CSI 2110 Assignment 2

Ivana Erlich - 8682436
Miner id: ierli042

Classes:

Transaction: Stores a sender, receiver, and the amount sent

Methods:

- Constructor
- Getters for each attribute
- toString() returns the string representation of the Transaction

Block: Stores a transaction, it's timestamp, it's index in the blockchain, the nonce, the hash, the hash of the previous block in the blockchain, and the number of hash trials to find the nonce

Methods:

- one Constructor for a Block read in from a file, one Constructor for a newly created Block
- getters for the class's attributes
- hash() finds and sets the nonce for the Block, as well as sets the hash for the block and returns the number of hash trials
- genNonce(int x) takes an integer x as input and returns a string which is then tested in hash() to see it passes the hash trial
- toString() returns the string representation of the Block

BlockChain: Stores a list of blocks, a hashmaps of balances of the users, and a list of the amount of hash trials for each of the blockchain

Methods:

- fromFile(String fileName) reads the file with file name fileName and adds it to the blockchain
- toFile(String fileName) writes the BlockChain into the file with file name fileName
- validateBlock() checks that all the hashes are correct, all the indexes are correct, the hash of each block matches the previoushash of the next block in the blockchain, and that no user has spent more bitcoin than they have
- getBalance(String username) returns the balace of the user username from the hashmap of balances
- setBalance(String username, int amount) sets the balance of the user in the hashmap of balances
- add(Block block) adds block to the blockchain
- getBlock(int index) returns block with index index
- size() returns size of blockchain
- main function runs the program by calling the above methods to read a file, validate the chain, add new blocks, and write to a file

Proof of work:
The proof of work algorithms implements the base conversion strategy taught in ITI1100 by going through the integers from 0 up and for each integer converting it to the nonce representation by using the array of characters allowed in a nonce. The algorithm for this conversion divides this integer by 94 (the number of characters allowed in a nonce) until the integer is not greater than 0 and taking the remainder each time, then taking the array at the index of the remainder for each remainder in reverse order to make the nonce string. This is repeated until a nonce is found such that the hash of the string representation of the Block starts with “00000”.

Statistics:

Average: 967187	
Index	# of hash trials
0	N/a (not generated)
1	N/a (not generated)
2	N/a (not generated)
3	283600
4	608037
5	120882
6	1043427
7	1021477
8	387319
9	2761928
10	2289839
11	514198
12	641167