HadoopCryptoLedger

Refactoring Code Smells

Name: Benny Daniel Tharigopala
Email ID: bn489600@dal.ca

Overview

The purpose of this activity is to eliminate code smells in the project – "HadoopCryptoLedger". The general process followed in this activity is as follows:

1. Use DesigniteJava (URL: DesigniteJava) to determine various types of smells present in the application. (OR) Analyze the packages for any violations of standard design principles and good practices.
2. Determine an appropriate refactoring technique to eliminate smells or resolve violations of principles.
3. Implement the technique.

1. Extract Method –

DesigniteJava Output before Refactoring:

```
Benny Daniel AV@Dannys-Desktop MINGW64 ~/Desktop/Dal Coursework/Winter_2022/CSCI
 5308/Assignments/A3_Backup
$ java -jar DesigniteJava.jar -i hadoopcryptoledger/ -o DesOut/
Searching classpath folders ...
Could not find any classpath folder.
Parsing the source code ...
Resolving symbols...
Computing metrics...
Detecting code smells...
Exporting analysis results...
wrapping up ...
--Analysis summary--
        Total LOC analyzed: 13200       Number of packages: 28
        Number of classes: 126  Number of methods: 1051
-Total architecture smell instances detected-
        Cyclic dependency: 0    God component: 0
        Ambiguous interface: 0  Feature concentration: 0
        Unstable dependency: 1  Scattered functionality: 0
        Dense structure: 0
-Total design smell instances detected-
        Imperative abstraction: 0       Multifaceted abstraction: 1
        Unnecessary abstraction: 0      Unutilized abstraction: 11
        Feature envy: 0 Deficient encapsulation: 12
        Unexploited encapsulation: 0    Broken modularization: 0
        Cyclically-dependent modularization: 1  Hub-like modularization: 0
        Insufficient modularization: 15 Broken hierarchy: 20
        Cyclic hierarchy: 0     Deep hierarchy: 0
        Missing hierarchy: 0    Multipath hierarchy: 0
        Rebellious hierarchy: 0 Wide hierarchy: 0
-Total implementation smell instances detected-
        Abstract function call from constructor: 0      Complex conditional: 10
        Complex method: 11      Empty catch clause: 0
        Long identifier: 42     Long method: 6
        Long parameter list: 13 Long statement: 775
        Magic number: 7389      Missing default: 3
----
Done.
```

DesigniteJava Output after Refactoring:

```
Benny Daniel AV@Dannys-Desktop MINGW64 ~/Desktop/Dal Coursework/Winter_2022/CSCI
 5308/Assignments/Assignment_3
$ java -jar DesigniteJava.jar -i hadoopcryptoledger/ -o DesOut/
Searching classpath folders ...
Parsing the source code ...
Resolving symbols...
Computing metrics...
Detecting code smells...
Exporting analysis results...
wrapping up ...
--Analysis summary--
        Total LOC analyzed: 13282       Number of packages: 28
        Number of classes: 126  Number of methods: 1057
-Total architecture smell instances detected-
        Cyclic dependency: 0    God component: 0
        Ambiguous interface: 0  Feature concentration: 0
        Unstable dependency: 1  Scattered functionality: 0
        Dense structure: 0
-Total design smell instances detected-
        Imperative abstraction: 0       Multifaceted abstraction: 1
        Unnecessary abstraction: 0      Unutilized abstraction: 11
        Feature envy: 0 Deficient encapsulation: 12
        Unexploited encapsulation: 0    Broken modularization: 0
        Cyclically-dependent modularization: 1  Hub-like modularization: 0
        Insufficient modularization: 15 Broken hierarchy: 20
        Cyclic hierarchy: 0     Deep hierarchy: 0
        Missing hierarchy: 0    Multipath hierarchy: 0
        Rebellious hierarchy: 0 Wide hierarchy: 0
-Total implementation smell instances detected-
        Abstract function call from constructor: 0      Complex conditional: 10
        Complex method: 11      Empty catch clause: 0
        Long identifier: 42     Long method: 4
        Long parameter list: 13 Long statement: 775
        Magic number: 7389      Missing default: 3
----
Done.
```

Description of Change made:

The tool detected the smell in this class because the following methods had 169 lines of code resulting in a long method:

1. parseBlock1346406AsEthereumBlockHeap
2. parseBlock1346406AsEthereumBlockDirect

The long methods were refactored by extracting each method into new ones. The following methods now represent the aforementioned 2 methods:

1. parseBlock1346406AsEthereumBlockHeap()
2. parseBlock1346406AsEthereumBlockHeapBlockChecks()
3. parseBlock1346406AsEthereumBlockHeapTransactionChecks0to2()
4. parseBlock1346406AsEthereumBlockHeapTransactionChecks3to5() &

5. parseBlock1346406AsEthereumBlockDirect()
6. parseBlock1346406AsEthereumBlockDirectBlockChecks()
7. parseBlock1346406AsEthereumBlockDirectTransactionChecks0to2()
8. parseBlock1346406AsEthereumBlockDirectTransactionChecks3to5()

A total of **6 methods** were labelled as "Long methods" by DesigniteJava. Only two methods were refactored and the other 4 were untouched. Therefore, the number of Long Methods, under Implementation smells is four and not zero since the following 4 methods have 105 lines of code.

1. public void parseBlock0to10AsEthereumBlockHeap()
2. public void parseBlock0to10AsEthereumBlockDirect()
3. parseBlock3510000to3510010AsEthereumBlockHeap()
4. parseBlock3510000to3510010AsEthereumBlockDirect()

However, these methods were not refactored into multiple methods, since, the blocks inside these methods are tested in **sequence** by the **EthereumBlock.readBlock()** method, and therefore cannot be split. Therefore, these 4 methods were left untouched.

```
@Test
public void parseBlock0to10AsEthereumBlockHeap() throws IOException, EthereumBlockReadException {...}

@Test
public void parseBlock0to10AsEthereumBlockDirect() throws IOException, EthereumBlockReadException {...}

@Test
public void parseBlock3510000to3510010AsEthereumBlockHeap() throws IOException, EthereumBlockReadException {...}

@Test
public void parseBlock3510000to3510010AsEthereumBlockDirect() throws IOException, EthereumBlockReadException {...}
```

```
assertEquals( expected: 15, eTransactions.size(), message: "Block 3510000 contains 15 transactions");
assertEquals( expected: 0, eUncles.size(), message: "Block 3510000 contains 0 uncleHeaders");
byte[] expectedParentHash = new byte[] {(byte)0x63,(byte)0x74,(byte)0x6f,(byte)0x5b,(byte)0xcf,(byte)0xa3,(by
assertArrayEquals( expectedParentHash, eblockHeader.getParentHash(), message: "Block 3510000 contains a correct
eblock = ebr.readBlock();
eblockHeader = eblock.getEthereumBlockHeader();
eTransactions = eblock.getEthereumTransactions();
eUncles = eblock.getUncleHeaders();
assertEquals( expected: 0, eTransactions.size(), message: "Block 3510001 contains 0 transactions");
assertEquals( expected: 0, eUncles.size(), message: "Block 3510001 contains 0 uncleHeaders");
```

Relevant Files & Links:

Project	Package	File	Method	Line #
hadoopcryptoledger	org.zuinnote.hadoop.ethereum.format.common	EthereumFormatReaderTest	**parseBlock1346406AsEthereumBlockHeap**	**713**
hadoopcryptoledger	org.zuinnote.hadoop.ethereum.format.common	EthereumFormatReaderTest	**parseBlock1346406AsEthereumBlockDirect**	**904**

Branch: refactoring_implementation_smells

Commit#: 8683a997bac22c51568337326664e1a52b59eeb1

2. Rename Method / Variable –

Description of Change made:

- The byte array - “TEST_RLP_LIST_LARGELIST” was renamed to “TEST_RLP_LIST_LARGE_LIST” to remove a typographical error.

Before:

```
48      public final static byte[] TEST_RLP_EMPTY_LIST = new byte[]{(byte) 0xc0};
49      public final static byte[] TEST_RLP_ELEMENT_INT_15 = new byte[] {0x0f};
50      public final static byte[] TEST_RLP_ELEMENT_INT_1024 = new byte[] {(byte) 0x82,0x04,0x00};
51      public final static byte[] TEST_RLP_LIST_SET3 = new byte[] {(byte) 0xc7,(byte) 0xc0,(byte) 0xc1,(byte) 0xc0,(byt
52      public final static byte[] TEST_RLP_ELEMENT_LARGESTRING = new byte[] {(byte) 0xb8,0x4E, 'a','b','c','d','e','f',
53      public final static byte[] TEST_RLP_LIST_LARGELIST = new byte[] {(byte) 0xf8,0x50,(byte) 0xb8,0x4E, 'a','b','c',
54
55       @Test
56        public void checkTestDataBlock1346406Avail
57          ClassLoader classLoader = getClass().getClassLoader();
58          String fileName="eth1346406.bin";
59          String fileNameGenesis=classLoader.getResource( name: "testdata/"+fileName).getFile();
60
rors
```

Typo: In word 'LARGELIST'

Typo: Rename to... Alt+Shift+Enter More actions... Alt+Enter

After:

```
49      public final static byte[] TEST_RLP_ELEMENT_INT_15 = new byte[] {0x0f};
50      public final static byte[] TEST_RLP_ELEMENT_INT_1024 = new byte[] {(byte) 0x82,0x04,0x00};
51      public final static byte[] TEST_RLP_LIST_SET3 = new byte[] {(byte) 0xc7,(byte) 0xc0,(byte) 0xc1,(byte) 0xc0,(byte) 0xc3
52      public final static byte[] TEST_RLP_ELEMENT_LARGESTRING = new byte[] {(byte) 0xb8,0x4E, 'a','b','c','d','e','f','g','h'
53      public final static byte[] TEST_RLP_LIST_LARGE_LIST = new byte[] {(byte) 0xf8,0x50,(byte) 0xb8,0x4E, 'a','b','c',
54
```

- The String - “CONF_ISSPLITABLE” was renamed to “CONF_IS_SPLITABLE” to remove a typographical error.
- The String - “DEFAULT_ISSPLITABLE” was renamed to “DEFAULT_IS_SPLITABLE” to remove a typographical error.

Before:

```
EthereumUtilTest.java   AbstractBitcoinFileInputFormat.java
DEFAULT_IS_SPLITABLE                                   0 results

public abstract class AbstractBitcoinFileInputFormat<K,V> extends FileInputFormat<K,V> implements JobConfigurable  {
public static final String CONF_ISSPLITABLE=org.zuinnote.hadoop.bitcoin.format.mapreduce.AbstractBitcoinFileInputFormat.CONF_ISSPL
public static final boolean DEFAULT_ISSPLITABLE=org.zuinnote.hadoop.bitcoin.format.mapreduce.AbstractBitcoinFileInputFormat.DEFAUL

Typo: In word 'ISSPLITABLE'
Typo: Rename to...  Alt+Shift+Enter    More actions...  Alt+Enter

private boolean isSplitable=DEFAULT_ISSPLI
private CompressionCodecFactory compressionCodecs = null;
```

After:

```
EthereumUtilTest.java   AbstractBitcoinFileInputFormat.java
import ...

public abstract class AbstractBitcoinFileInputFormat<K,V> extends FileInputFormat<K,V> implements JobConfigurable  {
public static final String CONF_IS_SPLITTABLE =org.zuinnote.hadoop.bitcoin.format.mapreduce.AbstractBitcoinFileInputFormat.CONF_ISSPLITABLE;
public static final boolean DEFAULT_IS_SPLITTABLE =org.zuinnote.hadoop.bitcoin.format.mapreduce.AbstractBitcoinFileInputFormat.DEFAULT_ISSPLITA
```

Relevant Files & Links:

Project	Package	File	Line#
hadoopcryptoledger	org.zuinnote.hadoop.ethereum.format.common	EthereumUtilDecodeTest	**53**
hadoopcryptoledger	org.zuinnote.hadoop.ethereum.format.common	EthereumUtilBockTest	**32**

hadoopcryptoledger	org.zuinnote.hadoop.bitcoin.format.mapred	AbstractBitcoinFileInputFormat	**38 & 39**
hadoopcryptoledger	org.zuinnote.hadoop.bitcoin.format.mapreduce	AbstractBitcoinFileInputFormat	**36 & 37**

Branch: refactoring_implementation_smells

Commit#: 3e54e78c2113674325e0272ea732aff7527b3635

3. Change bidirectional association to unidirectional association

DesigniteJava Output before Refactoring:

```
Benny Daniel AV@Dannys-Desktop MINGW64 ~/Desktop/Dal Coursework/Winter_2022/CSCI
 5308/Assignments/A3_Backup
$ java -jar DesigniteJava.jar -i hadoopcryptoledger/ -o DesOut/
Searching classpath folders ...
Could not find any classpath folder.
Parsing the source code ...
Resolving symbols...
Computing metrics...
Detecting code smells...
Exporting analysis results...
wrapping up ...
--Analysis summary--
        Total LOC analyzed: 13200       Number of packages: 28
        Number of classes: 126  Number of methods: 1051
-Total architecture smell instances detected-
        Cyclic dependency: 0    God component: 0
        Ambiguous interface: 0  Feature concentration: 0
        Unstable dependency: 1  Scattered functionality: 0
        Dense structure: 0
-Total design smell instances detected-
        Imperative abstraction: 0       Multifaceted abstraction: 1
        Unnecessary abstraction: 0      Unutilized abstraction: 11
        Feature envy: 0 Deficient encapsulation: 12
        Unexploited encapsulation: 0    Broken modularization: 0
        Cyclically-dependent modularization: 1  Hub-like modularization: 0
        Insufficient modularization: 15 Broken hierarchy: 20
        Cyclic hierarchy: 0     Deep hierarchy: 0
        Missing hierarchy: 0    Multipath hierarchy: 0
        Rebellious hierarchy: 0 Wide hierarchy: 0
-Total implementation smell instances detected-
        Abstract function call from constructor: 0      Complex conditional: 10
        Complex method: 11      Empty catch clause: 0
        Long identifier: 42     Long method: 6
        Long parameter list: 13 Long statement: 775
        Magic number: 7389      Missing default: 3
----
Done.
```

DesigniteJava Output after Refactoring:

```
Benny Daniel AV@Dannys-Desktop MINGW64 ~/Desktop/Dal Coursework/Winter_2022/CSCI
 5308/Assignments/Assignment_3
$ java -jar DesigniteJava.jar -i hadoopcryptoledger/ -o DesOut/
Searching classpath folders ...
Parsing the source code ...
Resolving symbols...
Computing metrics...
Detecting code smells...
Exporting analysis results...
wrapping up ...
--Analysis summary--
        Total LOC analyzed: 13214       Number of packages: 28
        Number of classes: 127  Number of methods: 1053
-Total architecture smell instances detected-
        Cyclic dependency: 0    God component: 0
        Ambiguous interface: 0  Feature concentration: 0
        Unstable dependency: 1  Scattered functionality: 0
        Dense structure: 0
-Total design smell instances detected-
        Imperative abstraction: 0       Multifaceted abstraction: 1
        Unnecessary abstraction: 0      Unutilized abstraction: 11
        Feature envy: 0 Deficient encapsulation: 12
        Unexploited encapsulation: 0    Broken modularization: 0
        Cyclically-dependent modularization: 0  Hub-like modularization: 0
        Insufficient modularization: 15 Broken hierarchy: 20
        Cyclic hierarchy: 0     Deep hierarchy: 0
        Missing hierarchy: 0    Multipath hierarchy: 0
        Rebellious hierarchy: 0 Wide hierarchy: 0
-Total implementation smell instances detected-
        Abstract function call from constructor: 0      Complex conditional: 10
        Complex method: 11      Empty catch clause: 0
        Long identifier: 42     Long method: 6
        Long parameter list: 13 Long statement: 775
        Magic number: 7389      Missing default: 3
----
Done.
```

Description of Change made:

The tool detected the smell in this class because this class participates in a cyclic dependency.
The participating classes in the cycle are: EthereumUtil & EthereumTransaction

The cyclic dependency was resolved by introducing an interface into the equation. **EthereumUtil** now implements the interface – **EthereumTransactionInterface,** and **EthereumTransaction** is dependent on the interface instead of EthereumUtil.

Relevant Files & Links:

Project	Package	File	Line#
hadoopcryptoledger	org.zuinnote.hadoop.ethereum.format.common	EthereumUtil	**48**
hadoopcryptoledger	org.zuinnote.hadoop.ethereum.format.common	EthereumTransaction	**59, 79 & 90**
hadoopcryptoledger	org.zuinnote.hadoop.ethereum.format.common	EthereumTransactionInterface	**New Interface**

Branch: refactoring_design_smells

OLD Commit#: 3162dbfb08ca1a80c1079df7f7982e8ef4be2882
Commit#: bd7fe898b4d1130eed6f02a58ff720196d2d846a

4. Extract Class

DesigniteJava Output before Refactoring:

```
Benny Daniel AV@Dannys-Desktop MINGW64 ~/Desktop/Dal Coursework/Winter_2022/CSCI
 5308/Assignments/A3_Backup
$ java -jar DesigniteJava.jar -i hadoopcryptoledger/ -o DesOut/
Searching classpath folders ...
Could not find any classpath folder.
Parsing the source code ...
Resolving symbols...
Computing metrics...
Detecting code smells...
Exporting analysis results...
wrapping up ...
--Analysis summary--
        Total LOC analyzed: 13200       Number of packages: 28
        Number of classes: 126  Number of methods: 1051
-Total architecture smell instances detected-
        Cyclic dependency: 0    God component: 0
        Ambiguous interface: 0  Feature concentration: 0
        Unstable dependency: 1  Scattered functionality: 0
        Dense structure: 0
-Total design smell instances detected-
        Imperative abstraction: 0       Multifaceted abstraction: 1
        Unnecessary abstraction: 0      Unutilized abstraction: 11
        Feature envy: 0 Deficient encapsulation: 12
        Unexploited encapsulation: 0    Broken modularization: 0
        Cyclically-dependent modularization: 1  Hub-like modularization: 0
        Insufficient modularization: 15 Broken hierarchy: 20
        Cyclic hierarchy: 0     Deep hierarchy: 0
        Missing hierarchy: 0    Multipath hierarchy: 0
        Rebellious hierarchy: 0 Wide hierarchy: 0
-Total implementation smell instances detected-
        Abstract function call from constructor: 0      Complex conditional: 10
        Complex method: 11      Empty catch clause: 0
        Long identifier: 42     Long method: 6
        Long parameter list: 13 Long statement: 775
        Magic number: 7389      Missing default: 3
----
Done.
```

DesigniteJava Output after Refactoring:

```
Benny Daniel AV@Dannys-Desktop MINGW64 ~/Desktop/Dal Coursework/Winter_2022/CSCI 5308/Assignments/Assignment_3
$ java -jar DesigniteJava.jar -i hadoopcryptoledger/ -o DesOut/
Searching classpath folders ...
Parsing the source code ...
Resolving symbols...
Computing metrics...
Detecting code smells...
Exporting analysis results...
wrapping up ...
--Analysis summary--
        Total LOC analyzed: 13216       Number of packages: 29
        Number of classes: 129  Number of methods: 1053
-Total architecture smell instances detected-
        Cyclic dependency: 0    God component: 0
        Ambiguous interface: 0  Feature concentration: 0
        Unstable dependency: 1  Scattered functionality: 0
        Dense structure: 0
-Total design smell instances detected-
        Imperative abstraction: 0       Multifaceted abstraction: 0
        Unnecessary abstraction: 0      Unutilized abstraction: 11
        Feature envy: 0 Deficient encapsulation: 12
        Unexploited encapsulation: 0    Broken modularization: 0
        Cyclically-dependent modularization: 0  Hub-like modularization: 0
        Insufficient modularization: 15 Broken hierarchy: 20
        Cyclic hierarchy: 0     Deep hierarchy: 0
        Missing hierarchy: 0    Multipath hierarchy: 0
        Rebellious hierarchy: 0 Wide hierarchy: 0
-Total implementation smell instances detected-
        Abstract function call from constructor: 0      Complex conditional: 10
        Complex method: 11      Empty catch clause: 0
        Long identifier: 42     Long method: 6
        Long parameter list: 13 Long statement: 775
        Magic number: 7395      Missing default: 3
----
Done.
```

Description of Change made:

The tool detected a “Multifaceted abstraction” smell in this class because the cohesion among the methods of this class is low. The Lack of Cohesion among methods (LCOM) of this class is: 0.857. The participating class is EthereumUtilTest.

The smell was resolved by refactoring the class into 3 cohesive classes, namely – EthereumUtilBlockTest, EthereumUtilEncodeTest and EthereumUtilDecodeTest.

EthereumUtilBlockTest contained test methods - checkTestDataBlock1346406Available, calculateChainIdBlock1346406(), getTransActionHashBlock1346406() and getTransActionSendAddressBlock1346406().

EthereumUtilEncodeTest and EthereumUtilDecodeTest now contain Test methods for Encode and Decode methods, respectively.

Relevant Files & Links:

Project	Package	File	Line#
hadoopcryptoledger	org.zuinnote.hadoop.ethereum.format.common	EthereumUtilTest	**Class Extracted**
hadoopcryptoledger	org.zuinnote.hadoop.ethereum.format.common	EthereumUtilBlockTest	**New Class**
hadoopcryptoledger	org.zuinnote.hadoop.ethereum.format.common	EthereumUtilEncodeTest	**New Class**
hadoopcryptoledger	org.zuinnote.hadoop.ethereum.format.common	EthereumUtilEncodeTest	**New Class**

Branch: refactoring_design_smells

OLD Commit #: 0d26ba2626945a50317c6a1d4efdf777605e38e0
Commit #: f682830b65bd34f438d48d00dd4a7c507ac4f570

5. Pull-Up Method

Description of Change made:
This refactoring technique was employed to remove code duplication in sub-classes.
The participating classes are: **BitcoinBlockFlinkInputFormat**, **BitcoinRawBlockFlinkInputFormat** & **BitcoinTransactionFlinkInputFormat**.

The method which's duplicated across all these classes is: **reachedEnd()**

The duplication in code was resolved by pulling the method up to the Abstract class – **AbstractBitcoinFlinkInputFormat** which is extended by all 3 of the aforementioned sub-classes.

Relevant Files & Links:

Project	Package	File	Line#
hadoopcryptoledger	org.zuinnote.flink.bitcoin	AbstractBitcoinFlinkInputFormat	**102**
hadoopcryptoledger	org.zuinnote.flink.bitcoin	BitcoinBlockFlinkInputFormat	**56**
hadoopcryptoledger	org.zuinnote.flink.bitcoin	BitcoinRawBlockFlinkInputFormat	**57**
hadoopcryptoledger	org.zuinnote.flink.bitcoin	BitcoinTransactionFlinkInputFormat	**61**

Branch: refactoring_implementation_smells

Commit#: 2da51bffaa6e3b19993f23cf62d26c0f877ff008

URL to GitHub Issue: https://github.com/ZuInnoTe/hadoopcryptoledger/issues/87

Pull Request URL: https://github.com/ZuInnoTe/hadoopcryptoledger/pull/88

Citations

[1] "A Taxonomy of Software Smells", *Tusharma.in*, 2022. [Online]. Available: https://tusharma.in/smells/. [Accessed: 21- Mar-2022].

[2] “Refactoring and Design Patterns,” *refactoring.guru*. [Online]. Available: https://refactoring.guru/. [Accessed: 22-Mar-2022]