|  |  |  |  |
| --- | --- | --- | --- |
| Государственное бюджетное профессиональное образовательное учреждение Новосибирской области | | | |
| **«Новосибирский химико-технологический колледж им. Д.И. Менделеева»** | | | |
| (ГБПОУ НСО «Новосибирский химико-технологический колледж им. Д.И.Менделеева») | | | |
| **ИНДИВИДУАЛЬНЫЙ ПРОЕКТ**  **ИЗУЧЕНИЕ ТЕХНОЛОГИИ BLOCKCHAIN** | | | |
|  | | | |
|  | | | |
| Специальность: 09.02.07 «Информационные системы и программирование» | | | |
| Выполнил студент группы 09.07.32 | | | |
|  |  |  |  |
| *подпись Самойлов кирилл Денисович* | | | |
|  | | | |
|  | | | |
| Руководитель проекта преподаватель Основ Программирования | | | |
| *подпись Ворбьёв Александр Александрович* | | | |
|  | | | |
| «Нормоконтроль» | | | |
| *подпись* | | | |
|  | | | |
|  | | | |
|  | | | |
|  | | | |
| Новосибирск 2023 | | | |
|  | | | |

# ОГЛАВЛЕНИЕ

[ОГЛАВЛЕНИЕ 2](#__RefHeading___Toc247_271921410)

[​ ВВЕДЕНИЕ 3](#__RefHeading___Toc249_271921410)

[ОСНОВНЫЕ ПОНЯТИЯ 3](#__RefHeading___Toc1918_2251456438)

[​ ЧАСТЬ 1 ТЕОРИЯ 4](#__RefHeading___Toc251_271921410)

[1.1 Принцип работы технологии 4](#__RefHeading___Toc1920_2251456438)

[1.2 Преимушества технологии 8](#__RefHeading___Toc1922_2251456438)

[​ ЧАСТЬ 2 ПРАКТИКА 10](#__RefHeading___Toc253_271921410)

[​ ЗАКЛЮЧЕНИЕ 19](#__RefHeading___Toc548_2558360452)

[​ СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 20](#__RefHeading___Toc1604_2251456438)

# ВВЕДЕНИЕ

**Актуальность:** Актуальность данной темы заключается в том, что на этой технологии сейчас работают все криптобиржи и некоторые серьёзные бизнессы. Изучая технологию у меня будет возможность написать и продать очень важнные программы. Актуальность данной темы пока на низком уровне т.к., так называемый, бум майнинга уже прошёл. Но эта тема может стать вновь актуальной т.к. курс крипто валют не постоянен.

**Цель:** Изучить техногию Blockchaine.

**Задачи:** Изучить основные принципы работы технологии; Изучить материал по теме, который бы расскрывал реализацию этой темы в виде программы; Написать программу на java, которая будет демонстрировать эти принципы.

ОСНОВНЫЕ ПОНЯТИЯ

Blockchaine — выстроенная по определённым правилам непрерывная последовательная цепочка блоков, содержащих информацию о транзакциях. Связь между блоками обеспечивается не только нумерацией, но и тем, что каждый блок содержит свою собственную хеш-сумму и хеш-сумму предыдущего блока. Blockchaine позволяет осуществить открытый обмен инфомацией.

Майнинг — это присоединение блоков, в которых хранится информация о проведенных транзакциях, за счет мощьности компьютеров и других устройств, которые предоставляют майнеры.

Хеш — отображение данных в виде короткой строки, в идеале — уникальной для каждого набора информации. Размер строки может быть одинаковым для информации разного объема.

Хеширование — это преобразование информации с помощью особых математических формул. В результате возникает хеш.

# ЧАСТЬ 1 ТЕОРИЯ

Блокчейн применяется во всех сферах, где необходима быстрота передачи информации с высокой степенью ее защиты. Технология используется для запуска и работы криптовалют и цифровых валют, при заключении смарт-контрактов на поставку товаров, при генерации невзаимозаменяемых токенов (non-fungible tokens, NFT), в банковской и правовой сферах, при администрировании сетей и в игровой индустрии. Технологии блокчейна находят применение в работе органов государственной власти (например, при проведении и обработке результатов референдумов и голосований), в деятельности публичных и непубличных корпораций, общественных организаций и частных лиц.

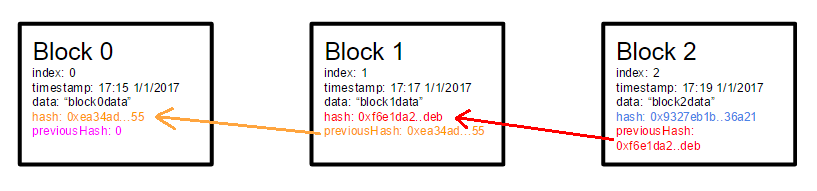
1.1 Принцип работы технологии

Традиционные технологии баз данных создают ряд проблем, связанных с учетом финансовых операций. Рассмотрим пример с продажей недвижимости. После передачи денег право собственности переходит покупателю. Как покупатель, так и продавец могут самостоятельно регистрировать денежные операции, но ни одной из сторон нельзя доверять. Получив деньги, продавец может легко утверждать, что он их не получил, а покупатель может утверждать, что деньги отправлены, даже если это не так.

Блокчейн предвосхищает подобные проблемы путем создания децентрализованной, защищенной от несанкционированного доступа системы для записи операций. В случае сделки с недвижимостью блокчейн создает единый реестр для покупателя и продавца. Все транзакции должны быть одобрены обеими сторонами и автоматически обновляться в их реестрах в режиме реального времени. Любое несоответствие в истории транзакций отразится во всем реестре. Эти свойства технологии блокчейн сделали ее популярной в различных секторах. Например, они использовалась при создании цифровой валюты.

Система генерирует два разных ключа для каждого пользователя сети. Один ключ — публичный, общий для всех участников сети. Второй — уникальный частный ключ. Сочетание частного и публичного ключей разблокируют данные в реестре. Ключи позволяют подтвердить подленность транзакции.

Когда происходит каждая транзакция, она записывается как блок данных (рисунок 1).

рисунок 1 — пример цепочки блоков

Эти транзакции показывают движение актива, который может быть материальным (продукт) или нематериальным (интеллектуальным).

Для учета, хранения и иных действий с цифровыми активами используется блокчейн-кошелек — специальная программа, которая хранит записи о состоянии счета его владельца и всю историю транзакций.

Блок данных может записывать информацию по вашему выбору: кто, что, когда, где, сколько и даже состояние — например, температура груза с едой. Новые блоки в цепь добавляют майнеры.

Майнеры выполняют в блокчейне несколько функций:

* хранят копии блокчейна и тем самым защищают информацию от потери или подделки;
* подтверждают транзакции;
* проверяют транзакции, которые зарегистрировали другие майнеры.

Как правило, число майнеров не ограничено. Чем их больше, тем лучше — такая сеть надежнее. Майнерами могут стать все желающие. Для этого им нужны специализированные компьютеры и программное обеспечение.За поддержание работы сети майнеры получают награду. Как правило, это комиссии от всех участников транзакций, записанных в блоке, и вознаграждение от самой сети. Сеть генерирует это вознаграждение по определенному алгоритму.

Каждый блок связан с предыдущими и последующими блоками. Эти блоки образуют цепочку данных по мере того, как актив перемещается с места на место или переходит из рук в руки. Блоки подтверждают точное время и последовательность транзакций, и блоки надежно связаны друг с другом, чтобы предотвратить изменение любого блока или вставку блока между двумя существующими блоками.

Каждый дополнительный блок усиливает проверку предыдущего блока и, следовательно, всей цепочки блоков. Это делает блокчейн защищенным от несанкционированного доступа, обеспечивая ключевую силу неизменности. Это устраняет возможность вмешательства злоумышленника и создает реестр транзакций, которому вы и другие участники сети можете доверять.

В блокчейне реестр владельцев хранится на множестве независимых компьютеров, объединенных через интернет. Как следствие, в блокчейне реестры с данными о владельцах активов невозможно подделать. Ведь эти данные хранятся на компьютерах огромного числа участников сети (рисунок 2). А чтобы информация у всех пользователей была абсолютно полной и верной, в блокчейне ввели понятие консенсус. Процедура консенсуса — процедура достижения согласия.

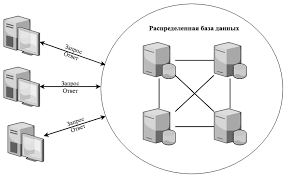


Рисунок 2 — визуализация работы реестров в блокчейне.

В блокчейн существует четыре основных типа децентрализованных или распределенных сетей:

Публичные. Публичные блокчейны не требуют разрешений и позволяют любому желающему присоединиться к сети. Все участники блокчейна имеют равные права на чтение, редактирование и проверку информации. Для обмена и майнинга таких криптовалют, как Bitcoin, Ethereum и Litecoin, в основном используются публичные блокчейны.

Частный. Частные блокчейны, которые также можно назвать управляемыми, контролируются одной организацией. Уполномоченный орган определяет, кто может быть участником и какими правами в сети они обладают. Частные блокчейны децентрализованы только частично, поскольку включают ограничения доступа. Примером частного блокчейна является Ripple — платформа для обмена цифровой валюты.

Гибридный блокчейн. Гибридный блокчейн сочетает в себе функции как частных, так и публичных сетей. Компании могут создавать как частные, так и публичные системы разрешений. Таким образом, они контролируют доступ к определенным данным в блокчейне, но при этом поддерживают общедоступный доступ к другим данным. Они используют смарт-контракты, позволяющие публичным участникам удостовериться в проведении частных транзакций. Например, гибридные блокчейны могут предоставлять публичный доступ к цифровой валюте, сохраняя частный доступ к банковской валюте.

Блокчейн-консорциумы. Блокчейн-консорциумами управляет группа организаций. Выбранные заранее организации разделяют ответственность за функционирование блокчейна и определение прав доступа к данным. Блокчейн-консорциумы часто предпочитают компании-единомышленники, извлекающие выгоду из общей ответственности. Например, Global Shipping Business Network — это некоммерческий блокчейн-консорциум, специализирующийся на цифровизации судоходной отрасли и расширении сотрудничества между операторами морских перевозок.

1.2 Преимушества технологии

С блокчейном вы можете быть уверены, что получаете точные и своевременные данные, и что ваши конфиденциальные записи блокчейна будут доступны только тем участникам сети, которым вы специально предоставили доступ.

От всех участников сети требуется консенсус в отношении точности данных, и все проверенные транзакции неизменны, поскольку они записываются постоянно. Никто, даже системный администратор, не может удалить транзакцию.

Благодаря распределенному реестру, совместно используемому членами сети, исключается трата времени на согласование записей. А для ускорения транзакций набор правил, называемый смарт-контрактом, может храниться в блокчейне и выполняться автоматически.

Сделки конфиденциальны и анонимны: покупатель указывает только номер своего криптокошелька.

Комиссии минимальны, поскольку вместо централизованных посредников транзакции регистрируют майнеры. Комиссии — это их вознаграждение за поддержку работы блокчейн-сети. Но майнеров обычно очень много и конкуренция между ними высока — это и позволяет удерживать комиссии на низком уровне.

Информация надежно хранится, поскольку история всех операций записана в блокчейне и распределена по всем участникам сети. Каждый блок содержит информацию обо всех предыдущих операциях.

# ЧАСТЬ 2 ПРАКТИКА

Свою программу я писал на языке java в Intelj IDEA. В своей программе я написал простой код, который, правда, не расскрывает и половины всех возможностей технологии блокчейн. Я начал с написания класса «Block», который отвечает только за то, из чего будет состоять объект этого класса(блок):

import java.util.Date;

public class Block {

String Hash, PrevHash;

int Index;

String Key, data;

Date date;

public Block( int Index, Date date, String PrevHash,

String data, String Hash, String Key) {

this.PrevHash = PrevHash;

this.Index = Index;

this.date = date;

this.data = data;

this.Hash = Hash;

this.Key = Key;

}

}

Здесь функция с именем класса — это конструктор, который и определяет то, какие данные обязательно должны входить в объекты этого класса. Далее я принялся реализовать генерацию блоков и их хеша. Генерация блоков не составила проблем т.к. это было лишь генерацией объетов класса «Block», с нужными полями их запись в реестры объектов класса Bank, которые представляют собой txt файлы:

import java.security.MessageDigest;

import java.security.NoSuchAlgorithmException;

import java.util.ArrayList;

import java.util.Date;

public class GenerateBlock {

<...>

void echo(Block block, Bank b1) throws IOException {

File file1 = new File(b1.log\_name);

FileWriter fw1 = new FileWriter(file1, true);

fw1.write(block.Hash + " : Block " + block.Index + " : " + block.data + " | " + block.date + " | " + block.PrevHash + " + " + block.Key + "\n");

fw1.close();

}

<...>

public void GenBlock(int times, String filename, Bank b1, Bank b2, Bank b3, Bank b4) throws NoSuchAlgorithmException, IOException {

String Hash, PrevHash = "0";

int Index = 1;

String OurHashString;

File logs = new File(filename);

String data = "1 transaction: NULL :: 7ae16e3ac9c3a7a6b08e96c8adf54035+;";

while (times > 0) {

System.out.println(times + "\n");

RandomKey();

Date date = new Date();

OurHashString = PrevHash + Index + date + data + Key;

Hash = hash256(OurHashString);

if (Hash.contains("000000")) {

Block newBlock = new Block( Index, date, PrevHash, data, Hash, Key);

Index++;

PrevHash = Hash;

echo(newBlock, b1);

echo(newBlock, b2);

echo(newBlock, b3);

echo(newBlock, b4);

}

times--;

}

}

Затем были разработанны функции для хеширования блоков, для перевода массивов перменных типа «byte» в HEX формат:

public class GenerateBlock {

<...>

public static String hash256(String data) throws NoSuchAlgorithmException {

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

md.update(data.getBytes());

return bytesToHex(md.digest());

}

public static String bytesToHex(byte[] bytes) {

StringBuffer result = new StringBuffer();

for (byte byt : bytes) result.append(Integer.toString((byt & 0xff) +

0x100, 16).substring(1));

return result.toString();

}

public String Key = "";

public void RandomKey() {

final int KeyLength = 20;

char[] str = new char[KeyLength];

int RandomNumber;

Key = "";

for (int i = 0; i < KeyLength; i++) {

RandomNumber = 65 + (int) (Math.random() \* 25);

str[i] = (char) RandomNumber;

}

for (int i = 0; i < KeyLength; i++) {

Key = Key + str[i];

}

}

<...>

}

Здесь функция SHA256 реализованна через функцию MessageDigest, т.к. в java она прописанна не так, как другие(базовые) функции, и throws NoSuchAlgorithmException здесь нужен как раз для этой функции, вернее для её корректной работы. И в завершении всего проекта я написал порстой код для запуска всего проекта я написал простенький main класс:

import java.io.IOException;

import java.security.NoSuchAlgorithmException;

class Main {

public static void main(String[] args) throws NoSuchAlgorithmException, IOException {

int t = 1000000;

Bank bank1 = new Bank("Bank1\_log.txt");

Bank bank2 = new Bank("Bank2\_log.txt");

Bank bank3 = new Bank("Bank3\_log.txt");

Bank bank4 = new Bank("Bank4\_log.txt");

bank1.GetBlock(t, bank1, bank2, bank3, bank4);

bank1.Consensus(bank2.log\_name, bank3.log\_name, bank4.log\_name);

}

}

Что здесь стоит подметить так это то что вся программа запускается через методы класса Bank. Кстати, вот сам класс:

import java.io.\*;

import java.security.NoSuchAlgorithmException;

import java.util.ArrayList;

import java.util.Scanner;

public class Bank {

String log\_name;

public Bank(String log\_name) {this.log\_name = log\_name;}

void GetBlock(int times, Bank b1, Bank b2, Bank b3, Bank b4) throws NoSuchAlgorithmException, IOException {

GenerateBlock GB = new GenerateBlock();

GB.GenBlock(times, log\_name , b1, b2, b3, b4);

}

void Consensus(String logs1, String logs2, String logs3) throws IOException {

File OurLogs = new File(log\_name);

File log1 = new File(logs1);

File log2 = new File(logs2);

File log3 = new File(logs3);

ArrayList<String> Ourstr = new ArrayList<>();

String str1, str2, str3;

FileReader OurFR = new FileReader(OurLogs);

Scanner OurScan = new Scanner(OurFR);

FileReader FR1 = new FileReader(log1);

Scanner ScanLog1 = new Scanner(FR1);

FileReader FR2 = new FileReader(log2);

Scanner ScanLog2 = new Scanner(FR2);

FileReader FR3 = new FileReader(log3);

Scanner ScanLog3 = new Scanner(FR3);

while (OurScan.hasNextLine()) {

Ourstr.add(OurScan.nextLine());

}

int i = 0;

for (; ScanLog1.hasNextLine() && ScanLog2.hasNextLine() && ScanLog3.hasNextLine(); i++) {

str1 = ScanLog1.nextLine();

str2 = ScanLog2.nextLine();

str3 = ScanLog3.nextLine();

if (!Ourstr.get(i).equals(str1) || !Ourstr.get(i).equals(str2) || !Ourstr.get(i).equals(str3)) {

if (str1.equals(str2) && str2.equals(str3)) {

Ourstr.set(i, str1);

} else if (str1.equals(str2)) {

Ourstr.set(i, str1);

} else if (str2.equals(str3)) {

Ourstr.set(i, str2);

} else if (str1.equals(str3)) {

Ourstr.set(i, str1);

}

}

}

OurFR.close();

OurScan.close();

FR1.close();

ScanLog1.close();

FR2.close();

ScanLog2.close();

FR3.close();

ScanLog3.close();

FileWriter FW = new FileWriter(OurLogs);

for (i = 0; i < Ourstr.size(); i++) {

FW.write(Ourstr.get(i) + "\n");

}

FW.close();

}

}

Здесь самый большой метод это метод consensus. Он отвечает за сравнение файлов и запись в 1 файл всех правильных блоков. FileWriter и FileReader, которых в этом классе очень много, отвечают за запись в файл и чтение из него строк соответвенно.

# ЗАКЛЮЧЕНИЕ

В заключении хочется отметить что, я так не смог в полной мере показать все основные особенности технологии т.к. это оказалось гораздо сложнее, чем я думал изначально. Тем не мение, я в полной мере ознакомился с технологией блокчейн в теоритическом плане и вполне смогу реализовать программу, которая будет демонстрировать все возможности блокчейна, но на это понадобится гораздо больше времени, опыта и знаний. В целом, считаю что я выполнил все поставленные задачи в пределах своих возможностей.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

<https://blog.skillfactory.ru/glossary/heshirovanie/>

<https://www.ibm.com/topics/what-is-blockchain>

<https://aws.amazon.com/ru/what-is/blockchain/?aws-products-all.sort-by=item.additionalFields.productNameLowercase&aws-products-all.sort-order=asc>

<https://fincult.info/article/blokcheyn-chto-eto-takoe-i-kak-ego-ispolzuyut-v-finansakh/>