

Java

Занятие 6. Многопоточность.

Для чего нужна многопоточность?

Для чего нужна многопоточность:

- оптимальное использование процессорного времени
- многозадачность
- асинхронная обработка
- повышение производительности

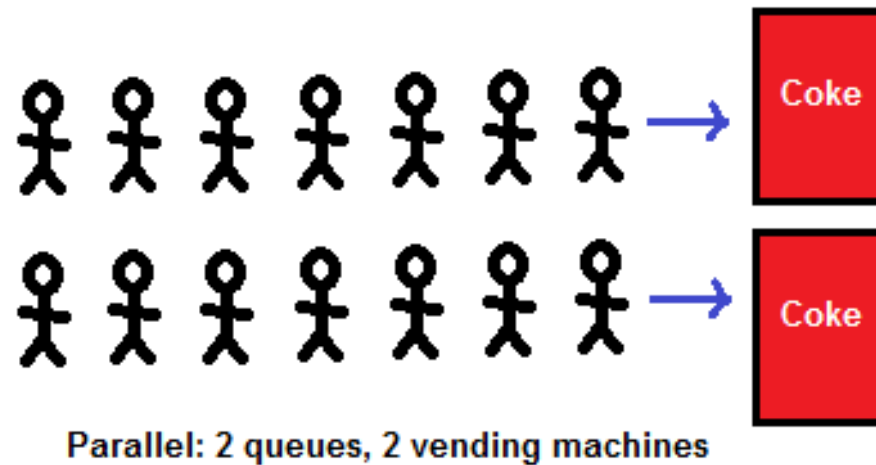
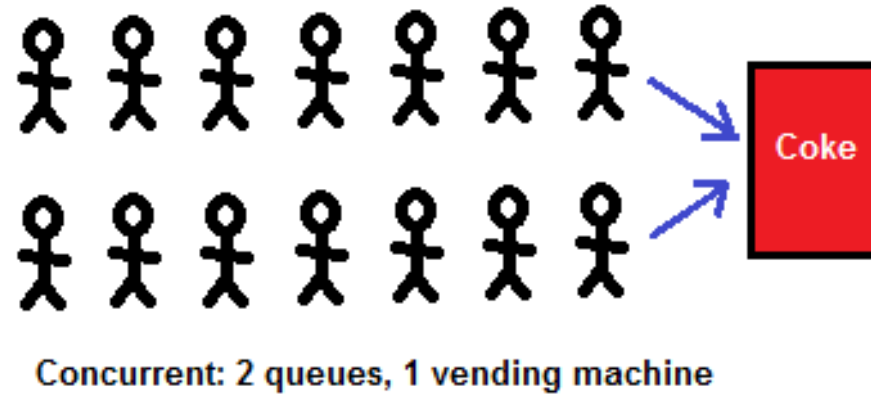
Multithreaded programming



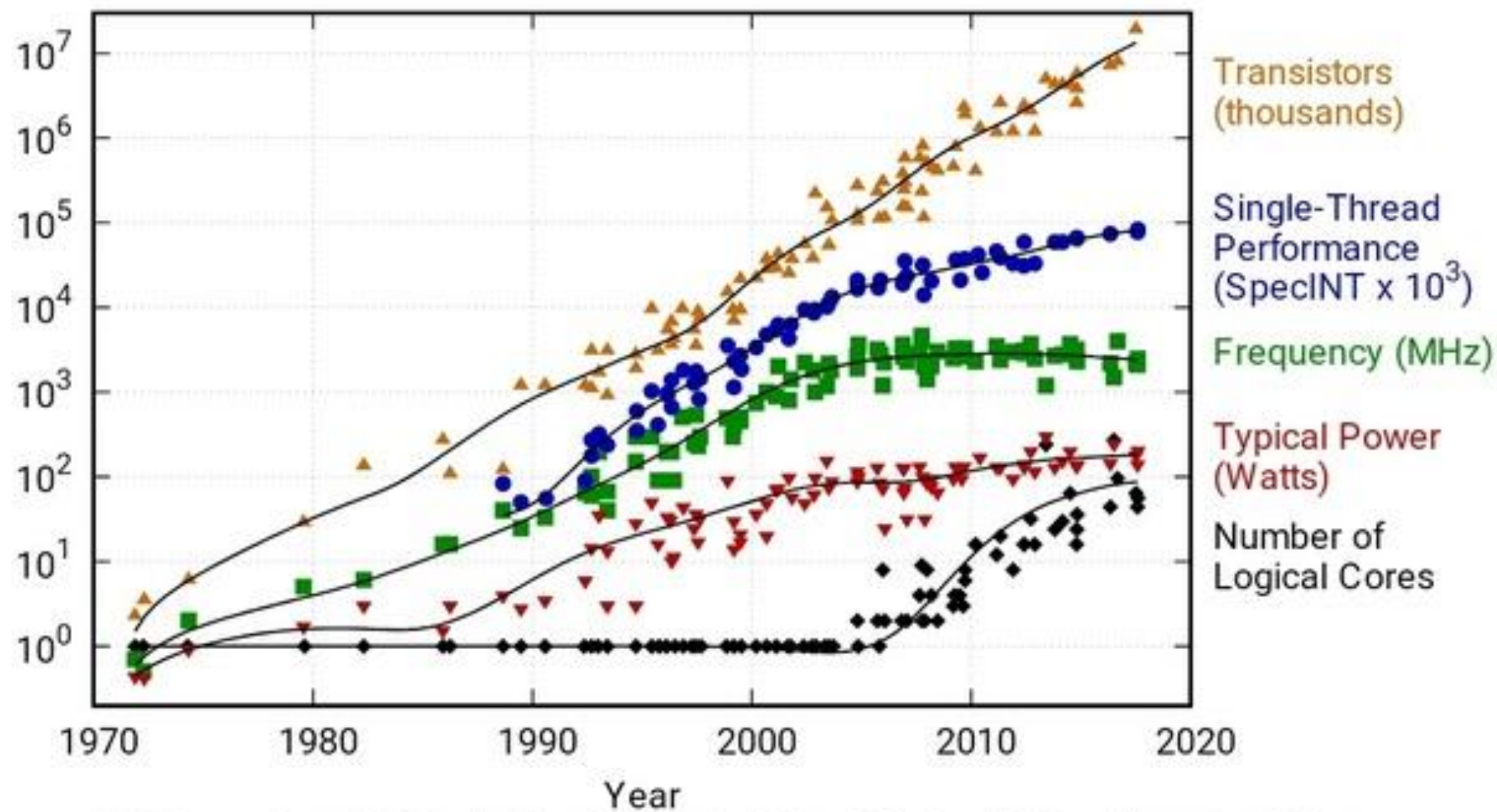
Следует различать процессы и потоки



Следует различать Concurrent и Parallel



42 Years of Microprocessor Trend Data



Original data up to the year 2010 collected and plotted by M. Horowitz, F. Labonte, O. Shacham, K. Olukotun, L. Hammond, and C. Batten
New plot and data collected for 2010-2017 by K. Rupp

Способы создать поток: Thread

```
public class MyThread extends Thread {  
    @Override  
    public void run() {  
        for (int i = 0; i < 10; i++){  
            System.out.println("MyThread extends Thread: "+i);  
        }  
    }  
}
```

.....

```
MyThread t1 = new MyThread();  
t1.start();
```


Способы создать поток: Runnable

```
public class MyRunnable implements Runnable {  
    @Override  
    public void run() {  
        for (int i = 0; i < 10; i++){  
            System.out.println("MyRunnable implements Runnable: "+i);  
        }  
    }  
}
```

.....

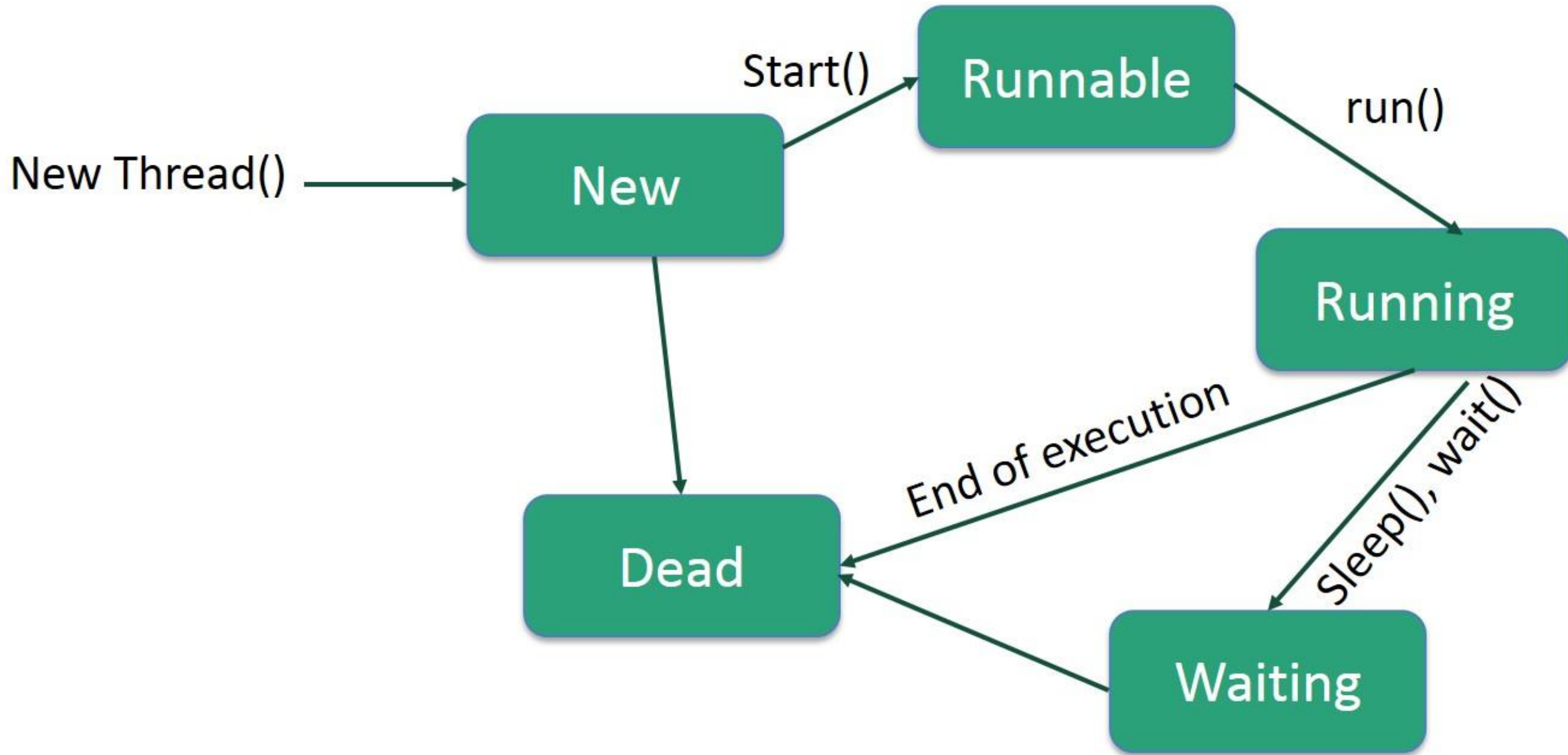
```
Thread t2 = new Thread(new MyRunnable());  
t2.start();
```

Способы создать поток: Timer

```
Timer timer = new Timer();
timer.schedule(new TimerTask() {
    @Override
    public void run() {
        System.out.println("timer thread");
    }
}, 1000);
```

Методы для работы с потоками

- `start()`
- `sleep()`
- `yield()`
- `setDaemon()`
- `join()`



Как остановить поток?



Как остановить поток?



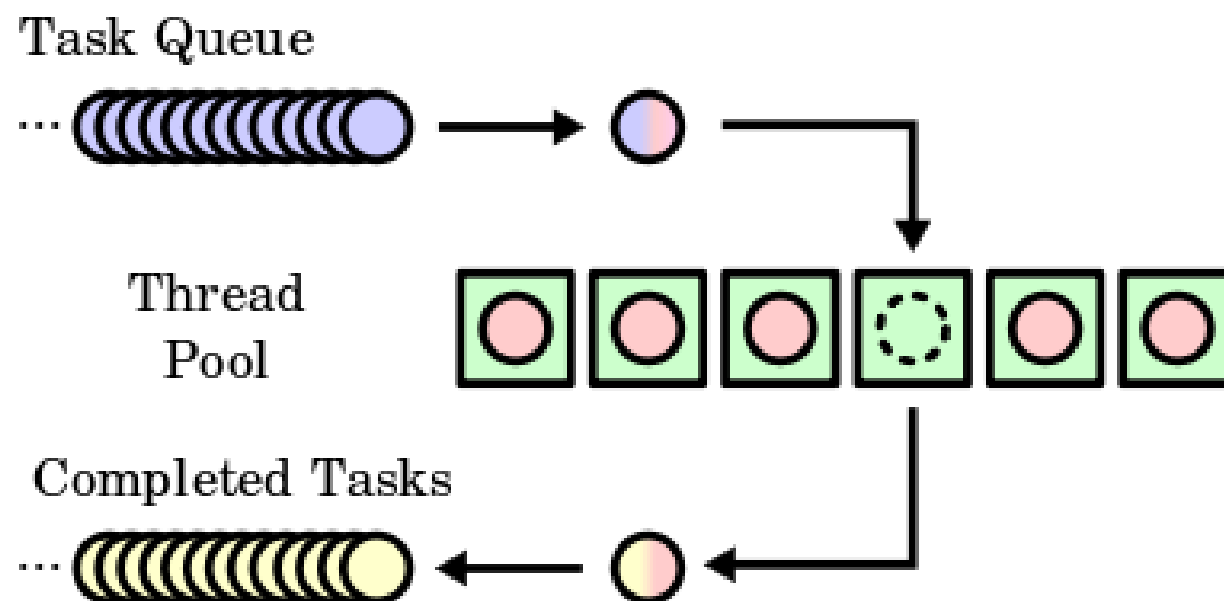
Как остановить поток?



Как остановить поток?



Thread pool



Thread pool

```
ThreadPool threadPool = new ThreadPool(100);
```

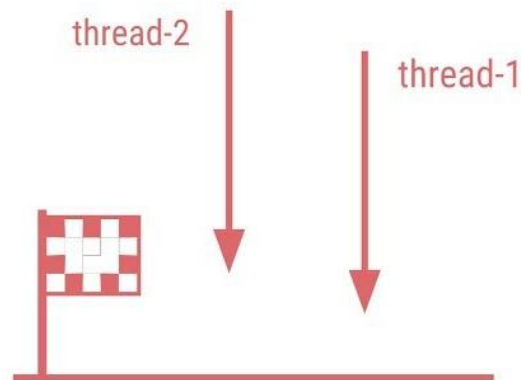
```
....
```

```
int cores = Runtime.getRuntime().availableProcessors();
```

Counter

Race condition

Race Conditions



Атомарные операции

- Чтение/запись примитива (кроме long и double)
- Чтение/запись ссылок
- Специальные атомарные операции

Compare and swap (CAS)

```
1 public class CasCounter {
2     private SimulatedCAS value;
3
4     public int getValue() {
5         return value.getValue();
6     }
7
8     public int increment() {
9         int oldValue = value.getValue();
10        while (value.compareAndSwap(oldValue, oldValue + 1) == false)
11            oldValue = value.getValue(); // retrieve the changed value
12        return oldValue + 1;
13    }
14 }
```

Критическая секция



Критическая секция: Lock

```
ReentrantLock lock = new ReentrantLock();  
lock.lock();
```

```
try {  
    //  
}finally {  
    lock.unlock();  
}
```


Критическая секция: synchronize

```
public class Main {  
    private Object obj = new Object();  
    public void doSomething() {  
        //...какая-то логика, доступная для всех потоков  
        synchronized (obj) {  
            //логика, которая одновременно доступна только для одного  
потока  
        }  
    }  
}
```

Критическая секция: synchronize

```
public class Main {  
    private Object obj = new Object();  
    public void doSomething() throws InterruptedException {  
        //...какая-то логика, доступная для всех потоков  
  
        //логика, которая одновременно доступна только для одного потока:  
  
        /*до тех пор, пока мьютекс объекта занят -  
        любой другой поток (кроме того, который его захватил), спит*/  
        while (obj.getMutex().isBusy()) {  
            Thread.sleep(1);  
        }  
  
        //позначить мьютекс объекта как занятый  
        obj.getMutex().isBusy() = true;  
  
        /*выполнить важную работу, при которой доступ к объекту  
        должен быть только у одного потока*/  
        obj.someImportantMethod();  
  
        //освободить мьютекс объекта  
        obj.getMutex().isBusy() = false;  
    }  
}
```

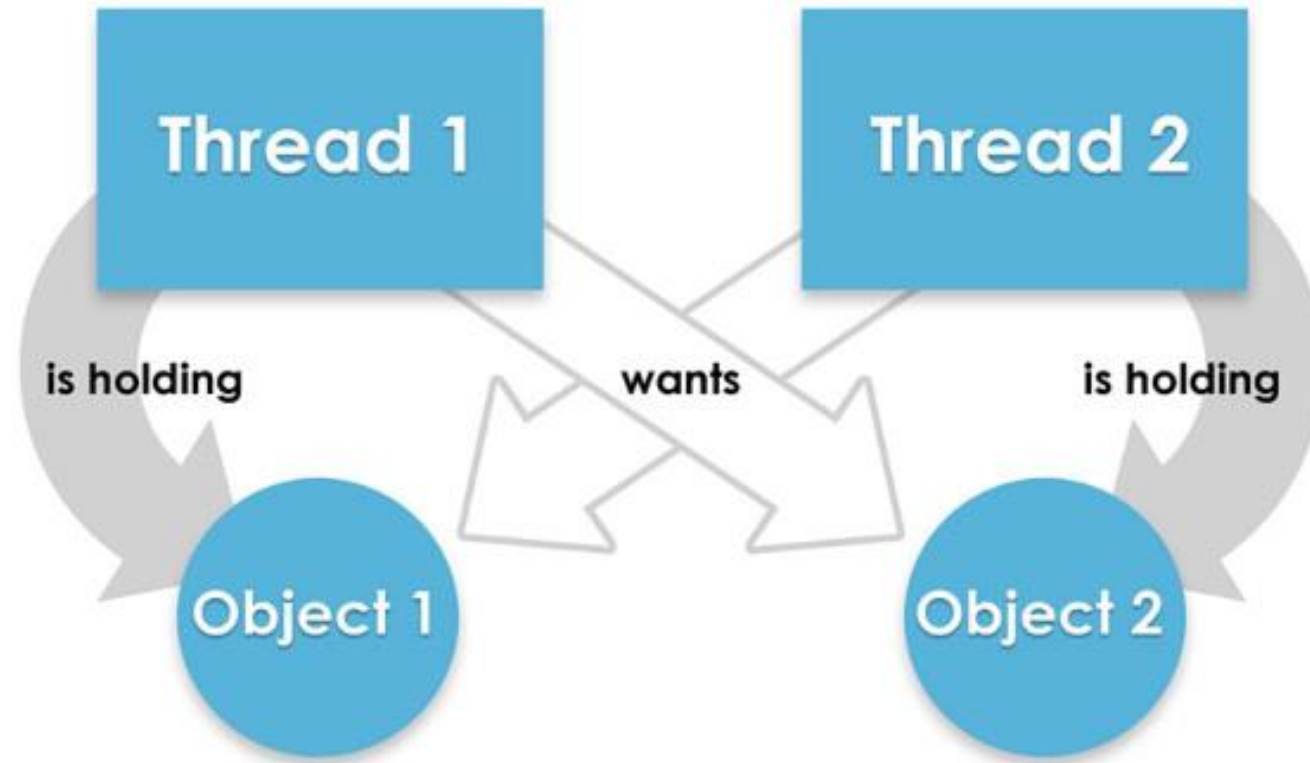
Критическая секция: семафор



Проблемы

- Race condition
- Starvation
- Deadlock

Deadlock



Полезные ссылки

Просто и коротко о потоках:

- <https://javarush.ru/groups/posts/1992-mnogopotochnostjh-v-java-sutjh-pliusih-i-chastihe-lovushki-> - многопоточность в Java: суть, «плюсы» и частые ловушки
- <https://javarush.ru/groups/posts/1993-mnogopotochnostjh-cto-delajut-metodih-klassa-thread-> - многопоточность: что делают методы класса Thread
- <https://javarush.ru/groups/posts/2174-v-chem-raznica-mezhdu-mjhjuteksom-monitorom-i-semaforom-> - в чем разница между мьютексом, монитором и семафором
- <https://javarush.ru/groups/posts/1994-sinkhronizacija-potokov-operator-synchronized-> - синхронизация потоков. Оператор synchronized

Сложно и подробно о потоках:

- <https://habr.com/ru/post/164487/> - многопоточность в Java
- <https://habr.com/ru/post/326146/> - многопоточность на корабliках (интересный материал)
- <https://habr.com/ru/post/326146/> - пул потоков

Вопросы