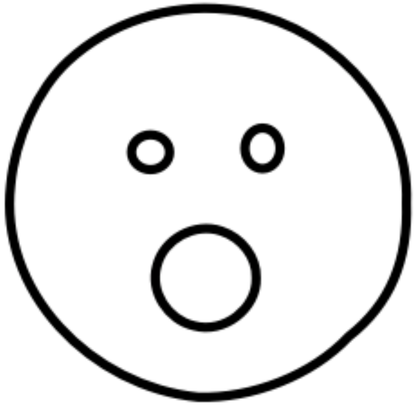
Things you *might* not know about JUnit

Richard Kettelerij

Beyond....

```
@Test
public void ourPlanetShouldBeEarth() {

    Planet ourPlanet = galaxyService.getOurPlanet();

    assertThat(ourPlanet, is(planetEarth));
}
```

Assumptions



`Assume` is the mother of `@Ignore`

Assumptions

```
@Test
public void planetShouldBeWithinReach() {

    assumeThat(System.getProperty("currentPlanet"), is("earth"));
```

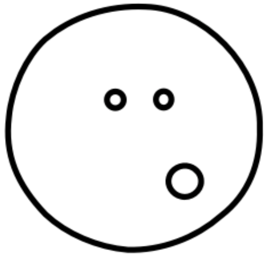
Assumptions

```java
@Test
public void planetShouldBeWithinReach() {

    assumeThat(System.getProperty("currentPlanet"), is("earth"));
```

```
Test 'nl.mindloops.GalaxyTest.planetShouldBeWithinReach' ignored
org.junit.AssumptionViolatedException: got: "mars", expected: is "earth"
```

Exception testing

```
@Test(expected = IllegalStateException.class)
public void shouldNotAllowRocketLaunch() {

    rocket.launchFrom(jupiter);
}
```



Exception testing

```
@Rule
public ExpectedException expectedException = ExpectedException.none();

@Test
public void shouldNotAllowRocketLaunchFromJupiter() {
    // other code

    expectedException.expect(IllegalStateException.class);
    expectedException.expectMessage("due to gas-surface");

    rocket.launchFrom(jupiter);
```

Exception testing

```
@Rule
public ExpectedException expectedException = ExpectedException.none();

@Test
public void shouldNotAllowRocketLaunchFromJupiter() {
    // other code

    expectedException.expect(IllegalStateException.class);
    expectedException.expectMessage("due to gas-surface");

    rocket.launchFrom(jupiter);
```

Exception testing (coming in v4.13)

```java
@Test
public void shouldNotAllowRocketLaunchFromJupiter() {
    // other code

    IllegalStateException e = expectThrows(IllegalStateException.class,
                                () -> rocket.launchFrom(jupiter));

    assertThat(e.getMessage(), containsString("due to gas-surface"));
}
```

Rules

```java
@Rule
public TemporaryFolder temporaryFolder = new TemporaryFolder();

                     temporaryFolder.newFile("rocketManual.pdf");
```

```java
@Rule
public TestRule timeout = new DisableOnDebug(Timeout.seconds(3));
```

Rules

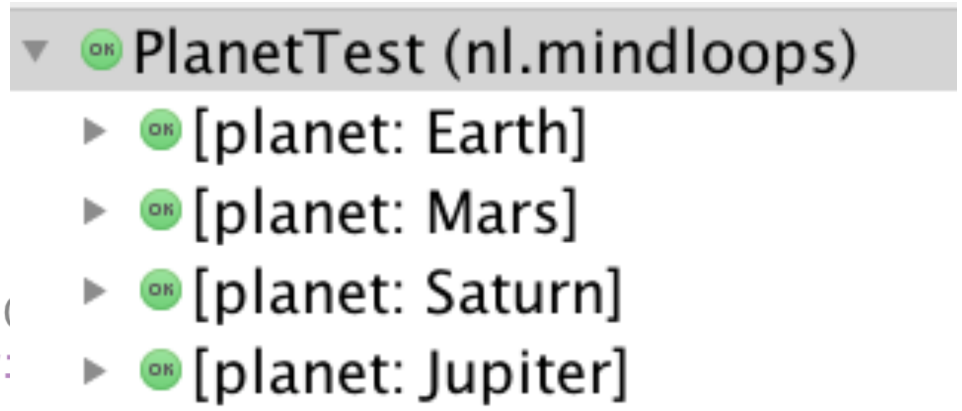
```
@ClassRule
public static TestRule timeoutAllTestsCombined
        = new DisableOnDebug(new Timeout(20, TimeUnit.MINUTES));
```

Parameterized tests

```
@RunWith(Parameterized.class)
public class PlanetTest {

    @Parameters(name = "planet: {0}")
    public static List<String> planets() {
        return Arrays.asList("Earth", "Mars", "Saturn", "Jupiter");
    }

    @Parameter
    public String planet;

    @Test
    public void shouldExistInGalaxy() {
        assertTrue(galaxyService.exists(planet));
    }
```

Parameterized tests

```
@RunWith(Parameterized.class)
public class PlanetTest {

    @Parameters(name = "planet: {0}")
    public static List<String> planets() {
        return Arrays.asList("Earth", "Mars", "Saturn", "Jupiter");
    }

    @Parameter
    public String planet;

    @Test
    public void shouldExistIn(
        assertTrue(galaxyServ
    }
```



```
▼ PlanetTest (nl.mindloops)
    ▶ [planet: Earth]
    ▶ [planet: Mars]
    ▶ [planet: Saturn]
    ▶ [planet: Jupiter]
```

Parameterized tests

```
@Test
@Parameters({"Earth", "Mars", "Saturn", "Jupiter" })
public void planetShouldExistInGalaxy(String planet) {
    assertTrue(galaxyService.exists(planet));
}
```

https://github.com/Pragmatists/**JUnitParams**

Parameterized tests

```
@Test
@FileParameters("classpath:planets.csv")
public void planetShouldExistInGalaxy(String planet) {
    assertTrue(galaxyService.exists(planet));
}
```

https://github.com/Pragmatists/**JUnitParams**

Theories

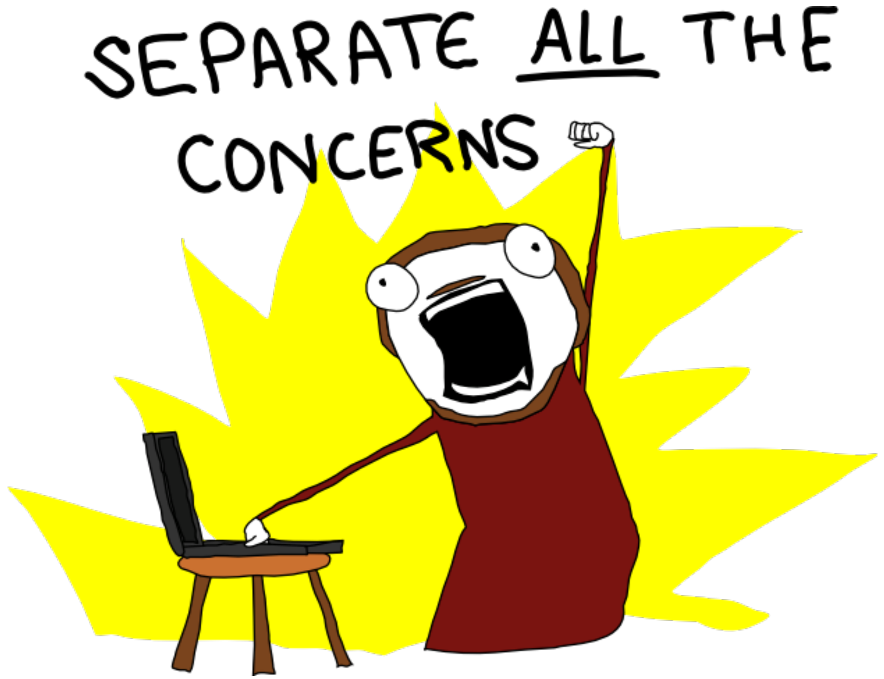
```
@Theory
public void rocketCanReachPlanet(
    @ParametersSuppliedBy(PlanetSupplier.class) Planet planet,
    @ParametersSuppliedBy(RocketSupplier.class) Rocket rocket) {

    assumeTrue(isInOurGalaxy(planet));
    // complex assertion
```

Alternatively: https://github.com/pholser/**junit-quickcheck**

Near future

JUnit Lambda

Goals

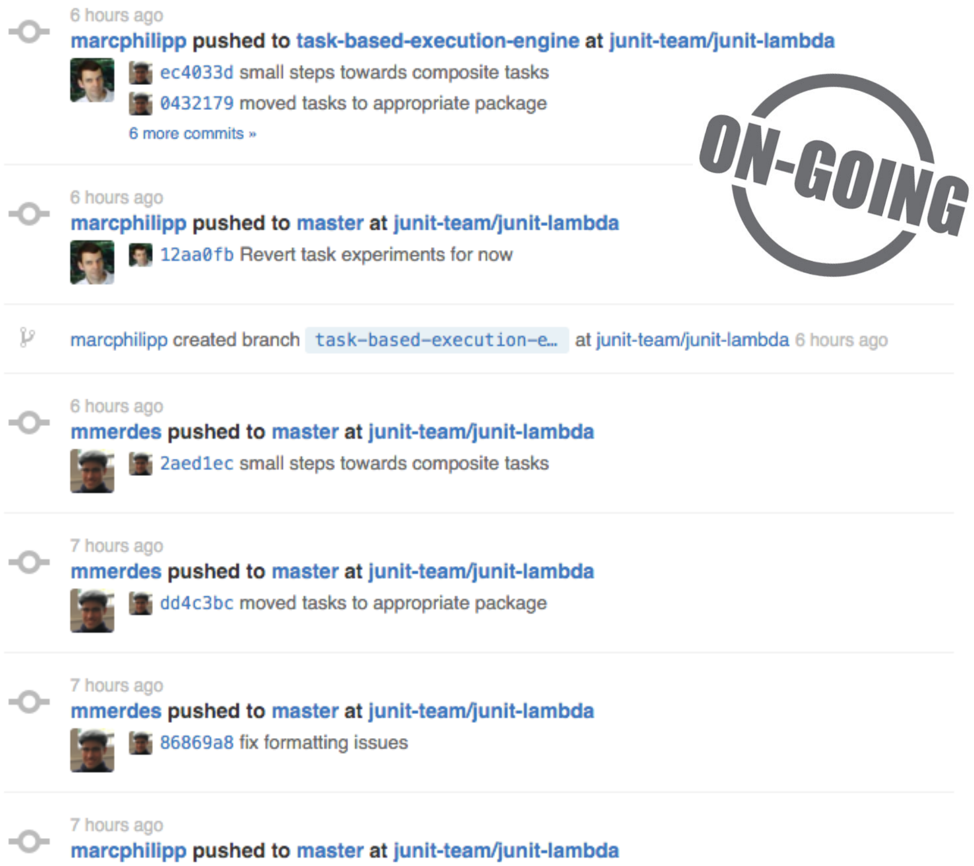


Goals

Java™ 8

Features

Near future


6 hours ago
marcphilipp pushed to task-based-execution-engine at junit-team/junit-lambda
ec4033d small steps towards composite tasks
0432179 moved tasks to appropriate package
6 more commits »

6 hours ago
marcphilipp pushed to master at junit-team/junit-lambda
12aa0fb Revert task experiments for now

marcphilipp created branch task-based-execution-e… at junit-team/junit-lambda 6 hours ago

6 hours ago
mmerdes pushed to master at junit-team/junit-lambda
2aed1ec small steps towards composite tasks

7 hours ago
mmerdes pushed to master at junit-team/junit-lambda
dd4c3bc moved tasks to appropriate package

7 hours ago
mmerdes pushed to master at junit-team/junit-lambda
86869a8 fix formatting issues

7 hours ago
marcphilipp pushed to master at junit-team/junit-lambda