



UNIwersYTET GDAŃSKI

# Wprowadzenie do JUnit5

---

Mateusz Miotk

12 marzec 2018

Instytut Informatyki UG

# JUnit4 vs JUnit5

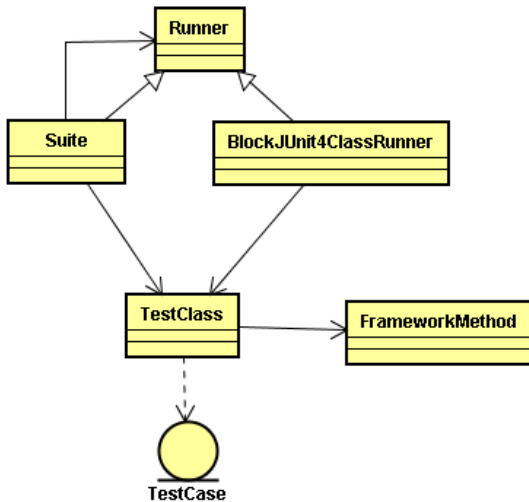
Największą bolączką narzędzia JUnit4 jest brak rozszerzalności tzn. nie można w łatwy sposób użyć tzw. dwóch runnerów (Słowo kluczowe **@RunWith**) w jednej klasie testowej.

To spowodowało że we wrześniu 2017 roku po ponad dwóch latach wyszła nowa wersja narzędzia JUnit5 (obecnie wersja 5.1).



Rysunek 1

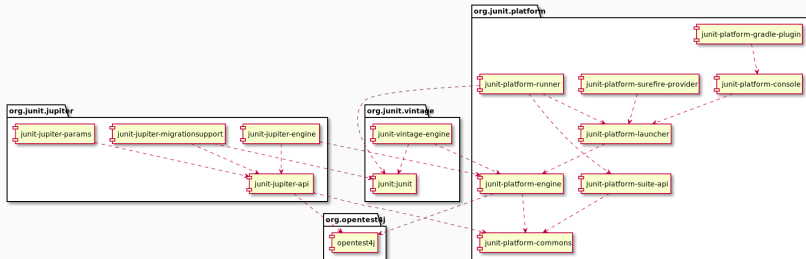
# JUnit4 vs JUnit5



Rysunek 2

# JUnit4 vs JUnit5

JUnit5 = JUnit Jupiter (Silnik) + JUnit Platform (Słowa kluczowe typu Before, Test etc.) + JUnit Vintage (Wsparcie dla starszych wersji JUnit)



Rysunek 3

# JUnit4 vs JUnit5

Różnica w słowach kluczowych:

JUnit4	JUnit5
@Before	@BeforeEach
@After	@AfterEach
@BeforeClass	@BeforeAll
@AfterClass	@AfterAll
@Ignore	@Disabled
@RunWith	@ExtendWith - większa funkcjonalność!!!

- Przedewszystkim rozszerzalność, dodając różne runnery do jednej klasy testowej.
- `assertThrows` (obsługa wyjątków).
- `assertAll` (sprawdza **wszystkie** asercje, które są zaznaczone).
- `@DisplayName` (opisywanie testów (nie tylko w nazwie metody!)).
- `@Nested` (zagnieżdżanie innych klas testowych w klasach).
- `@Tag` (kontrola uruchamiania testów (np. wyłączanie konkretnych) w narzędziach Maven czy Gradle).
- `@RepeatedTest` (powtarzanie testów, możliwość testów obciążeniowych).

**Asumpcja** - asercja/nie asercja, która odpalana jest przed testem. Przeważnie służy do sprawdzania środowiska. W przypadku nie zajścia asumpcji test zostanie oznaczony jako **nieuruchomiony**, mimo że zwróci się tutaj wyjątek.

```
@Test
```






```
void assumeExample() {  
    assumeTrue(!(5 != (3+3)));  
    assertEquals("Fizz", sut.play(15));  
}
```

## JUnit5 - testy parametryczne

```
@ParameterizedTest
    @ValueSource(ints = {15,30,150})
    public void testWithParameters(int p) throws Exception{
        assertEquals("Fizz", sut.play(p));
    }
```

Istnieją różne metody wywoływania testów parametrycznych (przez metody, test factory czy pliki .csv)



-  Dokumentacja JUnit5: <https://junit.org/junit5/>
-  Blog programisty: <https://koziolekweb.pl/>
-  Wprowadzenie do JUnit5 - confitura 2017:  
<https://www.youtube.com/watch?v=n5Bmqe9k9p0>
-  Introduction to JUnit5:  
<https://www.youtube.com/watch?v=-mlrA5cVfZ4>
-  B. Garcia, Mastering Software Testing with JUnit 5: Comprehensive guide to develop high quality Java applications, Packt Publishing, 2017.