Writing Tests

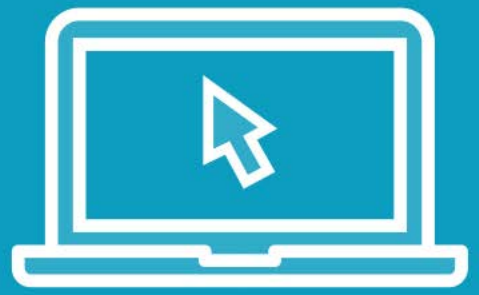
Esteban Herrera
JAVA ARCHITECT

@eh3rrera www.eherrera.net

Overview

- Test structure
- Lifecycle methods
- Test hierarchies
- Assertions
- Disabling tests
- Assumptions
- Test interfaces and default methods
- Repeating tests

Demo

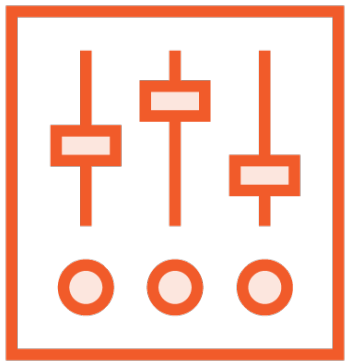


Write tests

How a good unit test is structured

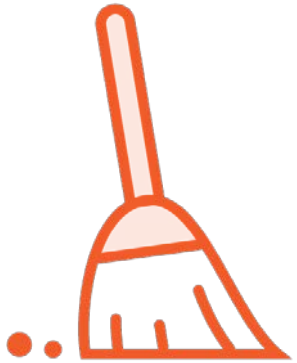
Four Phases of Every Test

Arrange



Act

Assert



Annihilation

Lifecycle Methods

Test Fixture

Everything we need to execute the test

Types for Managing Test Fixtures

Transient Fresh

Persistence Fresh

Persistence Shared

Lifecycle Annotations

Once per method	Once per class
@BeforeEach	@BeforeAll
@AfterEach	@AfterAll

Lifecycle Execution

Per method (default)

Per class

```
@TestInstance(TestInstance.Lifecycle.PER_METHOD)
@TestInstance(TestInstance.Lifecycle.PER_CLASS)
```

Lifecycle Execution

Annotations

```
-Djunit.jupiter.testinstance.lifecycle.default=per_method
-Djunit.jupiter.testinstance.lifecycle.default=per_class
```

Lifecycle Execution

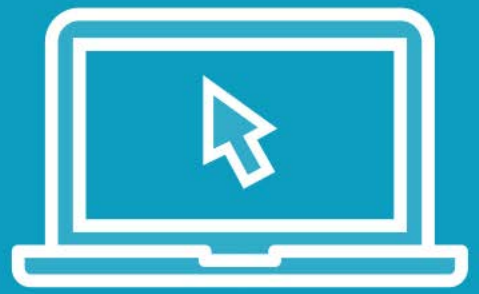
JVM options

```
junit.jupiter.testinstance.lifecycle.default=per_method
junit.jupiter.testinstance.lifecycle.default=per_class
```

Lifecycle Execution

junit-platform.properties

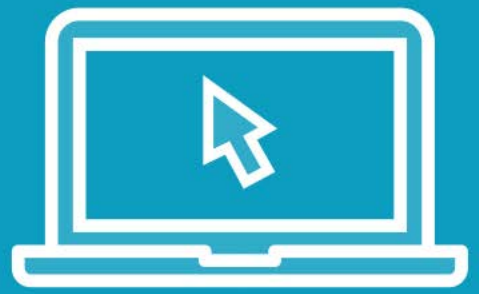
Demo



Lifecycle methods

- Annotations
- Per method/class

Demo



Test hierarchies

Behavior-Driven Development (BDD)

An application is specified and designed by describing how it should behave

BDD Naming Style

Test Phases	BDD
Arrange	Given
Act	When
Assert	Then

Nested Test Classes

Only non-static inner classes

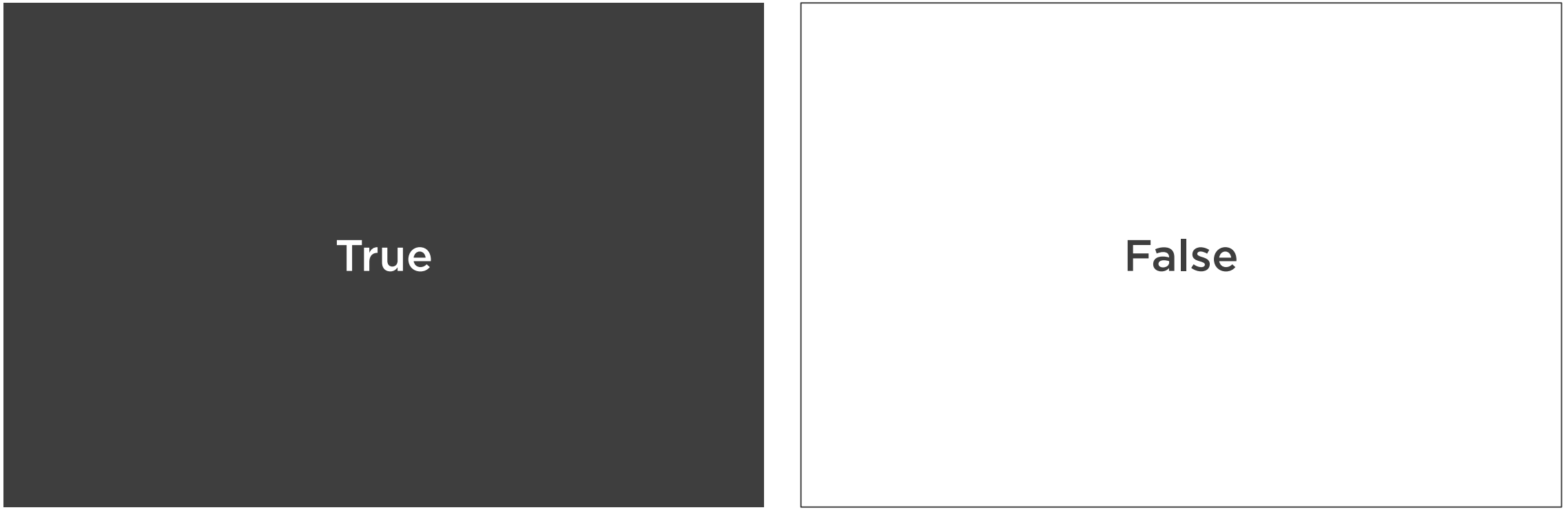
@BeforeAll and @AfterAll don't work by default

- Only with Lifecycle.PER_CLASS

Use them with @DisplayName

Assertions

Test Result



A Single Assertion?

```
void test() {
  ...
  assertTrue(...);
  assertNotNull(...);
  assertEquals(...);
}
```

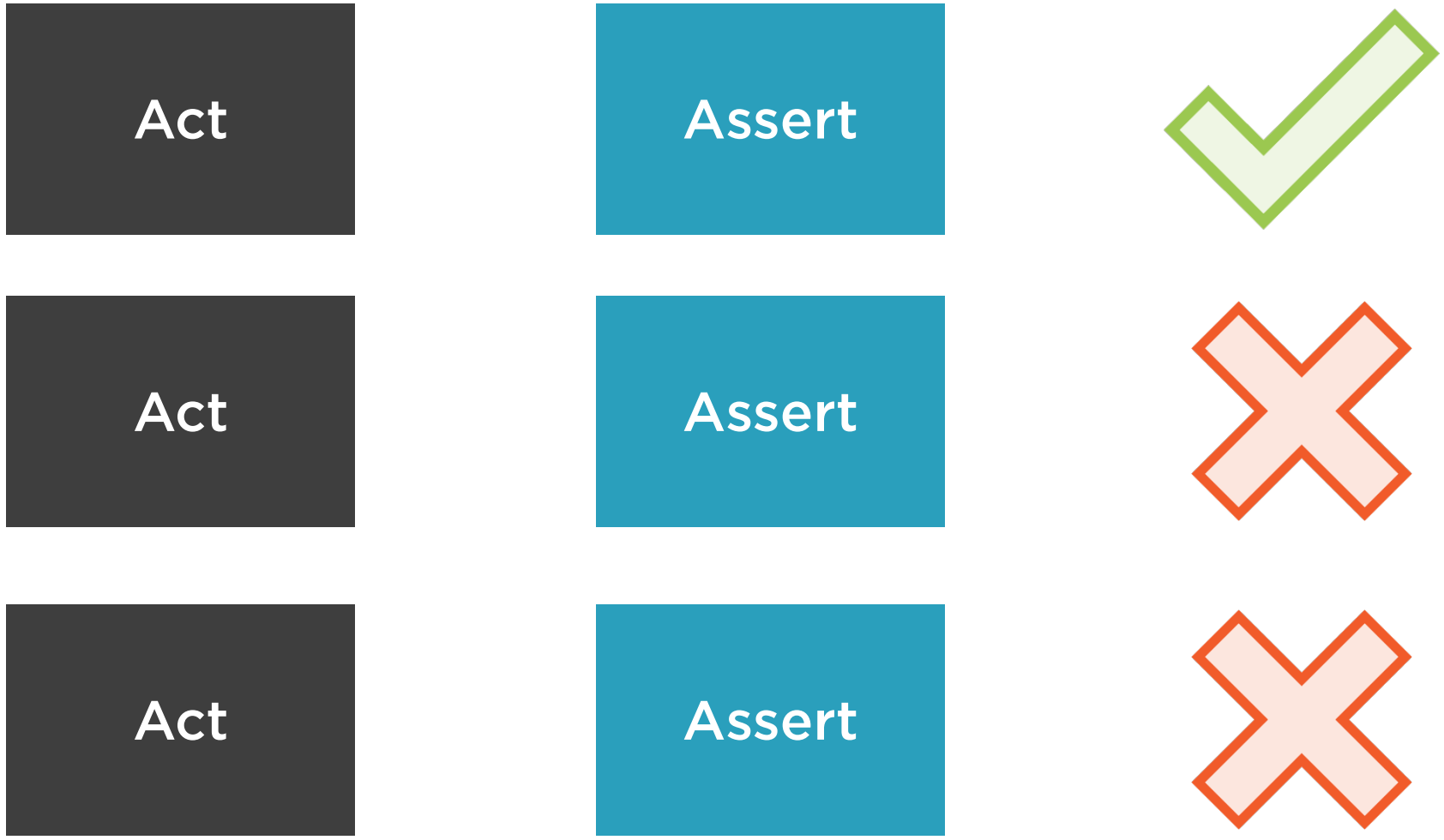
A Single Assertion?

```
void test() {
  ...

  conditionOne && conditionTwo || conditionThree

}
```

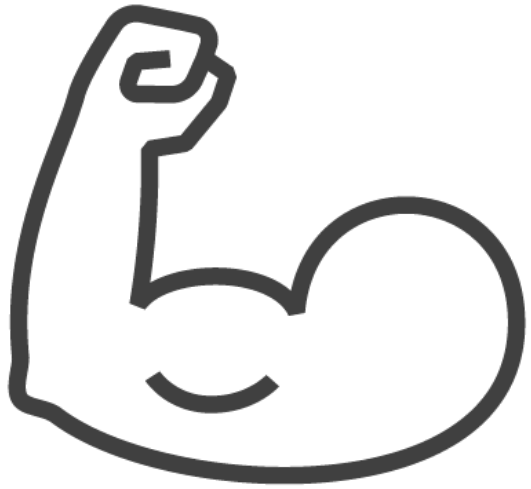
One Act/Assert Operations



JUnit Jupiter Assertions

- assertAll
- assertArrayEquals
- assertEquals
- assertFalse
- assertIterableEquals
- assertLinesMatch
- assertNotEquals
- assertNotNull
- assertNotSame
- assertNull
- assertSame
- assertThrows
- assertTimeout
- assertTimeoutPreemptively
- assertTrue
- fail

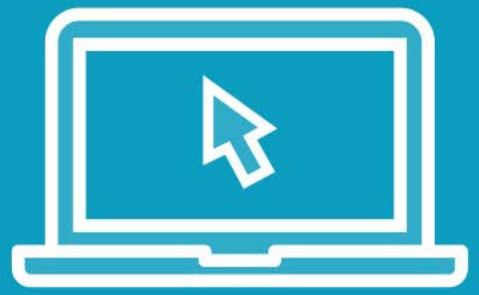
Need More Power?



External assertion libraries

- AssertJ
- Hamcrest

Demo



JUnit Jupiter assertions

- Error messages
- assertAll
- assertThrows
- assertTimeout
- assertTimeoutPreemptively

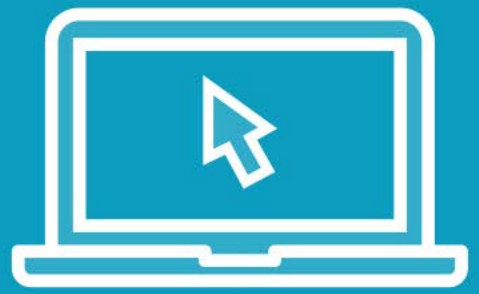
Disabling Tests

The Annotation

@Disabled

- Methods
- Classes

Demo



Disabling tests

Assumptions

Assumptions

Based on conditions

Don’t result in test failure like assertions

Abort the test

```
assumeTrue(boolean assumption)

assumeTrue(boolean assumption, String message)

assumeTrue(BooleanSupplier assumptionSupplier)

assumeTrue(boolean assumption, Supplier<String> message)

assumeTrue(BooleanSupplier assumptionSupplier, String message)

assumeTrue(BooleanSupplier assumptionSupplier, Supplier<String> message)
```

assumeTrue

```
assumeFalse(boolean assumption)

assumeFalse(boolean assumption, String message)

assumeFalse(BooleanSupplier assumptionSupplier)

assumeFalse(boolean assumption, Supplier<String> message)

assumeFalse(BooleanSupplier assumptionSupplier, String message)

assumeFalse(BooleanSupplier assumptionSupplier, Supplier<String> message)
```

assumeFalse

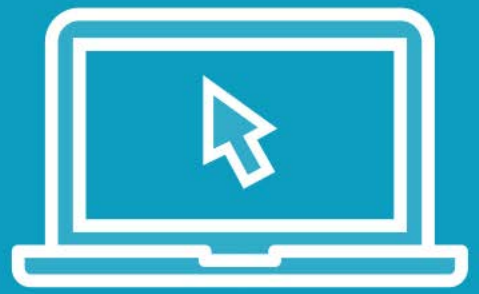
```
assumingThat(boolean assumption, Executable message)

assumingThat(boolean assumption, Executable executable)

assumingThat(BooleanSupplier assumptionSupplier, Executable executable)
```

assumingThat

Demo



Assumptions

Test Interfaces and Default Methods

What to Include in Interfaces?

@Test

@BeforeEach

@AfterEach

What to Include in Interfaces?

@RepeatedTest

@ParameterizedTest

@TestFactory

@TestTemplate

@ExtendWith

@Tag

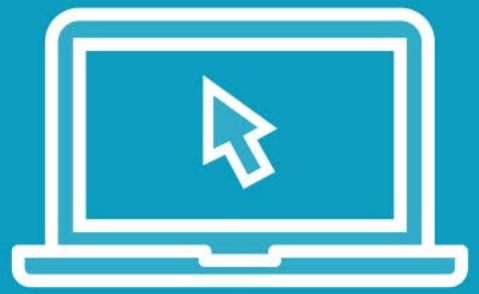
What to Include in Interfaces?

@BeforeAll

@AfterAll

@TestInstance(Lifecycle.PER_CLASS)

Demo



Test interfaces and default methods

- Extract methods

Repeating Tests

@RepeatedTest

Repeat a test

Fixed number of repetitions

Full support of lifecycle

{displayName}

{currentRepetition}

{totalRepetitions}

Custom Display Name

Placeholders

```
RepeatedTest.LONG_DISPLAY_NAME
{displayName} :: repetition {currentRepetition} of {totalRepetitions}
Ex: My Test :: repetition 1 of 10

RepeatedTest.SHORT_DISPLAY_NAME
repetition {currentRepetition} of {totalRepetitions}
Ex: repetition 1 of 10
```

Custom Display Name

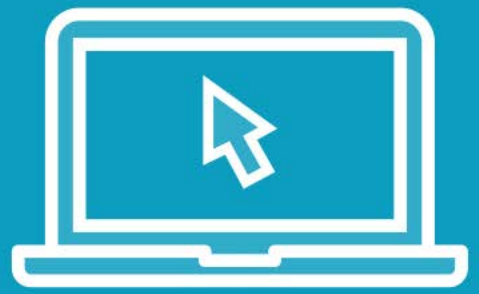
Predefined formats

```
int getCurrentRepetition();
int getTotalRepetitions();
```

RepetitionInfo Interface

@RepeatedTest, @BeforeEach, and @AfterEach

Demo



Repeating tests

- @RepeatedTest
- Custom display name
- RepetitionInfo interface

Summary

- Test structure
- Lifecycle methods
- Test hierarchies
- Assertions
- Disabling tests
- Assumptions
- Test interfaces and default methods
- Repeating tests