

Краткая информация о реализации тестового задания

Задание:

Расчет прибыли (убытка) по валютному активу.

Приложение при запуске показывает окно (Swing) со следующими компонентами:

- Дата (Date) - дата покупки валюты
- Сумма в Долларах США (Amount USD) - количество купленных Долларов США
- Кнопка для запуска расчета (Recalculate)
- Не редактируемое поле для вывода полученной прибыли (убытка)

При нажатии на кнопку, приложение должно скачать курс обмена Доллара США на рубли из сервиса Fixer.io (см. <https://fixer.io/documentation>) или любого другого сервиса по загрузке данных по курсам валют в формате JSON на дату покупки и на текущую. Далее нужно рассчитать прибыль или убыток если бы валюту продали сегодня по текущему курсу. При этом нужно учитывать помимо разницы курсов еще и спред (spread), равный 0.5% Прибыль выводится в рублях.

Приложение должно представлять из себя Maven или Gradle проект (Gradle в приоритете).

В проекте допускается использование сторонних библиотек (например для работы с JSON или выполнения HTTP-запросов).

Используемые библиотеки и фреймворки

gson 2.8.0 - парсинг Json строки ответа сервера fixer.io

jdatepicker 2.0.3 - сторонний GUI компонент библиотеки Swing

spring-web 3.0.2.RELEASE - формирование и обработка HTTP-запросов

jgoodies-forms 1.8.0 - компоненты GUI

log4j 1.2.17 - логгирование приложения

Сборка проекта осуществляется при помощи **Maven**.

Например: `mvn package`

```
[INFO] --- maven-shade-plugin:1.6:shade (default) @ currency-conv ---
[INFO] Including com.google.code.gson:gson:jar:2.8.0 in the shaded jar.
[INFO] Including io.github.lzh0379:jdatepicker:jar:2.0.3 in the shaded jar.
[INFO] Including org.springframework:spring-web:jar:3.0.2.RELEASE in the shaded jar.
[INFO] Including aopalliance:aopalliance:jar:1.0 in the shaded jar.
[INFO] Including org.springframework:spring-beans:jar:3.0.2.RELEASE in the shaded jar.
[INFO] Including org.springframework:spring-context:jar:3.0.2.RELEASE in the shaded jar.
[INFO] Including org.springframework:spring-aop:jar:3.0.2.RELEASE in the shaded jar.
[INFO] Including org.springframework:spring-expression:jar:3.0.2.RELEASE in the shaded jar.
[INFO] Including org.springframework:spring-asm:jar:3.0.2.RELEASE in the shaded jar.
[INFO] Including org.springframework:spring-core:jar:3.0.2.RELEASE in the shaded jar.
[INFO] Including commons-logging:commons-logging:jar:1.1.1 in the shaded jar.
[INFO] Including com.jgoodies:jgoodies-forms:jar:1.8.0 in the shaded jar.
[INFO] Including com.jgoodies:jgoodies-common:jar:1.8.0 in the shaded jar.
[INFO] Including log4j:log4j:jar:1.2.17 in the shaded jar.
[INFO] Replacing original artifact with shaded artifact.
[INFO] Replacing C:\Java\workspace3\currency-conv\target\currency-conv-0.1-SNAPSHOT.jar with
conv\target\currency-conv-0.1-SNAPSHOT-shaded.jar
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 14.057 s
[INFO] Finished at: 2018-03-27T23:42:58+03:00
[INFO] -----
```

В проект включены плагины для формирования manifest и шейдинга classpath.

Запуск приложения:

java -jar %JarFolder%\currency-conv\target\currency-conv-0.1-SNAPSHOT.jar

Общий вид

The screenshot shows a Java Swing window titled "The Profit/Loss Calculator (fixer.io)". The window has a light gray background. At the top, there's a label "I've bought:" followed by a text input field, the text "EUR", the text "on", another text input field, and a small blue button with three dots. Below this, there's a label "And my current profit is:" followed by a text input field, the text "RUB", the text "Spread:", a text input field containing "0.05", and a small blue button with three dots. At the bottom left, there are two stacked buttons: "Verify" and "Clean All". At the bottom right, there is a large empty rectangular area for output.

После заполнения всех полей необходимо осуществить их валидацию нажатием кнопки **Verify**.

В случае успешной валидации станет доступна кнопка расчета прибыли/убытка:

The screenshot shows the same Java Swing window after validation. The input fields now contain "1250.55" and "20.03.2018". The "Spread" input field now contains "0.05" and is highlighted in green. The "Verify" button is still present, but the "Clean All" button is no longer visible. A new button labeled "Calculate profit" has appeared to the right of the "Verify" button. The large empty rectangular area at the bottom now contains the text "OK!".

По нажатию кнопки **Calculate profit** будет возвращено значение текущей прибыли:

The Profit/Loss Calculator (fixer.io)

I've bought: 1250.55 EUR on 20.03.2018

And my current profit is: 961.6001242207778 RUB Spread: 0.05

Verify
Clean All

Calculating...
Buy rate is: 70.462323
Current rate is: 71.302147
The core object has been created
Done!

Необходимо верифицировать поля перед каждым расчетом.

Верификация позволяет определить некорректную дату, сумму валюты и спрэд:

Введенные данные:

The Profit/Loss Calculator (fixer.io)

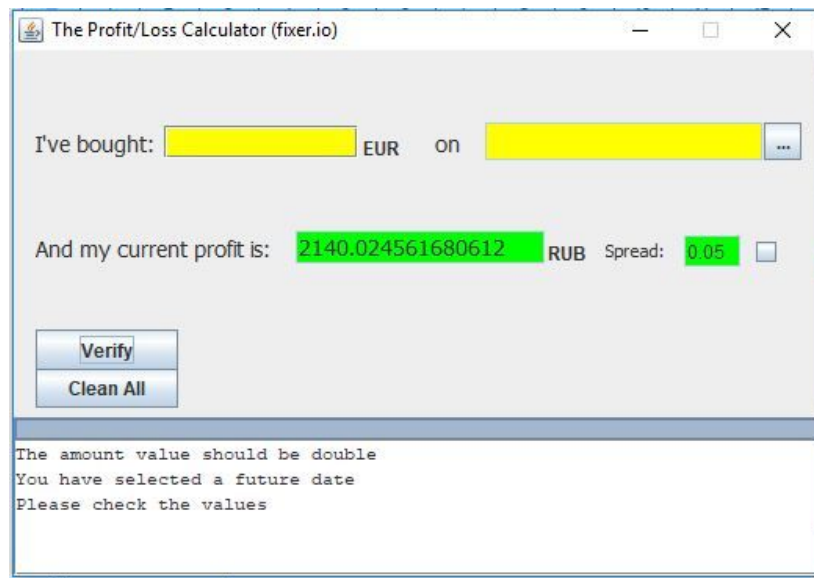
I've bought: ВАТНВ EUR on 28.12.2019

And my current profit is: 2140.024561680612 RUB Spread: 0.05

Verify
Clean All

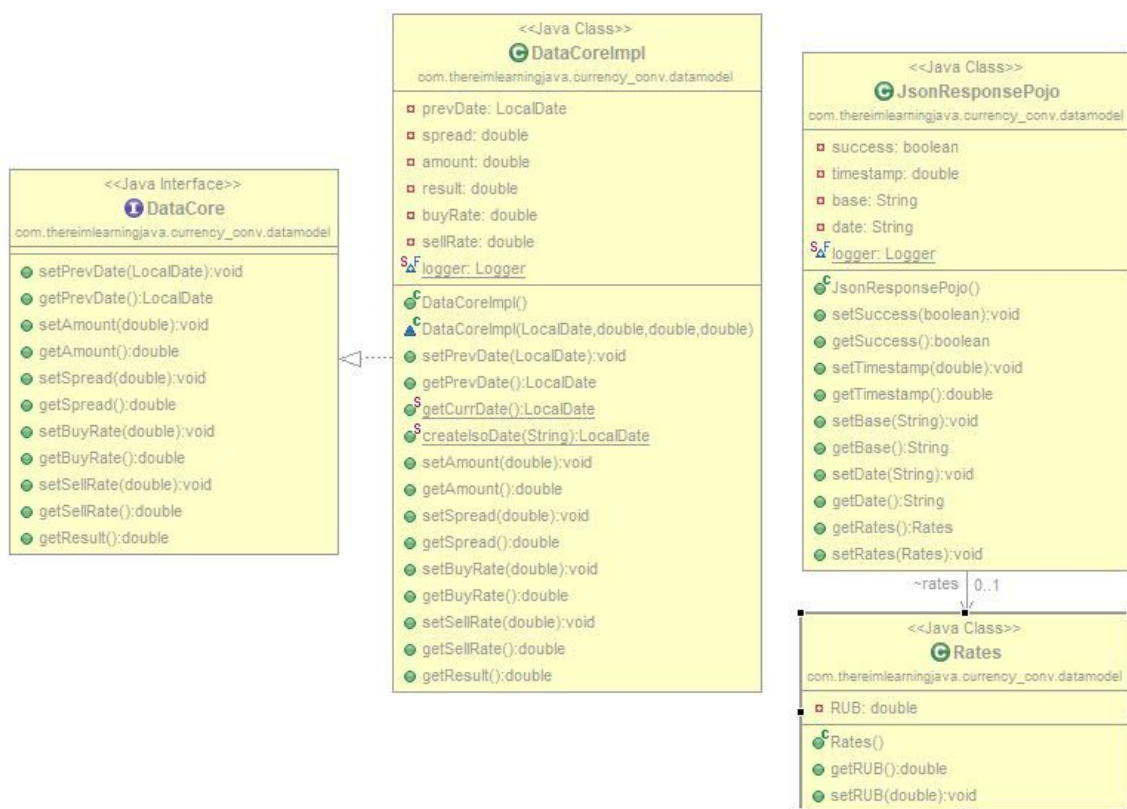
Buy rate is: 69.520469
Current rate is: 71.302147
The core object has been created
Done!

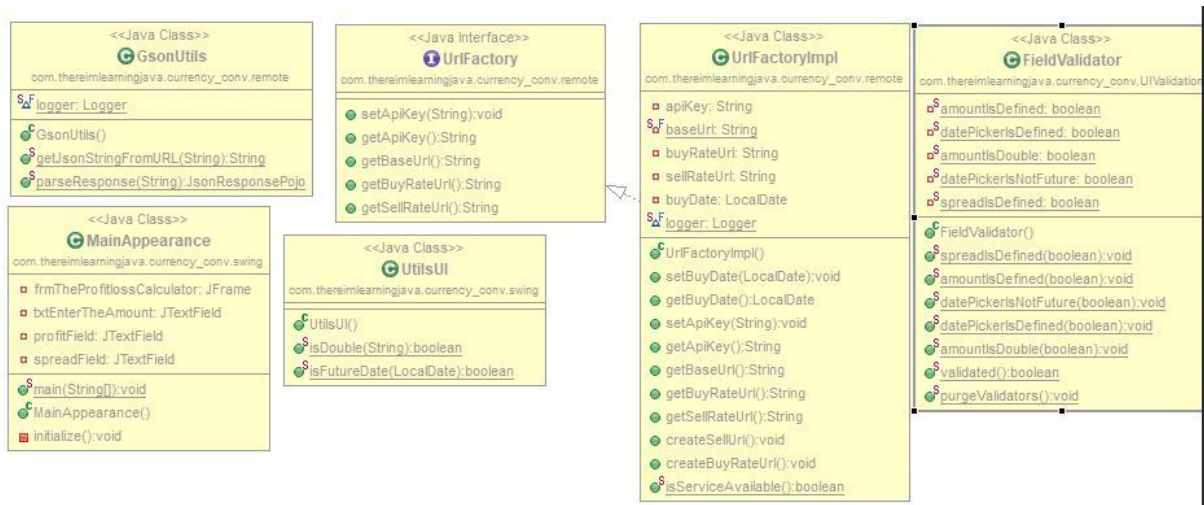
Результат верификации:



Лог-файл приложения находится в корневой папке, в конфигурации log4j по умолчанию указан аппендер DEBUG.

Диаграмма классов





Javadoc

Располагается в папке \doc проекта.

All Classes

Packages

com.thereimlearningjava.currency_conv
 com.thereimlearningjava.currency_conv.datamode
 com.thereimlearningjava.currency_conv.remote
 com.thereimlearningjava.currency_conv.swing

Classes

ConvertibleBlockImpl
 JsonHistPojo
 Rates

OVERVIEW PACKAGE CLASS USE TREE D

PREV CLASS NEXT CLASS FRAMES NO FI

SUMMARY: NESTED | FIELD | CONSTR | METHOD

com.thereimlearningjava.currency_conv.data

Class ConvertibleBlockImpl

java.lang.Object

com.thereimlearningjava.currency_conv.

```
public class ConvertibleBlockImpl
extends java.lang.Object
```

An implementation of main data core