Practical No.5

Name: Bhairavi Narendra Rewatkar

Roll No.: DMET1221006

Subject: Blockchain Technology Laboratory

Date: 09/12/2024

Title: Adding Transactions to the blocks in blockchain

Aim: Write a program to implement add transactions to the blocks in the blockchain.

Source Code:

Transaction.java

```
public class Transaction {
    private String sender;
    private String receiver;
    private double amount;

    public Transaction(String sender, String receiver, double amount) {
        this.sender = sender;
        this.receiver = receiver;
        this.amount = amount;
    }

    @Override
    public String toString() {
        return "Transaction{" +
                "sender='" + sender + '\'' +
                ", receiver='" + receiver + '\'' +
                ", amount=" + amount +
                '}';
    }
}
```

Block.java

```
import java.util.ArrayList;
import java.util.Date;
import java.util.List;

public class Block {
    public String hash;
    public String previousHash;
    private long timestamp;
    private List<Transaction> transactions;
    public int nonce;

    public Block(String previousHash, List<Transaction> transactions) {
        this.previousHash = previousHash;
        this.transactions = transactions != null ? transactions : new ArrayList<>();
        this.timestamp = new Date().getTime();
        this.hash = calculateHash();
    }

    public String calculateHash() {
        return HashUtil.applySHA256(previousHash + Long.toString(timestamp) +
transactions.toString() + nonce);
    }

    public void mineBlock(int difficulty) {
        String target = new String(new char[difficulty]).replace('\0', '0'); // Target hash prefix
        while (!hash.substring(0, difficulty).equals(target)) {
            nonce++;
            hash = calculateHash();
        }
        System.out.println("Block Mined: " + hash);
    }

    @Override
```

```
    public String toString() {
        return "Block{" +
                "previousHash='" + previousHash + '\'' +
                ", hash='" + hash + '\'' +
                ", transactions=" + transactions +
                ", nonce=" + nonce +
                '}';
    }
}
```

Blockchain.java

```
import java.util.ArrayList;
import java.util.List;

public class Blockchain {
    public ArrayList<Block> chain;
    public List<Transaction> transactionPool;

    public Blockchain() {
        chain = new ArrayList<>();
        transactionPool = new ArrayList<>();
        chain.add(createGenesisBlock());
    }

    private Block createGenesisBlock() {
        return new Block("0", new ArrayList<>());
    }

    public void addTransaction(String sender, String receiver, double amount) {
        Transaction newTransaction = new Transaction(sender, receiver, amount);
        transactionPool.add(newTransaction);
    }
```

```
    public void mineBlock(int difficulty) {
        Block newBlock = new Block(chain.get(chain.size() - 1).hash, new
ArrayList<>(transactionPool));
        newBlock.mineBlock(difficulty);
        chain.add(newBlock);
        transactionPool.clear(); // Clear the transaction pool after mining
    }

    public void printBlockchain() {
        for (Block block : chain) {
            System.out.println(block);
        }
    }

    public static void main(String[] args) {
        Blockchain blockchain = new Blockchain();

        // Adding transactions
        blockchain.addTransaction("Alice", "Bob", 100);
        blockchain.addTransaction("Bob", "Charlie", 50);

        // Mining the first block
        blockchain.mineBlock(4);

        // Adding more transactions
        blockchain.addTransaction("Charlie", "Dave", 200);
        blockchain.addTransaction("Dave", "Alice", 150);

        // Mining the second block
        blockchain.mineBlock(4);

        // Print the blockchain
        blockchain.printBlockchain();
```

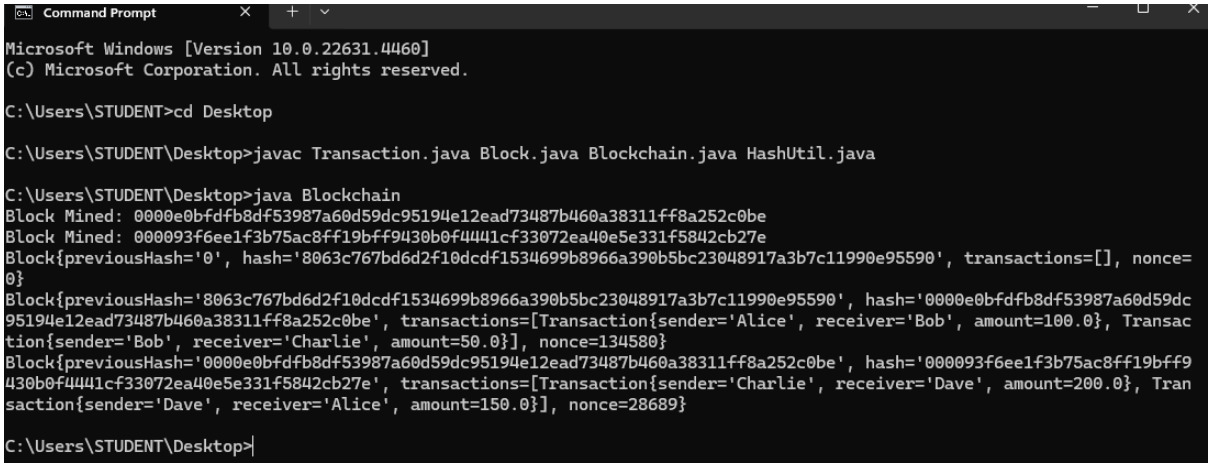
```
    }
}
```

HashUtil.java

```
import java.security.MessageDigest;


public class HashUtil {
    public static String applySHA256(String input) {
        try {
            MessageDigest digest = MessageDigest.getInstance("SHA-256");
            byte[] hash = digest.digest(input.getBytes("UTF-8"));
            StringBuilder hexString = new StringBuilder();
            for (byte b : hash) {
                String hex = Integer.toHexString(0xff & b);
                if (hex.length() == 1) hexString.append('0');
                hexString.append(hex);
            }
            return hexString.toString();
        } catch (Exception e) {
            throw new RuntimeException(e);
        }
    }
}
```

Output:



```
Command Prompt

Microsoft Windows [Version 10.0.22631.4460]
(c) Microsoft Corporation. All rights reserved.

C:\Users\STUDENT>cd Desktop

C:\Users\STUDENT\Desktop>javac Transaction.java Block.java Blockchain.java HashUtil.java

C:\Users\STUDENT\Desktop>java Blockchain
Block Mined: 0000e0bfdfb8df53987a60d59dc95194e12ead73487b460a38311ff8a252c0be
Block Mined: 000093f6ee1f3b75ac8ff19bff9430b0f4441cf33072ea40e5e331f5842cb27e
Block{previousHash='0', hash='8063c767bd6d2f10dcdf1534699b8966a390b5bc23048917a3b7c11990e95590', transactions=[], nonce=
0}
Block{previousHash='8063c767bd6d2f10dcdf1534699b8966a390b5bc23048917a3b7c11990e95590', hash='0000e0bfdfb8df53987a60d59dc
95194e12ead73487b460a38311ff8a252c0be', transactions=[Transaction{sender='Alice', receiver='Bob', amount=100.0}, Transac
tion{sender='Bob', receiver='Charlie', amount=50.0}], nonce=134580}
Block{previousHash='0000e0bfdfb8df53987a60d59dc95194e12ead73487b460a38311ff8a252c0be', hash='000093f6ee1f3b75ac8ff19bff9
430b0f4441cf33072ea40e5e331f5842cb27e', transactions=[Transaction{sender='Charlie', receiver='Dave', amount=200.0}, Tran
saction{sender='Dave', receiver='Alice', amount=150.0}], nonce=28689}

C:\Users\STUDENT\Desktop>
```