

# Kotest and his friends

Pasha Finkelshteyn, JetBrains

# Who am I

- Developer Advocate
- Love testing
- Love learning new things
- ❤️ Kotlin

# What I'm talking about

- Plenty of instruments for testing
- Java has rich ecosystem
- "Proven over years" technologies

# What Kotlin brought into the JVM world

- DSL
- Extension methods
- Functions as 1st class citizens

# And plenty of libraries!

A dark, atmospheric forest scene with tall trees and a path.

**But why?**

# Current test structure

Like *this*

```
class MyVeryImportantTest {  
    @Test  
    void feature_should_work_somewhow(){ /*snip*/ }  
    @Test  
    void feature_should_work_somewhow2(){ /*snip*/ }  
    @Test  
    void feature_should_work_somewhow3(){ /*snip*/ }  
    // much moar is coming!  
}
```



# Welcome to the Discoworld!

By Sir Terence David John Pratchett

# We wanna nice things!

```
passwordService {  
    shouldHash {  
        length { /* snip */ }  
        salt { /* snip */ }  
    }  
    shouldCheck {  
        validPasswords()  
        invalidPasswords()  
    }  
}
```

# 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 */}
```

# Still to noisy

- Annotations
- Arrays.asList (*Looks better in newer Java*)
- Strings where we will make mistakes
- Separate methods

# Tests should be easy to read





Sir can you spare a  
minute

To talk about enterprise security?

Sage of  
enterprise security

# We need to store passwords. How?

- Plain
- MD5(plain)
- MD5(plain+salt)
- MD5(md5(plain)+salt)

# We need to store passwords. How?

- Plain
- MD5(plain)
- MD5(plain+salt)
- MD5(md5(plain)+salt)

Rapid!

**200 GH/s** @ 8x Nvidia GTX 1080 Founders Edition with *Hashcat*

# What should we do now??

Cry

Or use bcrypt/scrypt

Hash starts with `$2a$10`

`$2a` – bcrypt version

`$10` – number of salting rounds

The are **very** slow (100/1000 per second)

# kotest

It's demo time!

# What should we remember

kotest solves problems od

- Structure
- Parametrized testing
- Property-based testing

# Thank you! Time for Q&A!

Pasha Finkelshteyn, JetBrains

 [@asm0di0](https://twitter.com/asm0di0)