package Blockchain;

import java.security.MessageDigest;

import java.util.Date;

public class Block {

public static String hash;

public String previousHash;

private String data; //our data will be a simple message.

public static long timeStamp; //as number of milliseconds since 1/1/1970.

public static int nonce;

//Block Constructor.

public Block(String data,String previousHash ) {

this.data = data;

this.previousHash = previousHash;

Block.timeStamp = new Date().getTime();

Block.hash = calculateHash(); //Making sure we do this after we set the other values.

}

//Applies Sha256 to a string and returns the result.

public static String applySha256(String input){

try {

MessageDigest digest = MessageDigest.getInstance("SHA-256");

//Applies sha256 to our input,

byte[] hash = digest.digest(input.getBytes("UTF-8"));

StringBuffer hexString = new StringBuffer(); // This will contain hash as hexidecimal

for (int i = 0; i < hash.length; i++) {

String hex = Integer.toHexString(0xff & hash[i]);

if(hex.length() == 1) hexString.append('0');

hexString.append(hex);

}

return hexString.toString();

}

catch(Exception e) {

throw new RuntimeException(e);

}

}

//Calculate new hash based on blocks contents

public String calculateHash() {

String calculatedhash = applySha256(

previousHash +

Long.toString(timeStamp) +

Integer.toString(nonce) +

data

);

return calculatedhash;

}

public void mineBlock1(int difficulty) {

String target = new String(new char[difficulty]).replace('\0', '0'); //Create a string with difficulty \* "0"

while(!hash.substring( 0, difficulty).equals(target)) {

nonce ++;

hash = calculateHash();

}

// System.out.println("Nonce Time "+nonce);

// System.out.println("Block Mined!!! : " + hash);

}

public void mineBlock2(int difficulty) {

String target = new String(new char[difficulty]).replace('\0', '0'); //Create a string with difficulty \* "0"

while(!hash.substring( 0, difficulty).equals(target)) {

nonce ++;

hash = calculateHash();

}

}

public void mineBlock3(int difficulty) {

String target = new String(new char[difficulty]).replace('\0', '0'); //Create a string with difficulty \* "0"

while(!hash.substring( 0, difficulty).equals(target)) {

nonce ++;

hash = calculateHash();

}

}

public void mineBlock4(int difficulty) {

String target = new String(new char[difficulty]).replace('\0', '0'); //Create a string with difficulty \* "0"

while(!hash.substring( 0, difficulty).equals(target)) {

nonce ++;

hash = calculateHash();

}

}

}