# JUnit

# Podstawowe dependency

<dependency>  
 <groupId>org.junit.jupiter</groupId>  
 <artifactId>junit-jupiter-api</artifactId>  
 <version>5.6.2</version>  
 <scope>test</scope>  
</dependency>

Dobrze jest też użyć następujących:

<dependency>  
 <groupId>org.hamcrest</groupId>  
 <artifactId>hamcrest-all</artifactId>  
 <version>1.3</version>  
 <scope>test</scope>  
</dependency>

<dependency>  
 <groupId>org.assertj</groupId>  
 <artifactId>assertj-core</artifactId>  
 <version>3.16.1</version>  
 <scope>test</scope>  
</dependency>

## Testy parametryzowane

Dependency

<dependency>  
 <groupId>org.junit.jupiter</groupId>  
 <artifactId>junit-jupiter-params</artifactId>  
 <version>5.6.2</version>  
 <scope>test</scope>  
</dependency>

## Przykład testu parametryzowanego

### Metoda zasilająca dane testowe

private static Stream<Arguments> feedcountAddedRandomAliens() {  
 return Stream.*of*(  
 Arguments.*of*(Arrays.*asList*(  
 new RandomAlien("sectoid", "leader"),  
 new RandomAlien("floater", "navigator"),  
 new RandomAlien("ethereal", "commander")  
 )  
 ),  
 Arguments.*of*(Arrays.*asList*(  
 new RandomAlien("sectoid", "leader"),  
 new RandomAlien("zombie", "terrorist"),  
 new RandomAlien("sectoid", "soldier")  
 )  
 )  
 );  
}

### Sam test

@ParameterizedTest  
@MethodSource(value = "feedcountAddedRandomAliens")  
public void countAddedRandomAliens\_byRaceandRank\_resultTrue(List<RandomAlien> list) {  
 addAliens(list);  
 Assertions.*assertThat*(countAliens()).isEqualTo(list.size());  
}