# Parameterized test

Problem: Eine Testmethode soll mit verschiedenen Werten ausgeführt werden.

Ein naiver und unflexibler Ansatz (Verletzung des DRY-Prinzips):

|  |
| --- |
| @Test  **public** **void** testConstructor() {  Money money = **new** Money(10, "USD");  Assert.*assertEquals*(10, money.getAmount());  Assert.*assertEquals*("USD", money.getCurrency());  money = **new** Money(20, "EUR");  Assert.*assertEquals*(20, money.getAmount());  Assert.*assertEquals*("EUR", money.getCurrency());  } |

## **JUnit’s eigene Variante: der Parameterized-Runner.**

Bei diesem Verfahren wird jede neue Instanz der Testklasse (welche ja für jede @Test-Methode erzeugt wird) nach und nach mit den Werten von einer @Parameters-annotierten Methode per Konstruktor(!) initialisiert. Dies ist zwar zunächst ein nützliches feature, schränkt aber gleichzeitig die Parametrisierbarkeit von Testklassen ein.

siehe **ch04.x01JUnitParameterized.PrimeNumberChecker** und **ch04.x01JUnitParameterized.PrimeNumberCheckerTest**

daher:

## **Neue Dependency**

|  |
| --- |
| <dependency>  <groupId>pl.pragmatists</groupId>  <artifactId>JUnitParams</artifactId>  <version>1.1.0</version>  <scope>test</scope>  </dependency> |

Über JUnitParams und die Unterschiede zur JUnit’s eigener Parameters-Variante:

|  |
| --- |
| About JUnitParams project adds a new runner to JUnit and provides much easier and readable parametrised tests for JUnit >= 4.12.  Main differences to standard JUnit Parametrised runner:   * more explicit - params are in test method params, not class fields * less code - you don't need a constructor to set up parameters * you can mix parametrised with non-parametrised methods in one class * params can be passed as a CSV string or from a parameters provider class * parameters provider class can have as many parameters providing methods as you want, so that you can group different cases * you can have a test method that provides parameters (no external classes or statics anymore) * you can see actual parameter values in your IDE (in JUnit's Parametrised it's only consecutive numbers of parameters): |

## **Anleitung: Entwickle parametrisierten Test**

Gegeben: SUT: **ch03.x02firstJUnitTest.Money**

Test: **ch04.x02JUnitParams.MoneyParameterizedTest**

## **JUnit 5 parametrisierte Tests**

Die neue JUnit Version 5 hat sich des Problems der Parametrisierung von Tests angenommen: <http://junit.org/junit5/docs/current/user-guide/#writing-tests-parameterized-tests> .

Im Blog unter <http://blog.codeleak.pl/2017/06/cleaner-parameterized-tests-with-junit-5.html> werden erste Code-Beispiele gezeigt. Auch CSV-Dateien können so einfach als Quelle für Testparameter eingebunden werden.

Über die notwendigen maven-dependencies wird unter <https://www.petrikainulainen.net/programming/testing/junit-5-tutorial-running-unit-tests-with-maven/> geschrieben.