Contents

Provided Money and Currency API

Introduce

JSR354 targets to support all general applications: eCommerce, Banking, Finance…

The specification supports java se platform version 9 and above. The API is backward compatible with java 7,8. The reference implementation "**Moneta**" is modular and supports the Java Platform Module System. We will work it.

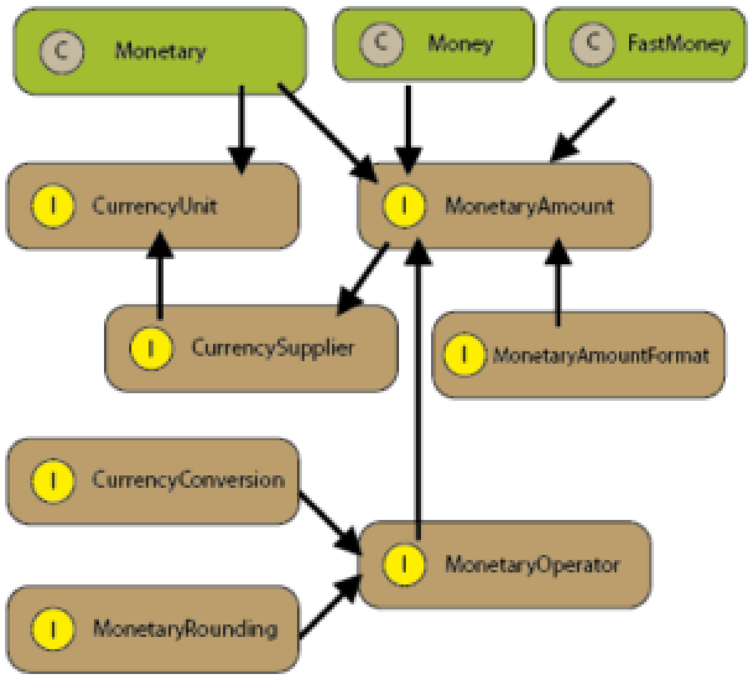
Setup

Link download: https://mvnrepository.com/artifact/org.javamoney/moneta

Or use Maven

```
<!-- https://mvnrepository.com/artifact/org.javamoney/moneta -->

<dependency>

    <groupId>org.javamoney</groupId>

    <artifactId>moneta</artifactId>

    <version>1.4.2</version>

    <type>pom</type>

</dependency>
```

Diagram Class and Interface of JSR354



CurrencyUnit and MonetaryAmount

CurrencyUnit models a currency. It is very similar to the existing java.util.Currency class, except it allows custom implementations. That get instance CurrencyUnit by MonetaryCurrencies factory.

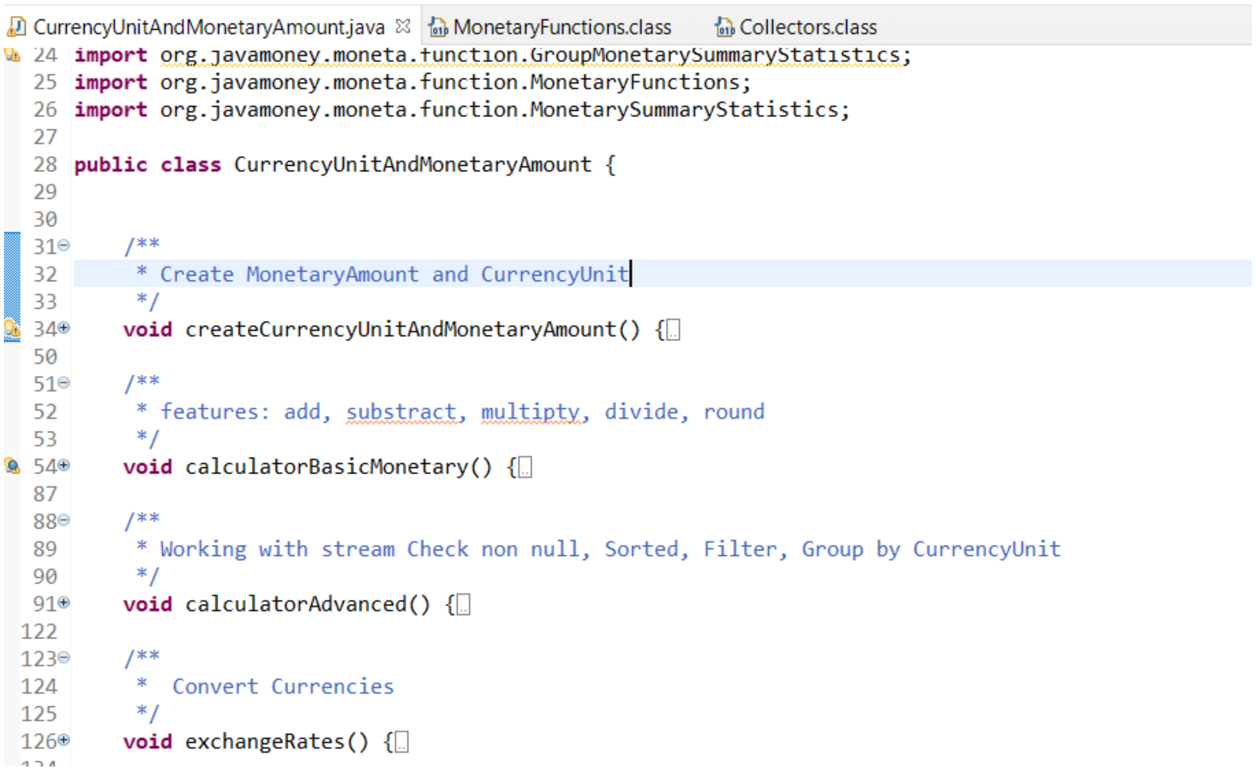
MonetaryAmount represents a concrete numeric representation of a monetary amount. A MonetaryAmount is always bound to a CurrencyUnit.

Money and FastMonney are two MonetaryAmount implementations of JavaMoney. Monet is default implementtation that stores number values using Bigdecimal. FastMoney is an alternative implementation which stores amounts in long fields.

According to documentation operation on FastMoney are 10-15 times faster compared to Money. However, FastMoney is limited by the size number and precision of the Long type. You can refer here about data type of java :

https://docs.oracle.com/javase/tutorial/java/nutsandbolts/datatypes.html

Demo



```java
import org.javamoney.moneta.function.GroupMonetarySummaryStatistics;
import org.javamoney.moneta.function.MonetaryFunctions;
import org.javamoney.moneta.function.MonetarySummaryStatistics;

public class CurrencyUnitAndMonetaryAmount {


    /**
     * Create MonetaryAmount and CurrencyUnit
     */
    void createCurrencyUnitAndMonetaryAmount() {

    /**
     * features: add, substract, multipty, divide, round
     */
    void calculatorBasicMonetary() {

    /**
     * Working with stream Check non null, Sorted, Filter, Group by CurrencyUnit
     */
    void calculatorAdvanced() {

    /**
     *  Convert Currencies
     */
    void exchangeRates() {
```

This section is non commercial mainly sharing and advance knowlage for java.This tutorials has referenced document from the list below if you has complain for license, i will remove all from internet. Thank you all everything.

https://download.oracle.com/otn-pub/jcp/money_currency-1_1-mrel-eval-spec/JavaMoneySpecification.pdf

https://www.baeldung.com/java-money-and-currency

https://dzone.com/articles/looking-java-9-money-and