

СПб АУ НОЦНТ РАН

Java 02

16.02.2016

Почему так делать плохо?

```
T value;  
  
T get() {  
    if (value == null) {  
        ...  
    }  
}
```

Вот, что ожидалось

```
private static final Object NONE = new Object();  
T value = NONE;  
  
T get() {  
    if (value == NONE) {  
        ...  
    }  
}
```

```
private T value;

T get() {
    if (value == NONE) {
        synchronized (this) {
            if (value == NONE) {
                value = supplier.get();
            }
        }
    }
    return value;
}
```

Volatile

```
private volatile T value;

T get() {
    if (value == NONE) {
        synchronized (this) {
            if (value == NONE) {
                value = supplier.get();
            }
        }
    }
    return value;
}
```

Volatile

```
private volatile T value;

T get() {
    if (value == NONE) {
        synchronized (this) {
            if (value == NONE) {
                value = supplier.get();
                supplier = null;
            }
        }
    }
    return value;
}
```

```
private final AtomicReference<T> reference;

T get() {
    if (reference.get() == NONE) {
        reference.compareAndSet((T) NONE, supplier.get());
    }
    return reference.get();
}
```

Разбор ДЗ про Lazy. Wait-free

```
private static final
    AtomicReferenceFieldUpdater<MyClass, Object> updater =
        AtomicReferenceFieldUpdater.newUpdater(
            MyClass.class, Object.class, "value");
private volatile T value;

T get() {
    if (value == NONE) {
        updater.compareAndSet(this, NONE, supplier.get());
    }
    return value;
}
```

Разбор ДЗ по Lazy. Wait-free. Release

```
private final static
    AtomicReferenceFieldUpdater<LazyD, Object> updater =
        AtomicReferenceFieldUpdater.newUpdater(
            MyClass.class, Object.class, "value");
private volatile T value;

T get() {
    if (value == NONE) {
        Supplier<T> local = supplier;
        if (local != null) {
            if (updater.compareAndSet(this, NONE, local.get()))
                supplier = null;
        }
    }

    return value;
}
```

Что делать?

- ▶ Дорешивать домашку после ревью
- ▶ Добавить меня в свой репозиторий
- ▶ Форкнуть репозиторий
<https://github.com/java-course-au/assignments-2>
- ▶ Сделать pull request в master с исправлениями
- ▶ Поругаться на checkstyle
- ▶ Попробовать отредактировать его (убрать самые строгие правила)
- ▶ Возможно последнее придется откатить

Что делать сегодня

- ▶ Задание в ветке 01-streams
- ▶ Решаем первую часть, тесты проходят
- ▶ Решаем вторую часть, пишем простые тесты
- ▶ Как обычно, pull request