



Module	Assessment Type
Distributed and Cloud Systems Programming	Individual Report

Workshop 10

Student Id : 2049867
Student Name : Roshan Parajuli
Section : L5CG3
Module Leader : Rupak Koirala
Lecturer : Saroj Sharma
Submitted on : 2021-05-14

Table of Contents

Introduction.....2

Workshop task2

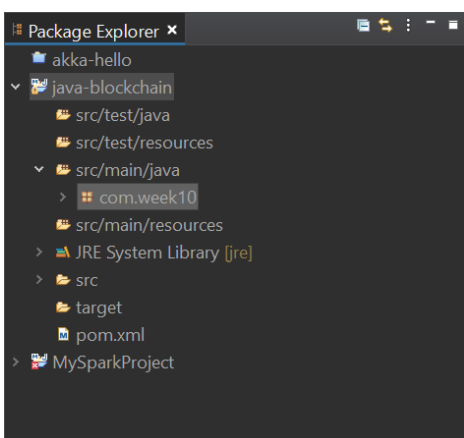
Conclusion5

Introduction

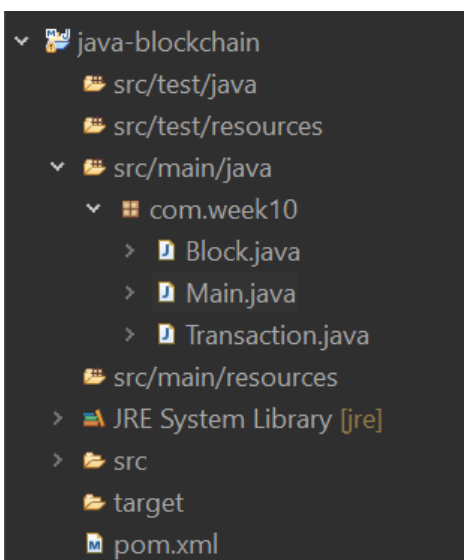
This workshop deals with the simple task of creating a blockchain in java. Blockchain is a digital ledger of transactions that holds sets of information on the structure known as blocks. It is decentralized and distributed. It has various use cases including healthcare, voting, as a digital currency and so on.

Workshop task

First, a new project was created in Eclipse. It was named “java-blockchain”. It was populated with the required files by eclipse. A new package was created inside src/main/java and named com.week10. It can be seen in the screenshot of the package explorer below:



After the project was created, three files were created namely: Block.java, Main.java and Transaction.java as shown below.



The coding part is now ready to begin with.

Transaction class

```
java-blockchain/pom.xml  Block.java  Transaction.java x  Main.java
1 package com.week10;
2
3 public class Transaction {
4
5     private String sourceName;
6     private String destinationName;
7     private Long sum;
8     public Transaction(String sourceName, String destinationName, Long sum) {
9         super();
10        this.sourceName = sourceName;
11        this.destinationName = destinationName;
12        this.sum = sum;
13    }
14    public String getSourceName() {
15        return sourceName;
16    }
17    public void setSourceName(String sourceName) {
18        this.sourceName = sourceName;
19    }
20    public String getDestinationName() {
21        return destinationName;
22    }
23    public void setDestinationName(String destinationName) {
```

In the transaction class, three variables are declared as a transaction requires three things: the source from which the transaction initiates, the destination where the transaction is headed to and the sum of amount that is being transferred from one account to another.

Getters and setters are made for all the variables to access them from other classes. hashCode() and equals() function is generated so that the transaction would retain the same hash everytime and not changed on every compilation.

Block class

As a block contains the hash of the previous class and the list of transactions in itself, this class has two variables defined for the same. Like the transaction class, it also has the getters and setters defined as the variables are private. Also it contains the hashCode() and the equals() method which are generated from the eclipse IDE.

```

1 package com.week10;
2
3 import java.util.List;
4
5 public class Block {
6
7     private int previousHash;
8     private List<Transaction> transactions;
9
10    public Block(int previousHash, List<Transaction> transactions) {
11        super();
12        this.previousHash = previousHash;
13        this.transactions = transactions;
14    }
15
16    public int getPreviousHash() {
17        return previousHash;
18    }
19
20    public void setPreviousHash(int previousHash) {
21        this.previousHash = previousHash;
22    }
23

```

Main class

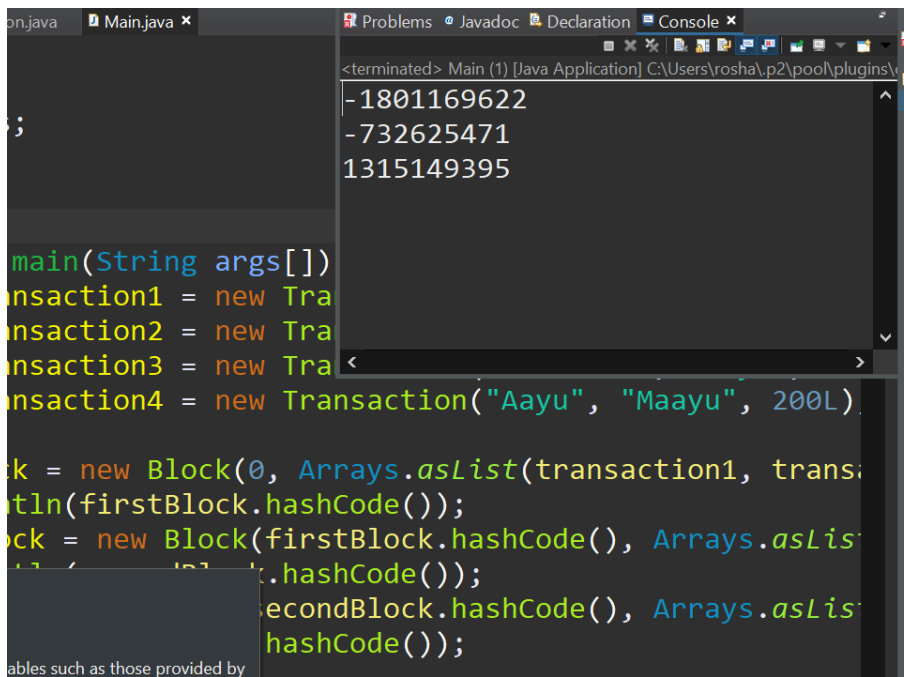
This class is responsible for the creating the transactions by making the objects of the transaction class and initializing the constructors with two parameters. Here, three blocks are made and the hash code of all the three blocks are printed in the screen.

```

1 package com.week10;
2
3 import java.util.Arrays;
4
5 public class Main {
6
7    public static void main(String args[]) {
8        Transaction transaction1 = new Transaction("Roshan", "Prakriti", 1000);
9        Transaction transaction2 = new Transaction("Prakriti", "Roshbot", 2000);
10       Transaction transaction3 = new Transaction("Prakriti", "Aayu", 2000);
11       Transaction transaction4 = new Transaction("Aayu", "Maayu", 2000);
12
13       Block firstBlock = new Block(0, Arrays.asList(transaction1, transaction2));
14       System.out.println(firstBlock.hashCode());
15       Block secondBlock = new Block(firstBlock.hashCode(), Arrays.asList(transaction3, transaction4));
16       System.out.println(secondBlock.hashCode());
17       Block thirdBlock = new Block(secondBlock.hashCode(), Arrays.asList(transaction1, transaction2));
18       System.out.println(thirdBlock.hashCode());
19
20    }
21 }
22

```

Upon running the main class, the following output is noticed.



The screenshot shows the Eclipse IDE with a Java project. The 'Main.java' file is open, and the 'Console' view is active. The code in 'Main.java' defines a 'Transaction' class and a 'Block' class. The 'main' method creates four transactions and two blocks. The console output shows the hash codes of the first and second blocks, which are -1801169622 and -732625471 respectively. The third block's hash code is 1315149395.

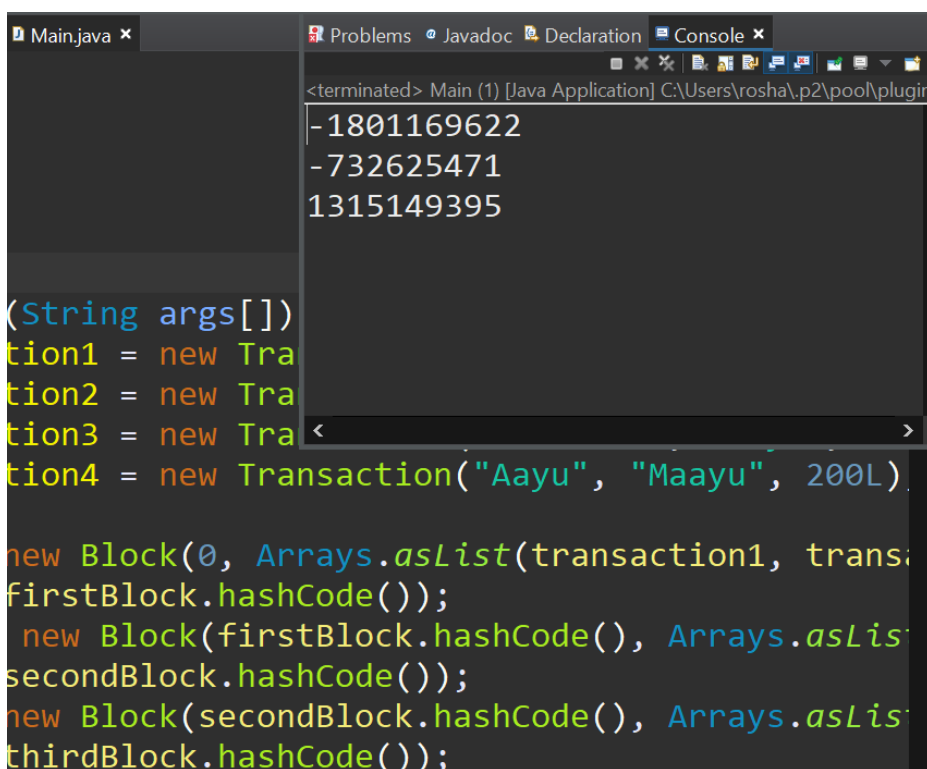
```
on.java Main.java x Problems Javadoc Declaration Console x
<terminated> Main (1) [Java Application] C:\Users\rosha\p2\pool\plugins\
-1801169622
-732625471
1315149395

;

main(String args[])
transaction1 = new Tra
transaction2 = new Tra
transaction3 = new Tra
transaction4 = new Transaction("Aayu", "Maayu", 200L).

Block = new Block(0, Arrays.asList(transaction1, transa
println(firstBlock.hashCode());
Block = new Block(firstBlock.hashCode(), Arrays.asList
transaction1, transaction2).hashCode());
println(secondBlock.hashCode(), Arrays.asList
transaction1, transaction2, transaction3).hashCode());
```

If the program is re-run without changing any part of it, it can be observed that the hash code does not change.



This screenshot is identical to the one above, showing the same Java code and console output in the Eclipse IDE. The console output shows the hash codes of the first and second blocks, which are -1801169622 and -732625471 respectively. The third block's hash code is 1315149395.

```
Main.java x Problems Javadoc Declaration Console x
<terminated> Main (1) [Java Application] C:\Users\rosha\p2\pool\plugins\
-1801169622
-732625471
1315149395

(String args[])
transaction1 = new Tra
transaction2 = new Tra
transaction3 = new Tra
transaction4 = new Transaction("Aayu", "Maayu", 200L).

new Block(0, Arrays.asList(transaction1, transa
firstBlock.hashCode());
new Block(firstBlock.hashCode(), Arrays.asList
secondBlock.hashCode());
new Block(secondBlock.hashCode(), Arrays.asList
thirdBlock.hashCode());
```

Conclusion

In this workshop, a custom implementation of block chain was created in java with the usage of three classes namely block, transaction and a main class in eclipse.