Specification of Blockchain Puzzle Program
Colin Gallaway and Andrew Chough

The code is written in Java and is thus geared towards an object oriented design.

Class Hash

This class intends to deal with SHA-256 hashing.

Constructors:

Hash(String string)
Hashes the given string with SHA-256 algorithm and constructs a Hash object

Hash()
Produces a pseudo-random number, converts it into a string and passes it into Hash(String string)
Intended for retrieving pseudo-random hash values

Hash(BigInteger bigInteger)
Converts the argument into a string and passes it into Hash(String string)

Hash(Hash hash)
Converts the argument into a string and passes it into Hash(String string)

Methods:

String toString()
Returns a String version of Hash (Uppercase Hexadecimal SHA-256 hash).

BigInteger toBigInteger()
Returns a BigInteger version of Hash

Class BlockHeader

A data structure for a block header.
Organizes a block header into private fields of a BlockHeader object.

A header contains the following:
blockVersion – version of block header (default is 2)
prevBlock – a 256-bit hash pointer to the previous block
merkleRoot – a 256-bit hash pointer to the merkle root
timeStamp – a unix-based time stamp created when block header is created
target – the solution is less than this value
nonce – 32-bit number used to solve blockchain puzzle

Constructors:

BlockHeader(int blockVersion, Hash prevBlock, Hash merkleRoot, BigInteger target, BigInteger nonce)

Constructs a BlockHeader object with given arguments. Time stamp is set to unix time.

BlockHeader(BigInteger target)

Constructs a BlockHeader object with given target value, random hashes, and default values for blockversion (2) and nonce (0). Time stamp is set to unix time.

Methods:

int getBlockVersion()

Returns block version as an int.

Hash getPrevBlock()

Return Hash object (hash pointer) of previous block.

Hash getMerkleRoot()

Returns Hash object (hash pointer) of merkle root.

long getTimeStamp()

Returns time stamp as a long.

BigInteger getTarget()

Returns target as a BigInteger.

BigInteger getNonce()

Returns nonce as a BigInteger.

toString()

Returns contents of block header formatted as a string.

Class Block

The main method is found in this class file. It iterates over 7 difficulties and finds 5 solutions of the blockchain puzzle for each difficulty. The difficulties, in hexadecimal are F, FF, FFF, FFFF, FFFFF, FFFFFF, FFFFFFF.

This class intends to solve the blockchain puzzle given a previous block header and difficulty.

Constructors:

Block(String difficulty)

Attempt to mine a block using a fake previous block (random hashes). The target is created by taking the max hash value (256-bits) divided by the difficulty.

Block(BlockHeader prevBlockHeader)

Attempt to mine a block using the previous block from the argument. If successful, the hash solution can be retrieved with getHashValue(). If the nonce value surpasses 32-bits, the hashing will stop and a solution will not be found.

Methods:

String getHashValue()

Returns a string version (Uppercase Hexadecimal) of the hash solution.