

Реактивный Хардкор

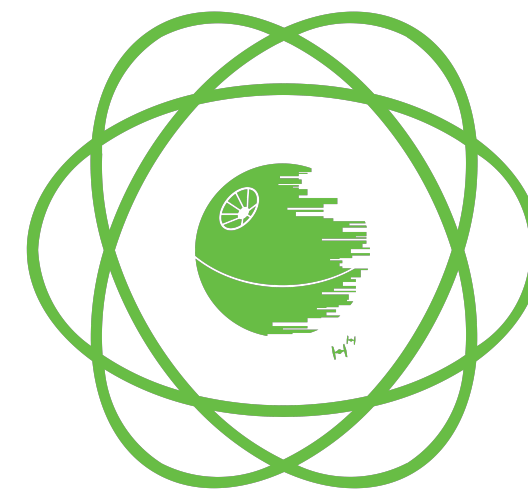
/ by Oleh Dokuka





Oleh Dokuka

netifi



Socket



План

/ Реактивная Спекаая

/ Имплементим хардкор

/ ИТОЖИМ

Для кого?

- / Вы психанули, и реализовываете свою реактивную библиотеку
- / Вам хочется понять как оно работает
- / Вы просто любите хардкора

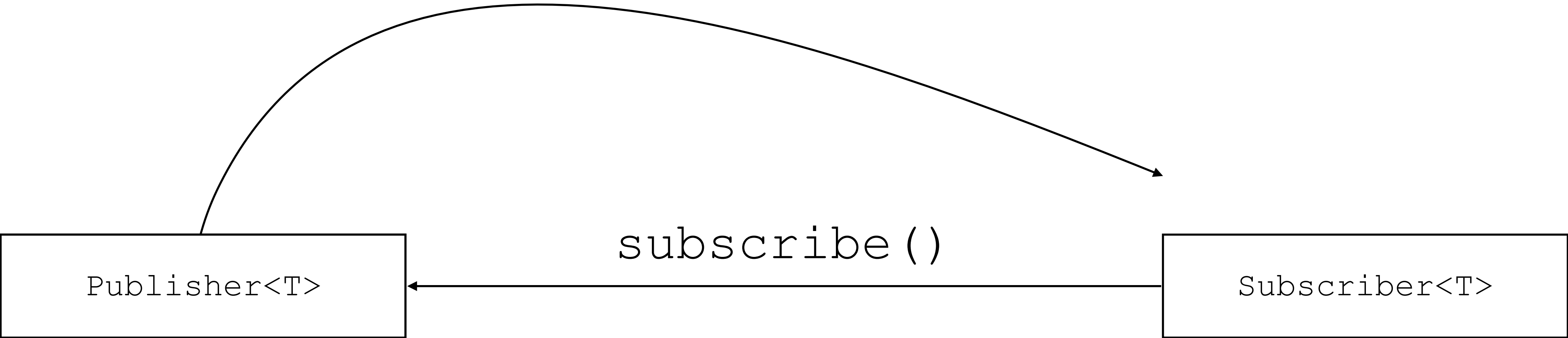
Реактивный Хардкор

/ Oleh Dokuka

Reactive-Streams

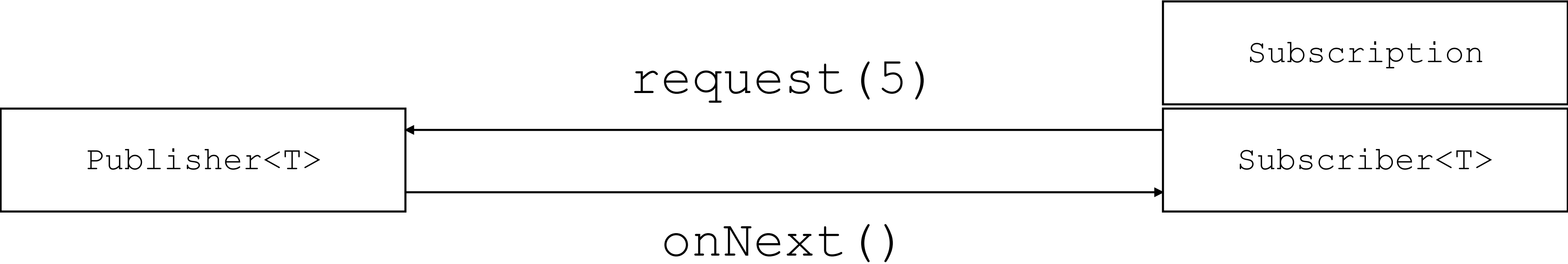
Реактивный Хардкор

/ Oleh Dokuka



Реактивный Хардкор

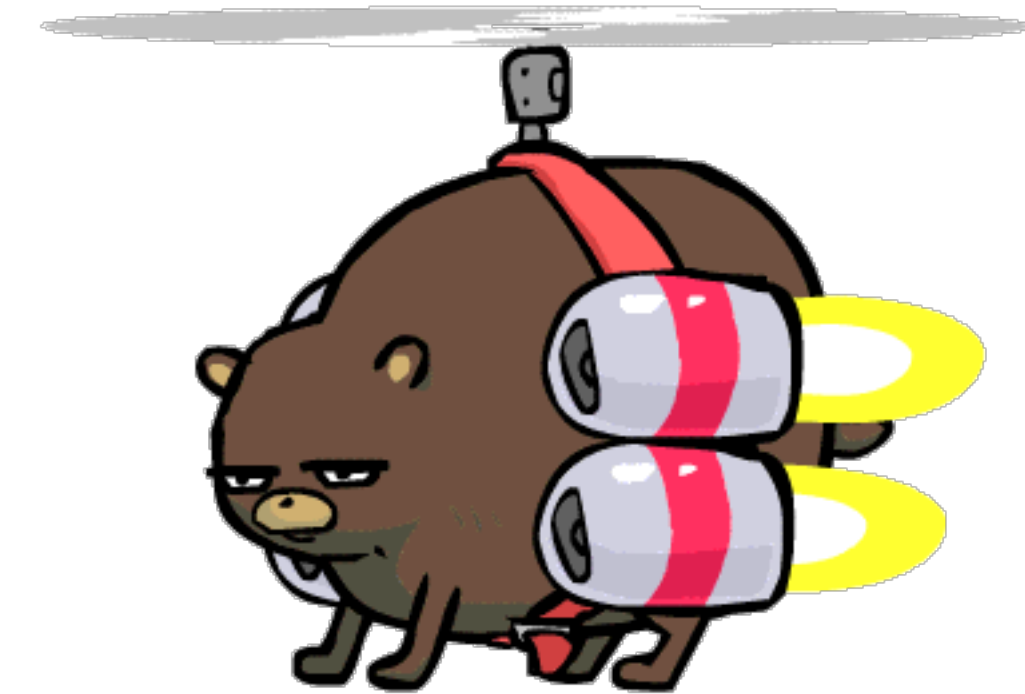
/ Oleh Dokuka



Реактивный Хардкор

/ Oleh Dokuka

```
public class ArrayPublisher<T> implements Publisher<T> {  
    private final T[] source;  
  
    public ArrayPublisher(T[] source) {  
        this.source = source;  
    }  
  
    @Override  
    public void subscribe(Subscriber<? super T> subscriber) {  
        for (int i = 0; i < source.length; i++) {  
            subscriber.onNext(source[i]);  
        }  
    }  
}
```



**Потому
что
спека**

104500

Все по порядку	/	<code>onSubscribe()</code>	1
	/	<code>onNext()</code>	0..N
	/	<code>onError onComplete()</code>	1

СКОЛЬКО
request()
СТОЛЬКО И
onNext()

```
request(5) {  
    onNext(a)  
    onNext(b)  
    onNext(c)  
    onNext(d)  
    onNext(e)  
}
```


He onNext(**null**)

Subscriber-у своему



Не убей себя рекурсией

```
onNext(a) {  
  request(1) {  
    onNext(b) {  
      request(1) {
```



```
}
```

```
}
```

```
}
```

```
}
```

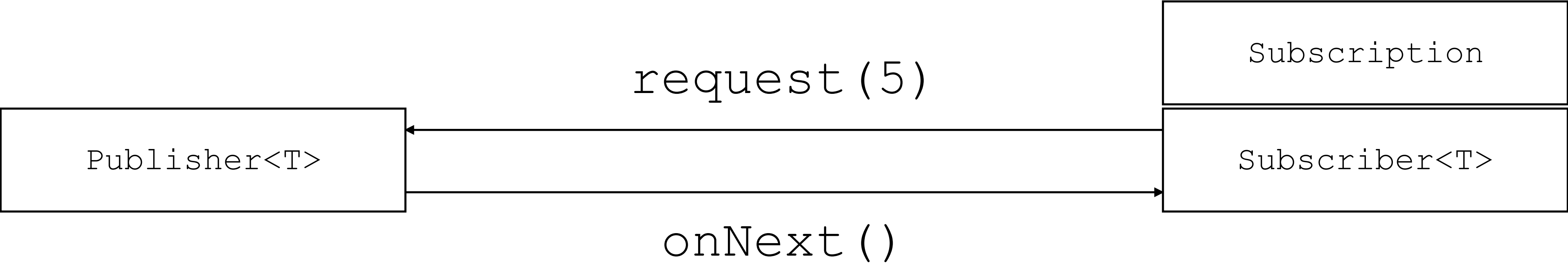
Если `cancel()` значит не надо

Демо 1

по простому

Реактивный Хардкор

/ Oleh Dokuka



ТСК

/ Проверяет все правила из
коробки

ИТОЖИМ

/ Работает

/ Тесты проходят

/ Немного сложно

Многопоточность

ПОТОКО - безопасный

/ onNext()

/ onComplete()

/ onError()

request() - Non-blocking

cancel() - Non-blocking

Демо 2

МНОГОПОТОЧНЕЙ

Реактивный Хардкор

/ Oleh Dokuka

```
AtomicLong requested = new AtomicLong();                                0

@Override
public void request(long n) {
    long initialRequested = requested.getAndAdd(n);                    0

    if (initialRequested > 0) {
        return;
    }

    for (...) {
        ...
    }
    ...

    requested.addAndGet(-sent);
}
```


Реактивный Хардкор

/ Oleh Dokuka

```
AtomicLong requested = new AtomicLong(); 0

@Override
public void request(long n) {
    long initialRequested = requested.getAndAdd(n); 0

    if (initialRequested > 0) {
        return;
    }

    for (...) {
        ...
    }
    ...

    requested.addAndGet(-sent);
}
```

Реактивный Хардкор

/ Oleh Dokuka

```
AtomicLong requested = new AtomicLong();
```

5

```
@Override
```

```
public void request(long n) {  
    long initialRequested = requested.getAndAdd(n);
```

0

```
    if (initialRequested > 0) {  
        return;
```

```
    }
```

```
    for (...) {
```

```
        ...
```

```
    }
```

```
    ...
```

```
    requested.addAndGet(-sent);
```

0

```
}
```

Реактивный Хардкор

/ Oleh Dokuka

```
AtomicLong requested = new AtomicLong();
```

5

```
@Override
```

```
public void request(long n) {  
    long initialRequested = requested.getAndAdd(n);
```

0

```
    if (initialRequested > 0) {  
        return;  
    }
```

```
    for (...) {  
        ...  
    }  
    ...
```

27

```
        requested.addAndGet(-sent);  
    }
```

0

Реактивный Хардкор

/ Oleh Dokuka

```
AtomicLong requested = new AtomicLong();
```

5

```
@Override
```

```
public void request(long n) {  
    long initialRequested = requested.getAndAdd(n);
```

0

```
    if (initialRequested > 0) {  
        return;  
    }
```

```
    for (...) {  
        ...  
    }  
    ...
```

28

```
    requested.addAndGet(-sent);  
}
```

0

Реактивный Хардкор

/ Oleh Dokuka

```
AtomicLong requested = new AtomicLong();
```

5

```
@Override
```

```
public void request(long n) {  
    long initialRequested = requested.getAndAdd(n);
```

0

```
    if (initialRequested > 0) {  
        return;  
    }
```

```
    for (...) {  
        ...  
    }  
    ...
```

29

```
        requested.addAndGet(-sent);  
    }
```

5

Реактивный Хардкор

/ Oleh Dokuka

```
AtomicLong requested = new AtomicLong();
```

```
@Override
```

```
public void request(long n) {  
    long initialRequested = requested.getAndAdd(n);
```

```
    if (initialRequested > 0) {  
        return;  
    }
```

```
    for (...) {  
        ...  
    }  
    ...
```

30

```
        requested.addAndGet(-sent);  
    }
```

5

Реактивный Хардкор

/ Oleh Dokuka

```
AtomicLong requested = new AtomicLong();                                0

@Override
public void request(long n) {
    long initialRequested = requested.getAndAdd(n);                    0

    if (initialRequested > 0) {
        return;
    }

    for (...) {
        ...
    }
    ...

    requested.addAndGet(-sent);
}
```

Реактивный Хардкор

/ Oleh Dokuka

```
AtomicLong requested = new AtomicLong();           5

@Override
public void request(long n) {
    long initialRequested = requested.getAndAdd(n);  0

    if (initialRequested > 0) {
        return;
    }

    for (...) {
        ...
    }
    ...

    requested.addAndGet(-sent);
}
```


Реактивный Хардкор

/ Oleh Dokuka

```
AtomicLong requested = new AtomicLong();
```

5

```
@Override
```

```
public void request(long n) {  
    long initialRequested = requested.getAndAdd(n);
```

0

```
    if (initialRequested > 0) {  
        return;  
    }
```

```
    for (...) {  
        ...  
    }  
    ...
```

33

```
        requested.addAndGet(-sent);  
    }
```

1

Реактивный Хардкор

/ Oleh Dokuka

```
AtomicLong requested = new AtomicLong();
```

5

```
@Override
```

```
public void request(long n) {  
    long initialRequested = requested.getAndAdd(n);
```

0

```
    if (initialRequested > 0) {  
        return;  
    }
```

```
    for (...) {  
        ...  
    }  
    ...
```

34

```
        requested.addAndGet(-sent);  
    }
```

5

Реактивный Хардкор

/ Oleh Dokuka

```
AtomicLong requested = new AtomicLong();
```

10

```
@Override
```

```
public void request(long n) {  
    long initialRequested = requested.getAndAdd(n);
```

5

```
    if (initialRequested > 0) {  
        return;  
    }
```

```
    for (...) {  
        ...  
    }  
    ...
```

35

```
    requested.addAndGet(-sent);  
}
```

5

Реактивный Хардкор

/ Oleh Dokuka

```
AtomicLong requested = new AtomicLong();
```

5

```
@Override
```

```
public void request(long n) {  
    long initialRequested = requested.getAndAdd(n);
```

5

```
    if (initialRequested > 0) {  
        return;  
    }
```

```
    for (...) {  
        ...  
    }  
    ...
```

36

```
        requested.addAndGet(-sent);  
    }
```

5

Реактивный Хардкор

/ Oleh Dokuka

```
AtomicLong requested = new AtomicLong();
```

5

```
@Override
```

```
public void request(long n) {  
    long initialRequested = requested.getAndAdd(n);
```

5

```
    if (initialRequested > 0) {
```

```
        return;
```

```
    }
```

```
    for (...) {
```

```
        ...
```

```
    }
```

```
    ...
```

37

```
    requested.addAndGet(-sent);
```

0

```
}
```


Реактивный Хардкор

/ Oleh Dokuka



ИТОЖИМ

/ Ну, кажись работает

/ Код - ужас

Оптимизируем



Оптимизируем

- / Доступ к переменным
- / Количество атомарных оп-ий
- / Количество созданных объектов
- / Путь выполнения

Попросили
Long.MAX_VALUE -
отсылай сколько хочешь

Демо 3

ОПТИМИЗИРУЕМ

ИТОЖИМ

/ Оптимизация успешна

/ Наш код == Reactor код

Паттерны

/ WIP - работа в прогрессе

/ бесконечный цикл

/ fastPath / slowPath

/ AtomicXXXFieldUpdater

Итог

- / Ничего страшного нету
- / Все для скорости и низкой задержки
- / ТСК - помогает, но кое-что сами
- / Мойте руки после говно-кода



Вопросы и Спасибо!



/OlehDokuka



/OlegDokuka



/oleh.dokuka