Assignment 2 Report

Andrew T. Nguyen
300019561
Oct 20th

Miner id: anguy116@uottawa.ca

Classes:

- Transaction:
    - A class that creates and stores transactions made in the BlockChain. The information stored in the transaction is the name of the Sender, name of the Receiver and the amount sent

  Transaction's Methods:
    - Transaction(String sender,String receiver,**int** amount)
        - Creates a Transaction with the given sender, receiver and amount
    - getSender()/getReceiver()/getAmount()
        - Getter Methods
    - toString()
        - Returns the given Transaction's sender, receiver and amount

- Block
    - A class that creates a Block in a BlockChain given the index of block, Timestamp of block creation, nonce string , previous hash code, hash code, and Transaction

  Block's Methods:
    - public Block(int index, long timestamp, String nonce, String previousHash, String hash, Transaction transaction)
        - Constructor
    - getIndex()/gettimestamp()/getSender()/getReceiver()/getAmount()/getNonce()/getPreviousHash()/getHash()
        - Getter methods
    - toString()
        - Prints the index of block, Timestamp of block creation, nonce string , previous hash code, hash code, and Transaction of the given Block

- BlockChain
    - A class that reads files into BlockChains, writes them into text files, validates BlockChains and adds to BlockChains

  BlockChain's Methods:
    - BlockChain(ArrayList<Block> blockchain)
        - Constructor
    - getSize()/getLastHash/getBalance
        - Getter Methods
    - isUsername(String username)
        - Checks if the user exists in the BlockChain
    - fromFile(String fileName)
        - Reads a text file and creates a BlockChain out of the given information in the text file
    - toFIle(String fileName)
        - Writes the existing BlockChain into a text file the same way fromFile(String fileName) reads files
    - validateBlockChain()

        - Validates the BlockChain by
            - Making sure all block's hash and hash generation of that block's toString equal
            - Making sure that all block's hashes and previous hashes consistent
            - Making sure that all block indexes are consistent
    - generateNonce()
        - Generates a random string with ascii values between 33 and 126
    - addBlock(Block block)
        - Adds the given block to the end of the BlockChain

Proof of Work:
The generation of nonce is randomized, this randomization if called upon several times will eventually make the hash code generation of a certain block in the BlockChain's toString() method start with "00000". This assures that the nonce string will not solely result in the generation of the hash code but along with the block's index, Timestamp, previous hash code, hash code, Transaction as well as the nonce. Thus, the random generation of nonce inside a group of specific values insures every blocks hash code creation is unique.

Statistics:

Transaction number	Number of Attempts
1	755946
2	3016874
3	241394
4	542639
5	28041
6	1644220
7	600189
8	2144548
9	199154
10	476587
Avg	964959.2

Notes for TA:
The scanner in the main should work to find a file but I guess because of different working directories on my computer I cannot simply type in a file ("file.txt") but I have to make a resources folder inside my project and put the file in there. Only then can I call the file ("resouces/file.txt") hopefully you won't have to do this but if for any reason it gives you the file not found exception from doing the initial "file.txt" that is the reason. I spend hours on stack overflow trying to find a solution but to no avail.