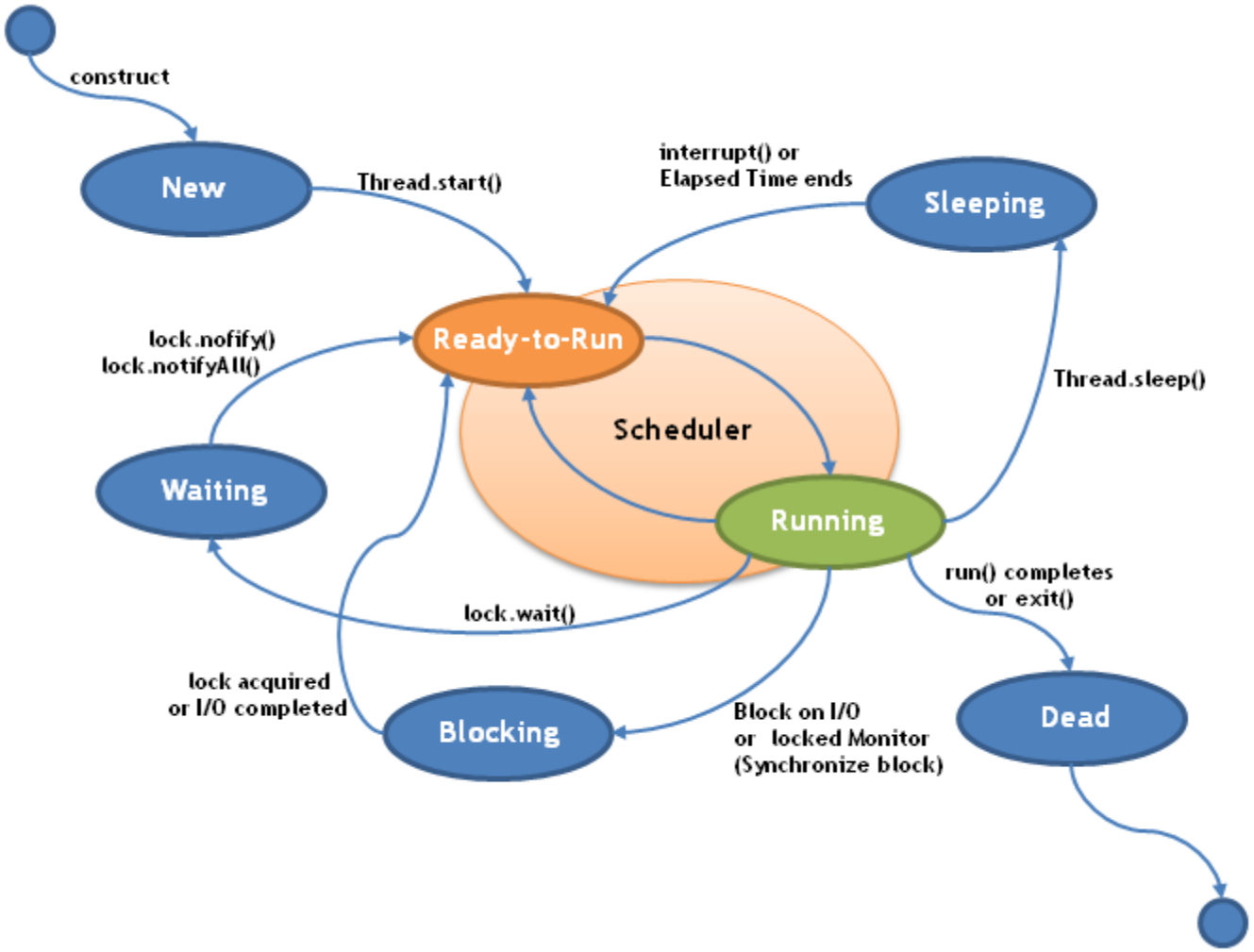
Оповещение потоков #1



Оповещение потоков #1



Жизненный цикл Java Thread



Доп замечание по потокам

implements Runnable:

- Можно наследовать некоторый другой класс;
- При необходимости можно просто запустить .run() без создания нового потока.

extends Thread:

- Можно переопределять основное поведение Thread;
- Поток Thread нельзя запустить через .start() дважды, так как после завершения он переходит в состояние Dead.