

Flaky tests

О нестабильных тестах
и как с ними бороться

Андрей Солнцев
 **codeborne**

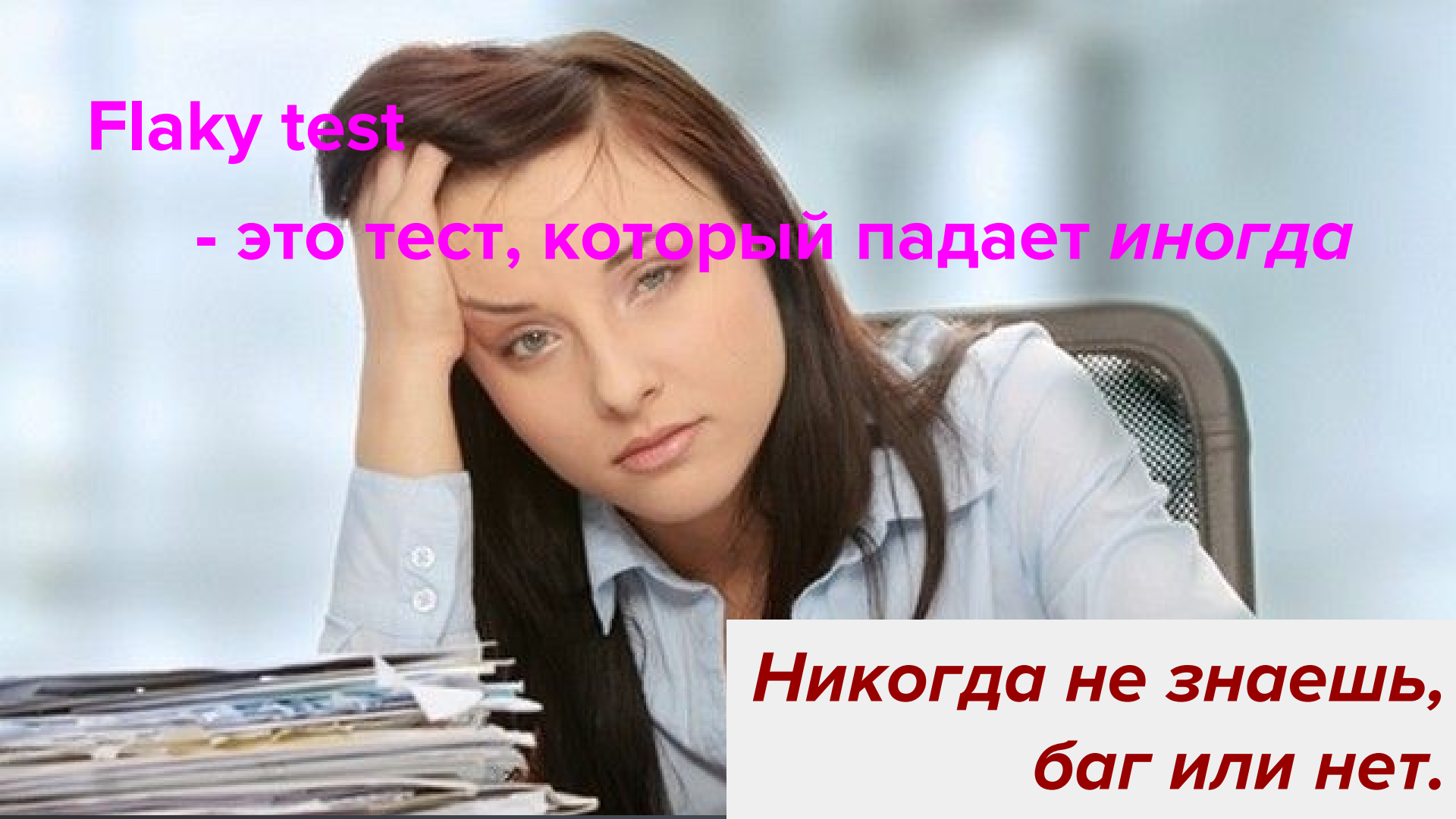
АД
автоматизаторов



Как у вас с
автоматизацией?



СТАБИЛЬНОСТИ
НЕТ!



Flaky test

- это тест, который падает *иногда*

***Никогда не знаешь,
баг или нет.***

Ой, 30 тестов упало.
Надо изучить!

... Впрочем,
П.....У..... !



Ой, 30 тестов упало.
Надо изучить!

... Впрочем,
Перезапущу!



Сколько у вас flaky тестов?

Часто слышу - 30%

Google - 1.5%

<https://testing.googleblog.com/2016/05/flaky-tests-at-google-and-how-we.html>

Codeborne - 0.1%

Даже 1.5% - очень плохо!

● #28 Aug 9, 2017 4:19 PM

Упало 15 из 1000

● #27 Aug 9, 2017 4:07 PM

|

● #26 Aug 9, 2017 12:13 PM

Упало 12 из 1000

● #25 Aug 9, 2017 11:06 AM

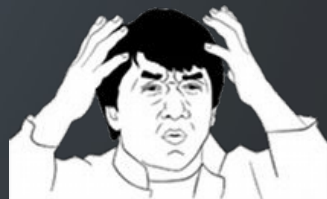
|

● #24 Aug 9, 2017 10:43 AM

Упало 18 из 1000

● #23 Aug 9, 2017 9:55 AM

Нужен ручной контроль!





Build History

[trend](#) —

x

- [#237](#) Apr 20, 2017 5:36 PM
- [#236](#) Apr 20, 2017 4:36 PM
- [#234](#) Apr 20, 2017 4:12 PM
- [#233](#) Apr 20, 2017 3:24 PM
- [#232](#) Apr 20, 2017 3:00 PM
- [#231](#) Apr 20, 2017 2:12 PM
- [#230](#) Apr 20, 2017 2:00 PM
- [#229](#) Apr 20, 2017 1:51 PM
- [#227](#) Apr 20, 2017 12:23 PM
- [#226](#) Apr 20, 2017 12:12 PM
- [#225](#) Apr 20, 2017 11:12 AM

И даже 0.1% - плохо:

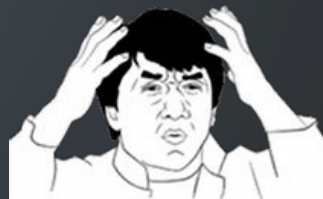
[Test Result](#) (1 failure / +1)

[ui.mobile.MobilePaymentToCorporateSpec.canMakePaymentToCorporateByVatin](#)

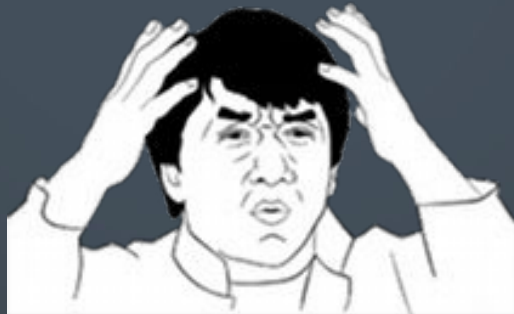
Test Name

[+ ui.card.OtherBankCardSpec.userCanBindOtherBankCard](#)

Ни дня без красного билда!



Индустрия в опасности!



План

1. Моя коллекция нестабильных тестов
2. Причины нестабильности
3. Как с ними бороться?



Моя коллекция

VampInMask.deviantart.com

Пример 1: классика

Этот динамический веб

Какая строка тут может сломаться?

```
driver.navigate().to("https://www.google.com/");  
driver.findElement(By.name("q")).sendKeys("selenide");  
driver.findElement(By.name("btnK")).click();
```

```
assertEquals(9, driver.findElements(  
    By.cssSelector("#ires .g")).size());
```

Какая строчка тут может сломаться?

ЛЮБАЯ!




```
driver.navigate().to("https://www.google.com/");  
driver.findElement(By.name("q")).sendKeys("selenide");  
driver.findElement(By.name("btnK")).click();  
assertEquals(9, driver.findElements(  
    By.cssSelector("#ires .g")).size());
```



Медленный интернет

```
driver.navigate().to("https://www.google.com/");  
driver.findElement(By.name("q")).sendKeys("selenide");  
driver.findElement(By.name("btnK")).click();  
assertEquals(9, driver.findElements(  
    By.cssSelector("#ires .g")).size());
```



Элемент не успел отрисоваться

```
driver.navigate().to("https://www.google.com/");  
driver.findElement(By.name("q")).sendKeys("selenide");
```

```
driver.findElement(By.name("btnK")).click();
```

```
assertEquals(0, driver.findElement(
```

WebDriverException:

**Element <input value="Google Search"
aria-label="Google Search" name="btnK"
type="submit" jsaction="sf.chk"> is not
clickable at point (448, 411). Other
element would receive the click: <div
class="sbqs_c">...</div>**



```
driver.navigate().to("https://www.google.com/");  
driver.findElement(By.name("q")).sendKeys("selenide");  
driver.findElement(By.name("btnK")).click();  
assertEquals(9, driver.findElements(  
    By.cssSelector("#ires .g")).size());
```



Не все строки успели отрисоваться

Причины 90% flaku тестов:

- Ајах запросы: скорость
- Ајах запросы: порядок
- Скорость JS

Есть лекарство!

- ~~Аjax запросы: скорость~~
- ~~Аjax запросы: порядок~~
- ~~Скорость JS~~_____



selenide.org

Динамика: Selenide поможет!

```
@Test
```

```
public void userCanLogin() {  
    open("http://localhost:8080/login");  
    $(By.name("username")).setValue("john");  
    $("#submit").click();  
    $(".menu").shouldHave(text("Hello, John!"));  
}
```

Умные ожидания

```
$(".loading_progress").shouldBe(visible);
```

```
$(By.name("gender")).should(disappear);
```

```
$("#menu")  
    .shouldHave(text("Hello"), text("John!"))  
    .shouldBe(enabled, selected);
```

Забудьте про Ajax

Умные ожидания

```
$(".loading_progress").shouldBe(visible);
```

Все `$.should*()` методы
могут подождать
до 4 секунд

Умные ожидания

```
$(".loading_progress").shouldBe(visible);
```

Все `$.should*()` методы

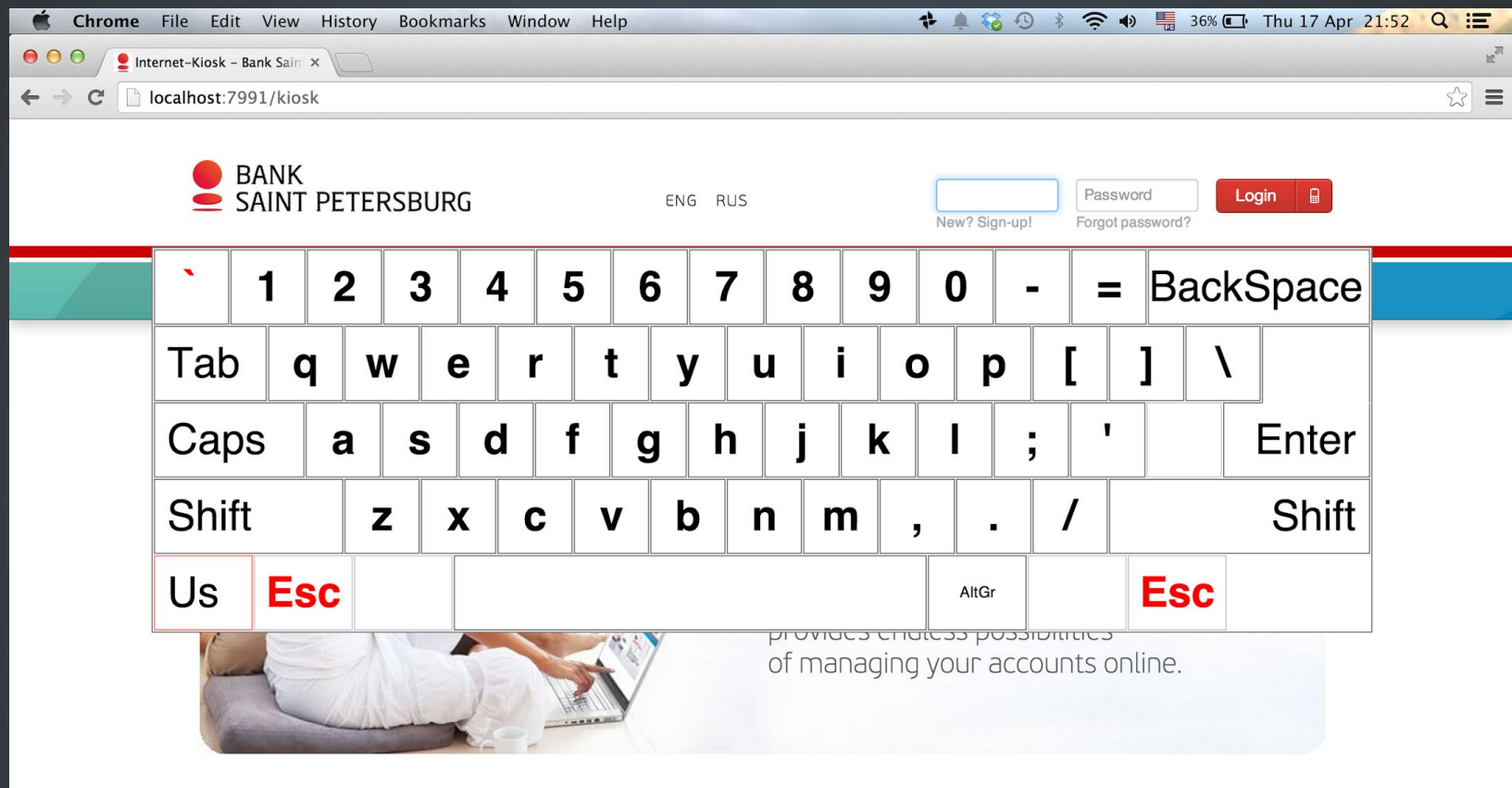
могут подождать

до 4 секунд

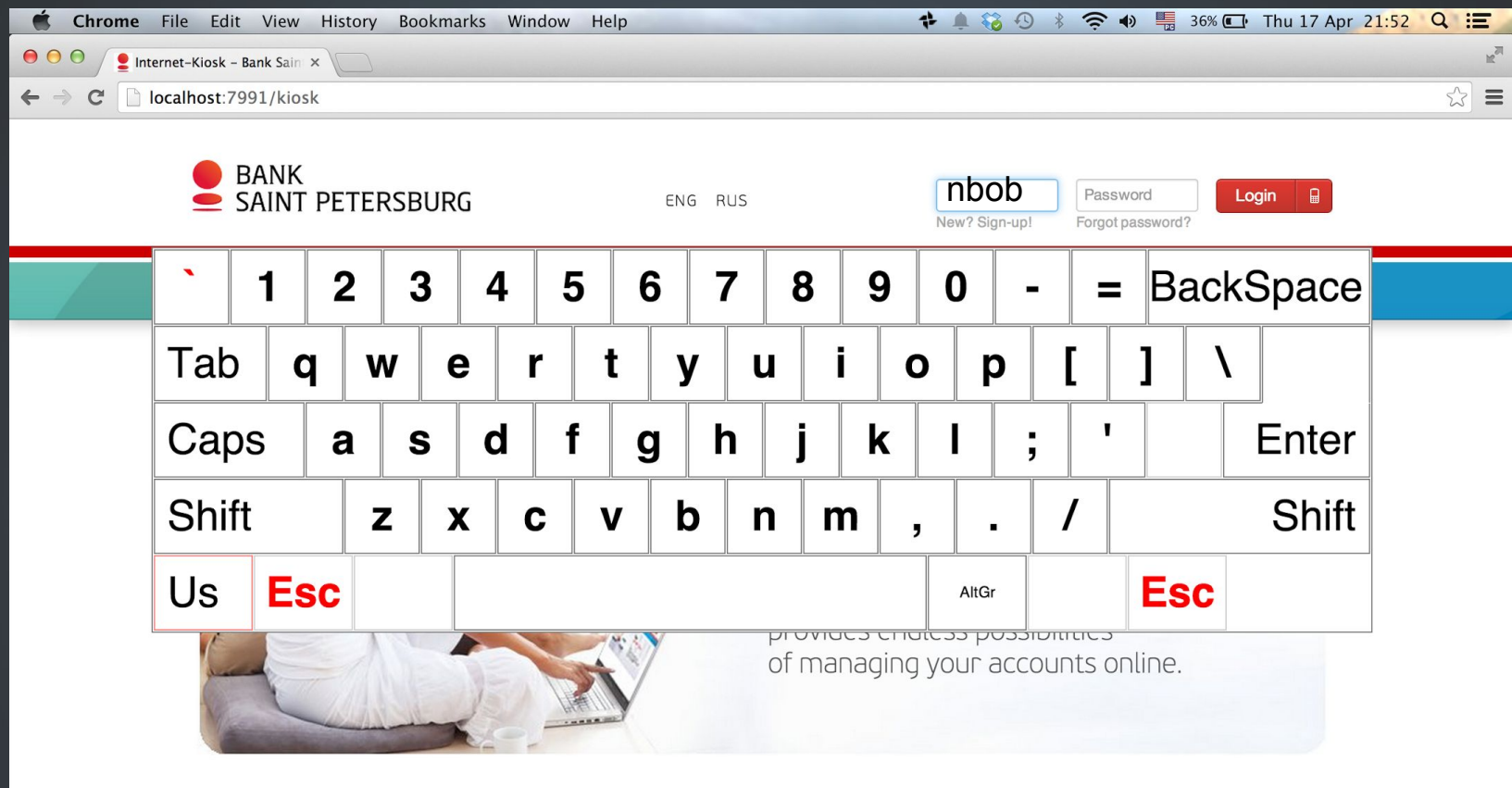
```
mvn -Dselenide.timeout=8000
```

Пример 2: **nbob**

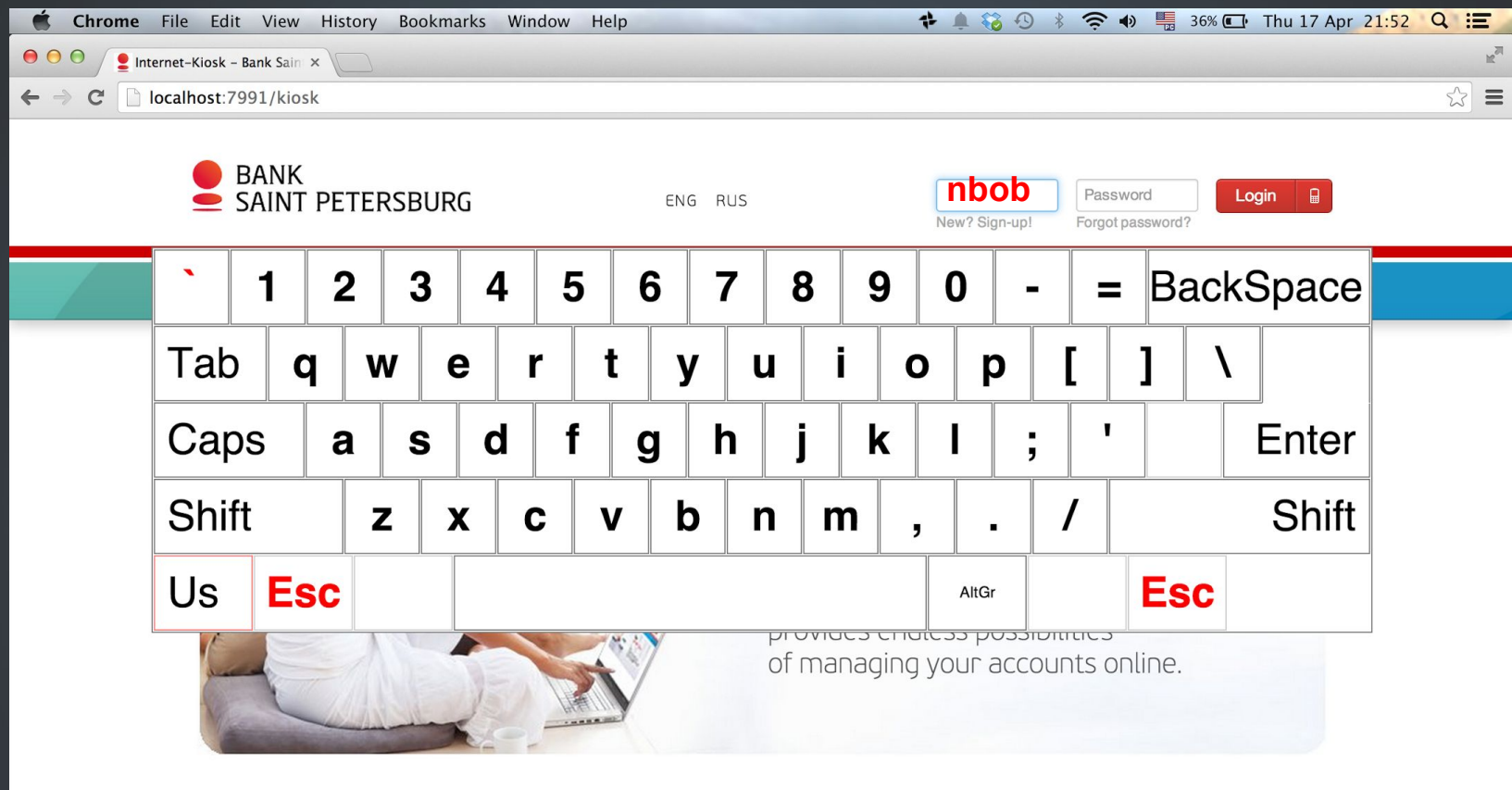
Это невозможно!



Хотим залогиниться юзером “**bob**”.



А в поле почему-то “nbob”.



Но слова “nbob” нет нигде в проекте...

Смотрим код:

```
@Test  
public void loginKiosk() {  
    open("http://localhost:9000/kiosk");  
    $("body").click();  
    $(By.name("username")).sendKeys("bob");  
    $("#login").click();  
}
```

Виновата эта строка:

```
@Test
public void loginKiosk() {
    open("http://localhost:9000/kiosk");
    $("body").click();    // появляется "n"
    $(By.name("username")).sendKeys("bob");
    $("#login").click();
}
```

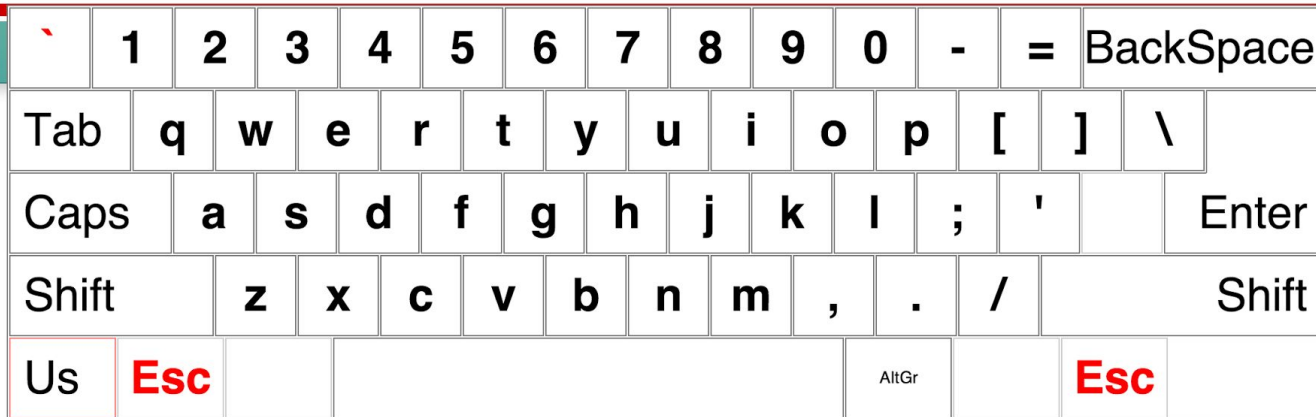



ENG RUS

New? Sign-up!

Forgot password?

Login



provides endless possibilities
of managing your accounts online.

завершить
работу

↑ вверх

↓ вниз

Chrome File Edit View History Bookmarks Window Help Thu 17 Apr 21:52

Internet-Kiosk - Bank Saint Petersburg

localhost:7991/kiosk

BANK SAINT PETERSBURG ENG RUS

Password **Login**

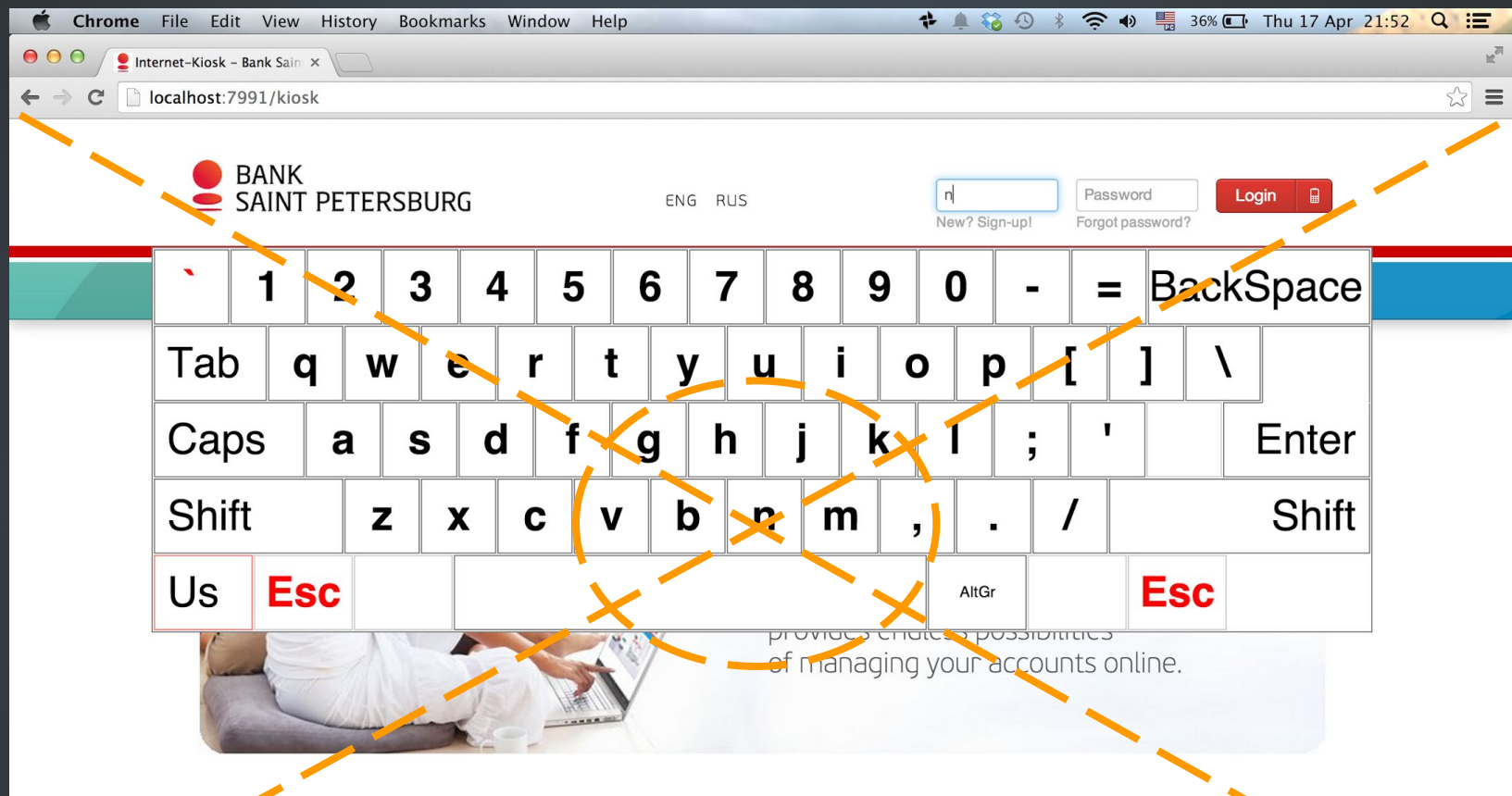
New? Sign-up! Forgot password?

	1	2	3	4	5	6	7	8	9	0	-	=	BackSpace
Tab	q	w	e	r	t	y	u	i	o	p	[	]	\
Caps	a	s	d	f	g	h	j	k	l	;	'		Enter
Shift		z	x	c	v	b	n	m	,	.	/		Shift
Us	Esc								AltGr			Esc	

provides endless possibilities of managing your accounts online.



`$(“body”).click(); // появляется “n”`



`$("#body").click(); // появляется "n"`

Зачем это было?

Когда-то это вставили, чтобы (наверное)
убрать фокус с поля:

```
$(“body”).click();
```

Мораль:

Не надо вставлять что попало в <body>!

Почему тесты нестабильные?

- Ajax запросы: скорость
- Ajax запросы: порядок
- Скорость JS
- **Размер окна браузера**
- **Суета!**
- ...

ПЛОХО:

```
Configuration.startMaximized = true;
```

хорошо:

```
Configuration.browserSize = "1024x768";
```



Пример 3: Фантомные счета

Миллион совпадений

Пример 3: фантомные счета



```
$$("#loans .loan").shouldHave(size(5));
```

Счёт

Кредитный лимит

Остаток

Зарплатный



350 000.00 RUB

50817 810 0 4800 0104467



100 500.00 RUB

6 780.00 RUB

40817 810 0 6666 6666667



5 000.00 EUR

40817 810 0 6666 5555557



13 000.00 USD

30601 810 8 3200 0012661



100 000.00 RUB

Пример 3: фантомные счета

```
$$("#loans tbody tr").shouldHave(size(5));
```

Но иногда -

ListSizeMismatch :

expected: = 5, actual: 6

Откуда-то берётся лишний счёт!

Пример 3: фантомные счета

LoansTest

// Создаёт счёт



100500 др.
тестов...

AccountsTest

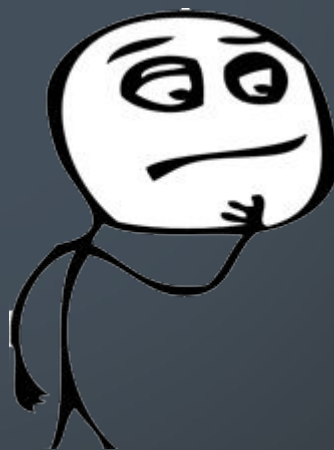
// Ожидает 5 счетов

Loa... Создает счёт



Асс... Ожидает 5 счетов

```
@After  
public void cleanup() {  
    loans.delete(  
        "where id = 6");  
}
```



КЭШ

AccountService

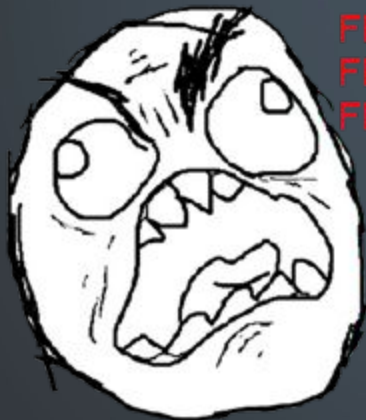
@CacheFor("5mn")

```
public List<Account> accounts(...) {  
    return ...  
}
```

```
public class UITest {  
    @Before  
    public void setUp() {  
        api.clearCache();  
    }  
}
```

```
public class AccountsTest  
    extends UITest {  
    @Before  
    public void setUp() {  
        open("/accounts");  
    }  
}
```

```
public class UITest {  
    @Before  
    public void setUp() {  
        api.clearCache();  
    }  
}
```

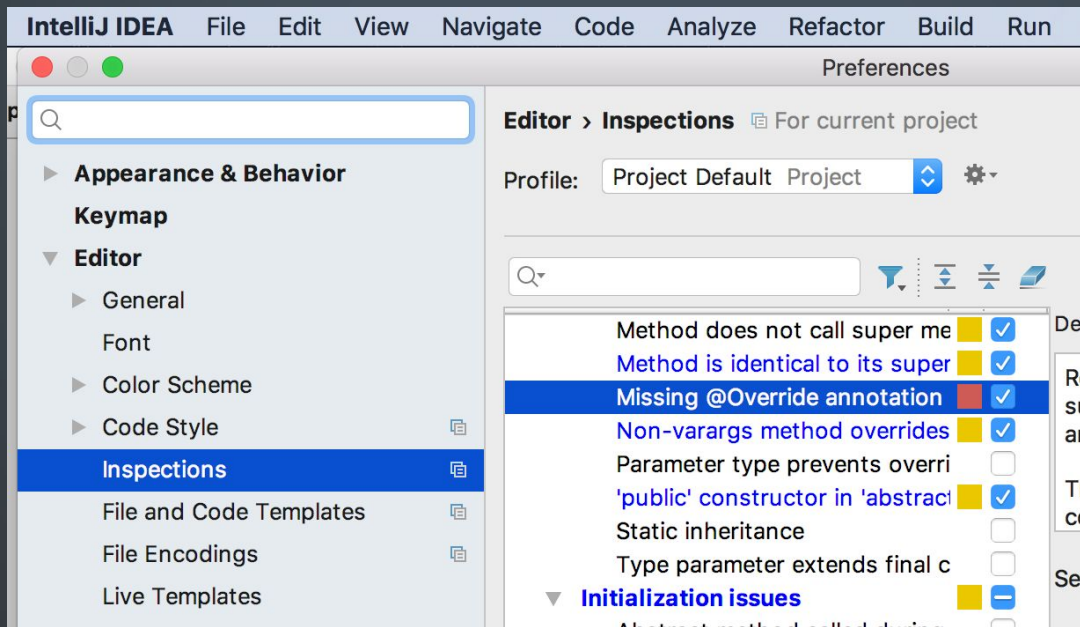


FFFFFFF
FFFFFFF
FFFFFFF
FFFFF
FFFFF
FFFFF
UUUU
UUUU
UUUU-

```
public class AccountsTest  
    extends UITest {  
    @Before  
    public void setUp() {  
        open("/accounts");  
    }  
}
```

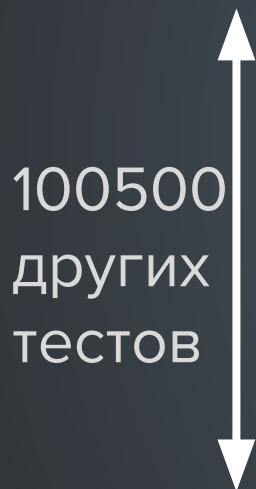
Мораль:

1. **Clean code** в тестах ТОЖЕ ВАЖЕН!
2. Используйте **проверки** в **IDEA**:



Зависит от скорости и порядка:

LoansTest



AccountsTest

> 5 минут - Тест ok

< 5 минут - Тест NOK

(кэш счетов - 5 минут)

Почему тесты нестабильные?

- Аjax запросы: скорость
- Аjax запросы: порядок
- Скорость JS
- Размер окна браузера
- **Кэш приложения**
- **Данные от предыдущих тестов**
- **Время**
- ...

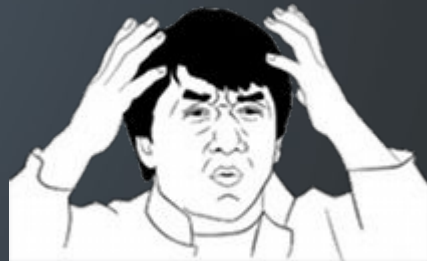
Пример 4: Время Java

миллисекунды или наносекунды?

Случай:

Иногда тест падает, потому что ...
дата платежа - **в будущем**

```
assert payment.time <= new Date();  
// иногда падает!  
...  
payment.time = new Date();
```



Как такое возможно?

Смотрим лог приложения:

```
19:35:25,145 [1e8-453] INFO request Payments.success  
127.0.0.1 ccd4be41-fed9-4637-ae6-57e6c50ced85  
Desktop c-private paymentId=100006 -> RenderTemplate  
/app/views/Payments/success.html -3254 ms
```

Смотрим лог приложения:

```
19:35:25,145 [1e8-453] INFO request Payments.success  
127.0.0.1 ccd4be41-fed9-4637-ae6-57e6c50ced85  
Desktop c-private paymentId=100006 -> RenderTemplate  
/app/views/Payments/success.html -3254 ms
```



Логируем текущее время

Fri Jun 23 23:58:34 MSK 2017

Fri Jun 23 23:58:35 MSK 2017

Fri Jun 23 23:58:36 MSK 2017

Fri Jun 23 23:58:37 MSK 2017

Fri Jun 23 23:58:35 MSK 2017

Fri Jun 23 23:58:39 MSK 2017

Fri Jun 23 23:58:40 MSK 2017



Разгадка века

На сервере дженкинса запущено **два разных сервиса** для синхронизации времени - **ntp** и **systemd-timesyncd**:

```
$ ps aux | grep ntp
```

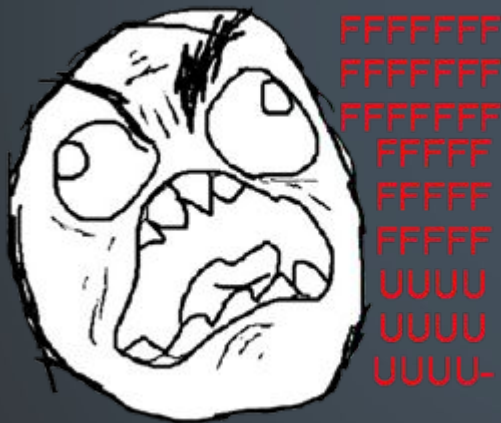
```
ntp          3485  0.0  0.0 110040  4556 ?  
Ssl  Jun20   0:22  /usr/sbin/ntpd -p  
/var/run/ntpd.pid -g -u 114:123
```

```
$ ps aux | grep systemd-timesyncd
```

```
systemd+    832  0.0  0.0 100324  2404 ?  
Ssl  Jun15   0:56  /lib/systemd/systemd-timesyncd
```

Разгадка века

На сервере дженкинса запущено **два разных сервиса** для синхронизации времени - **ntp** и **systemd-timesyncd**:



Конечно, они используют разные сервера и **мешают друг другу**.

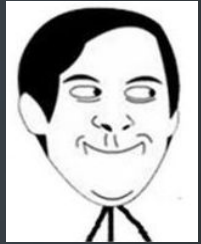
System.currentTimeMillis()

```
long start = System.currentTimeMillis();
```

... тра-ля-ля ...

```
long end = System.currentTimeMillis();
```

```
log.info("Loaded in {} ms", end-start);
```



System.nanoTime()

```
long start = System.nanoTime();  
... тра-ля-ля ...  
long end = System.nanoTime();  
  
log.info("Loaded in {} ms",  
        (end-start)/100000000);
```



Я всю жизнь думал, что

```
System.currentTimeMillis() == System.nanoTime() / 1_000_000;
```

А вот и НЕТ!

System.currentTimeMillis() - реальная дата

System.nanoTime() - некий счётчик

- Гарантированно растёт
- Связан со случайным моментом времени

Пример 5:

Проклятие зелёной кнопки

Помогло только видео

Пример 6:

Почему зависает Chrome?

Расследование длиной в 2 года

Проблема

Иногда Chrome зависает

- “Build is running 36 hours....”

Thread dump показывает:

- **Chromedriver зависает!**
- **Обычно в момент закрытия**

Since 2014

Поток висит в таком состоянии:

"Forwarding get on session 1a9a42 to remote"

```
java.lang.Thread.State: RUNNABLE
```

```
    at java.net.SocketInputStream.socketRead0(Native Method)
    at org.apache.http..(SessionInputBufferImpl.java:139)
    at org.openqa.selenium...(ApacheHttpClient.java:144)
    at com.codeborne.seleniumide.Selenide.open(Selenide.java:51)
    at ui.UITest.fastLogin(UITest.java:337)
    at ui.mobile...(MobileLoanEarlyRepaymentSpec.java:19)
```

Наши попытки

1. Таймаут для открытия/закрытия браузера

- Открывать браузер в отдельном потоке
- Закрывать браузер в отдельном потоке
- Таймаут 15 секунд, до 3 попыток

Результат:

- Не помогло

Sep 2015

Наши попытки

2. Убивать старые процессы chrome

Джоб в Jenkins “kill-chrome”

- `killall --older-than 1h chromedriver`
- `killall --older-than 1h chrome`

Результат:

- Не помогло

Nov 2015

Наши попытки

3. Включить debug логи приложения

4. Включить debug логи вебдрайвера

Результат:

- Ничего не нашли

Mar 2016

Наши попытки

5. Переоткрывать браузер

После каждых 20 тестов:

- Заккрыть браузер
- Открыть браузер

Результат:

- Проблема стала повторяться чаще!
- УРА!

Mar 2016

Наши попытки

6. Попробуем повторить проблему

- Цикл 1..1000
 - **Открыть** браузер
 - `open("http://localhost:9000/app")`
 - **Закрыть** браузер

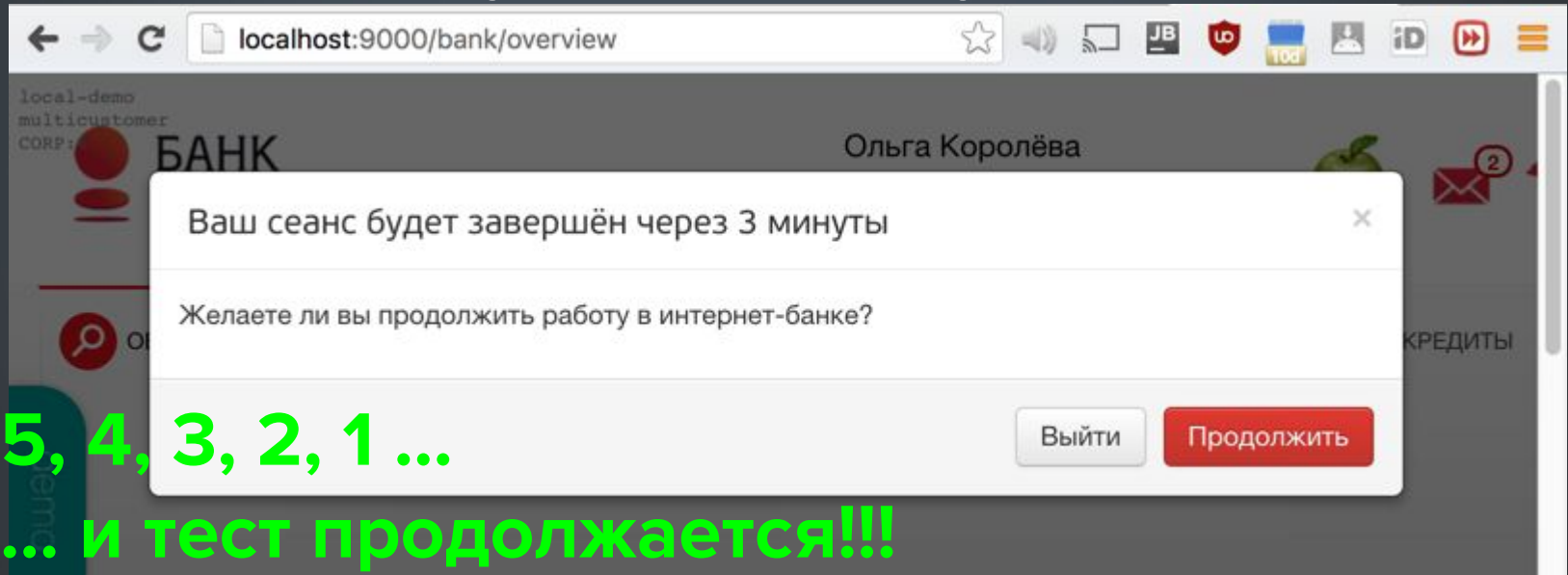
Результат:

- Повторяется *стабильно!*
- После каждых ~80 итераций

Mar 2016

И наконец...

- Случайно оставил тест бегать
- Через 20 минут - что я вижу:



- 5, 4, 3, 2, 1 ...

- ... и тест продолжается!!!

Результат:

- Chrome вовсе не зависает
- Chrome чего-то ждёт

Путём исключения нашли:

```
<script type="text/javascript">  
  var timeout =  
    setTimeout(sessionWatcher);  
</script>
```

Прогер, помни:

- **document.ready** или **\$(function);**

Правильно так:

```
<script type="text/javascript">  
  var timeout;  
  $(function() {  
    timeout = setTimeout(...);  
  });  
</script>
```


A still from the TV show 'The Walking Dead' featuring Rick Grimes and Carl Grimes. Rick is on the right, looking distressed with blood on his shirt. Carl is on the left, wearing a cowboy hat and looking down. The background is a brick wall with 'C BLOCK' painted on it. Large white text is overlaid in the center.

**2 ГОДА,
КАРЛ!!!**

Как ни странно,
Успешной оказалась
самая нелепая и
бессмысленная попытка



**Почему тесты
бывают нестабильными?**

Типичные проблемы:

- Ajax запросы: скорость, порядок
- Скорость JS
- Размер окна браузера
- Кэш
- Время
- Стабильность браузера
- UI эффекты
- Параллельные браузеры (потеря фокуса)

А также:

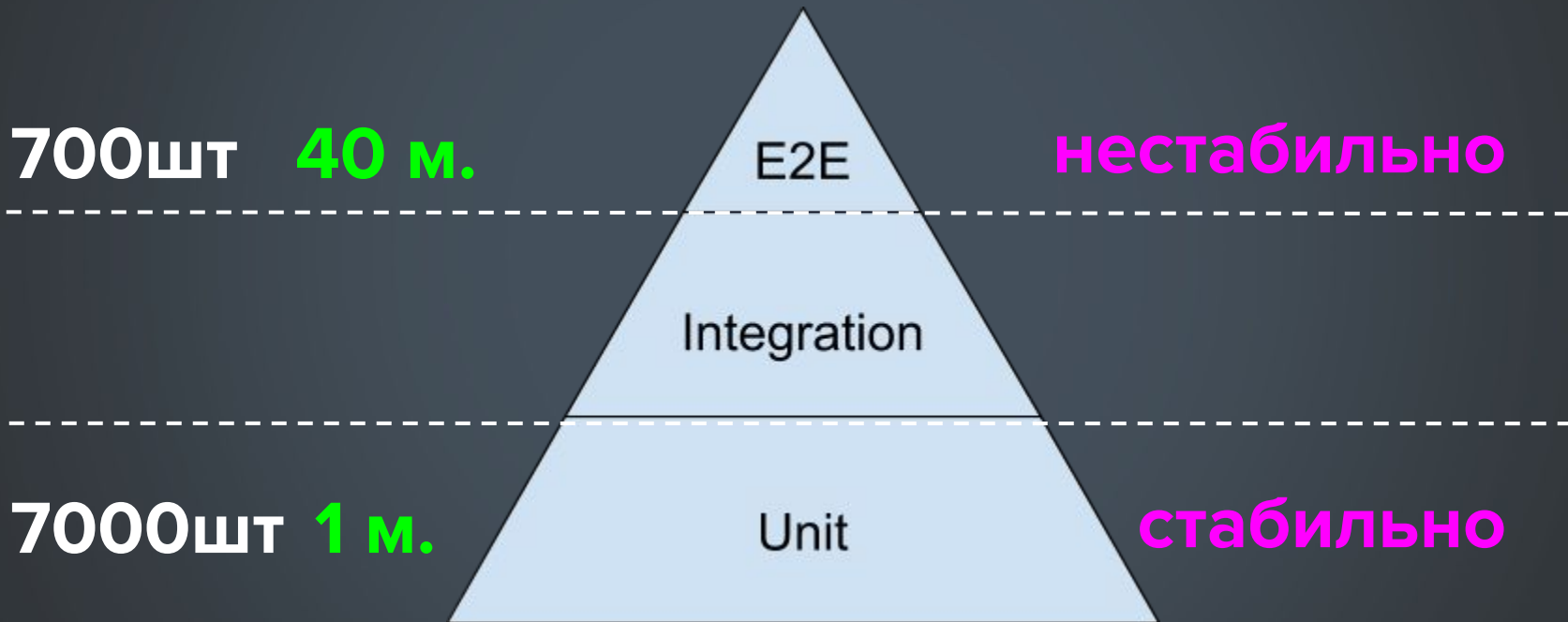
- БАГИ!
 - Трудновоспроизводимые
 - Нереальные
 - Некритичные
 - Usability

И поэтому никто их исправлять не будет :(

Профилактика

1. Пирамида
2. Selenide
3. Эмуляторы сервисов
4. Чистая база перед каждым тестом

Пирамида - это обязательно!



Вооружитесь:

1. Логи предыдущих билдов

Jenkinsfile

```
finally {  
    stage("Reports") {  
        junit 'build/test-results/**/*.xml'  
        artifacts =  
'build/reports/**/*.*,build/test-results/**/*.*,logs/**/*.*'   
        archiveArtifacts artifacts: artifacts  
    }  
}
```

Вооружитесь:

1. Логи предыдущих билдов



2. Скриншоты



Вооружитесь:

1. Логи предыдущих билдов



2. Скриншоты



3. Видео

Видео - вариант 1:

```
@Test @Video  
public void flakyTest() {  
    ....  
}
```

[http://automation-remarks.com/
video-recorder-java/](http://automation-remarks.com/video-recorder-java/)

Видео - вариант 2:

testcontainers.org



```
@Rule
```

```
public BrowserWebDriverContainer chrome =  
    new BrowserWebDriverContainer()  
        .withRecordingMode(RECORD_ALL, new File("build"))  
        .withDesiredCapabilities(chrome());
```

Разработчики!

Участвуйте в тестировании

Они без вас не справятся!

Менеджеры!

**Позвольте разработкам
участвовать в тестировании!**

Иначе ваши деньги псу под хвост

Автоматизаторы!

Когда вернётесь,

1. Проведите профилактику

2. Вооружитесь

3. И ждите

И когда тест упадёт

-

Радуйтесь!

Вы выходите на охоту



Андрей Солнцев
@asolntsev
ru.selenide.org



//codeborne

