

KOTLINTEST И ВСЕ-ВСЕ-ВСЕ



Паша Финкельштейн, Lamoda

КТО Я

- Разработчик
- Люблю тестировать
- Люблю изучать новое
- Люблю Kotlin

О ЧЁМ ЭТО

- Для тестирования используется множество инструментов
- В джаве богатая экосистема
- И «проверенные годами» инструменты

ЧТО ДАЛ МИРУ JVM KOTLIN

Леонид Руденко — Java → Kotlin: пишите тесты проще



И МНОЖЕСТВО НОВЫХ БИБЛИОТЕК!!!

НО ЗАЧЕМ?

СТРУКТУРА

Сейчас как-то так:

```
class MyVeryImportantTest {  
    @Test  
    void feature_should_work_somewhat(){ /*snip*/ }  
    @Test  
    void feature_should_work_somewhat2(){ /*snip*/ }  
    @Test  
    void feature_should_work_somewhat3(){ /*snip*/ }  
}
```

A surrealist illustration depicting a flat Earth. In the foreground, a large sea turtle is shown from a side profile, its head and front flippers visible. The turtle's shell is a brownish-orange color with a distinct scaly pattern. On the flat surface of the Earth, which is supported by the turtle, a herd of elephants is standing. The Earth itself is a thin, flat disk with a blue and white surface representing oceans and clouds. The edges of the Earth are jagged and uneven. The background is a dark, starry space. A bright sun is visible in the upper left corner, and a full moon is in the upper right corner. The overall scene is a visual metaphor for the flat Earth theory.

ДОБРО ПОЖАЛОВАТЬ В ПЛОСКИЙ МИР

А ХОЧЕТСЯ ПРЕКРАСНОГО

```
class PasswordServiceTest {  
    "password service" {  
        should("hash passwords"){  
            assertSomething  
        }  
        should("check passwords"){  
            assertCorrect  
        }  
    }  
    "hashed passwords" {  
        should("be different"){  
            assertSomethingElse  
        }  
    }  
}
```

PARAMETRIZED TESTS

TestNG

```
@DataProvider(name = "test1")
public static Object[][] primeNumbers() {
    return new Object[][] { {2, true}, {6, false},
        {19, true}, {22, false}, {23, true}};
}
```

```
@Test(dataProvider = "test1")
void testPrimeNumberChecker(Integer inputNumber,
    Boolean expectedResult){ /* snip */}
```

PARAMETRIZED TESTS

JUnit 5

```
static Stream<Arguments> stringIntAndListProvider() {  
    return Stream.of(  
        arguments("apple", 1, Arrays.asList("a", "b")),  
        arguments("lemon", 2, Arrays.asList("x", "y"))  
    );  
}  
@ParameterizedTest  
@MethodSource("stringIntAndListProvider")  
void testWithMultiArgMethodSource(String str,  
    int num, List<String> list) { /* snip */ }
```

ВСЁ ЕЩЁ ШУМНОВАТО

- Аннотации
- `Arrays.asList` (*в новой джаве лучше*)
- Строки, в которых можно ошибиться
- Отдельные методы

ТЕСТЫ ДОЛЖНЫ ЧИТАТЬСЯ ЛЕГКО



KOTLINTEST

ПОГОВОРИМ О БЕЗОПАСНОСТИ



КАК МОЖНО ХРАНИТЬ ПАРОЛИ?

КАК МОЖНО ХРАНИТЬ ПАРОЛИ?

- Plain text

КАК МОЖНО ХРАНИТЬ ПАРОЛИ?

- Plain text
- MD5(pass)

КАК МОЖНО ХРАНИТЬ ПАРОЛИ?

- Plain text
- MD5(pass)
- MD5(pass + salt)

КАК МОЖНО ХРАНИТЬ ПАРОЛИ?

- Plain text
- MD5(pass)
- MD5(pass + salt)
- MD5(MD5(pass + salt) + salt)

КАК МОЖНО ХРАНИТЬ ПАРОЛИ?

- Plain text
- MD5(pass)
- MD5(pass + salt)
- MD5(MD5(pass + salt) + salt)

MD5 — плохо потому что быстро

hashcat — очень быстрый

ПЛАКАТЬ!

Или использовать **BCrypt**

Чтобы узнать сколько раундов было использовано хэш начинается со структуры

\$2A\$12\$

- \$2a — версия bcrypt.
- \$12 - количество раундов шифрования

Hashcat на GeForce 1080Ti перебирает ≈ 200 паролей в секунду. Это 7 лет на 5-символьный пароль.

ДЕМО

НО ЕСТЬ И ПРОБЛЕМЫ

- Property-based тесты не имеют уникального идентификатора (*но в пути*)
 - В jqwik сделано лучше
- В репортах логируется только последняя текстовая часть теста
- В IDEA поддержка так себе: тесты нельзя запускать по одному

ЧТО НАДО ЗАПОМНИТЬ

Kotlintest решает для нас проблемы

- Структурирования
- Параметризованных тестов
- Property-based тестирования

СПАСИБО!

```
"Вопросы сейчас?"{ questions.shouldBeAskedNow() }
```

```
"Вопросы потом" {  
  questions shouldBeSentTo oneOf(  
    twitter to "asm0di0",  
    facebook to "asm0dey",  
    github to "asm0dey",  
    telegram to "asm0dey"
```