УДК 004

Создание блокчейн с использованием C++

Еровлева Регина Викторовна
Приамурский государственный университет имени Шолом-Алейхема
Студент

Еровлев Павел Андреевич
Приамурский государственный университет имени Шолом-Алейхема
Студент

Аннотация
В данной статье будут рассмотрены возможности создания собственного базового блокчейн. Так же будет внедрена система проверки доказательства работы. Данная работа будет выполнена в среде разработки CLion с использованием языка программирования C++.
Ключевые слова: C++, Blockchein, Bitcoin

Building a blockchain using C++

Erovleva Regina Viktorovna
Sholom-Aleichem Priamursky State University
Student

Erovlev Pavel Andreevich
Sholom-Aleichem Priamursky State University
Student

Abstract
This article will consider the possibilities of creating your own basic blockchain. A proof-of-work verification system will also be introduced. This work will be done in the CLion development environment using the C++ programming language.
Keywords: C++, Blockchein, Bitcoin

В настоящее время технологии развиваются очень быстро и потребность в информационной безопасности постоянно возрастает. В связи с этим становится востребованной технология Блокчейн, позволяющая сохранять целостность информации. Кроме того, технология позволяет создать децентрализованную среду, в которой транзакции и данные происходят без какой-либо сторонней организации.

Цель работы – создать базовую модель технологии блокчейн с использованием языка C++.

Исследованиями в данной теме занимались следующие авторы.

П.А. Андреева, Р.А. Андреев в своей статье рассмотрела разновидности блокчейн технологий и виды их применения [1]. М.П. Воронов, В.П.Часовских, В.Г. Лабунец, рассмотрели в своей работе технологию блокчейн как основу развития и повышения эффективности электронной системы ВУЗа [2]. С.Г. Наумов в своей работе представил технологию блокчейн, способы добытия криптовалюты [3]. Б. Кумалаков и Я. Шакан предложили разработку децентрализованного студенческого приложения, который будет обрабатывать и хранить токены, которые будут представлять собой кредиты, которые студенты будут получать за прохождение определенных курсов [4]. Л.А. Симанович, расписала возможности использования блокчейн для электронных торговых площадок [5].

Создадим файл с именем «Block.h» в основной папке проекта. Внутри файла добавим код для импорта необходимых библиотек (рис.1).

```
#include <cstdint>
#include <iostream>
```

Рисунок 1 – Импорт библиотек

Так же добавим строку пространства имен (рис.2).

```
using namespace std;
```

Рисунок 2 – Пространство имен

Блокчейн состоит из серии блоков, содержащих данные, и каждый блок содержит криптографическое представление предыдущего блока, а это означает, что будет очень сложно изменить содержимое любого блока без необходимости изменения каждого последующего.

Создадим класс блока, для этого добавим в заголовочный файл «Block.h» следующие строки (рис.3).

```
class Block {
public:
    string sPrevHash;

    Block(uint32_t nIndexIn, const string &sDataIn);

    string GetHash();

    void MineBlock(uint32_t nDifficulty);

private:
    uint32_t _nIndex;
    int64_t _nNonce;
    string _sData;
    string _sHash;
    time_t _tTime;

    string _CalculateHash() const;
};
```

Рисунок 3 – Класс Создания блоков

Каждый блок связан с предыдущим блоком для этого создаем переменную, которая будет хранить предыдущих хэш блока. Сигнатура конструктора принимает три параметра для «nIndexIn» и «sDataIn». Далее указывается сигнатура метода «GetHash», за которой следует сигнатура метода «MineBlock», которая принимает параметр «nDifficulty». Здесь указываем приватный модификатор, за которым следуют приватные переменные «_nIndex», «_nNonce» , «_sData» , «_sHash» и «_tTime». В подписи для «_ CalculateHash» также есть ключевое слово «const», чтобы гарантировать, что метод не может изменить ни одну из переменных в классе блока, что очень полезно при работе с цепочкой блоков.

Теперь создадим заголовочный файл «Blockchain.h» в основной папке проекта (рис.4).

```
#include <cstdint>
#include <vector>
#include "Block.h"

using namespace std;
```

Рисунок 4 – Файл Blockchain.h

Теперь создадим класс блокчейна (рис.5).

```
class Blockchain {
public:
    Blockchain();

    void AddBlock(Block bNew);

private:
    uint32_t _nDifficulty;
    vector<Block> _vChain;

    Block _GetLastBlock() const;
};
```

Рисунок 5 – Реализация блокчейна

Как и в случае с классом блока, здесь вызывается класс блокчейна, за которым следует модификатор и сигнатура конструктора. Сигнатура «AddBlock» принимает параметр «bNew», который должен быть объектом созданного ранее класса «Block». Затем указываем приватный модификатор, за которым следуют приватные переменные для «_nDifficulty» и «_vChain», а также сигнатуру метода для «_GetLastBlock», за которой также следует константа, ключевое слово, для обозначения того, что вывод метода не может быть изменен.

Поскольку блокчейны используют криптографию, необходимо добавить некоторые криптографические функции в блокчейн. Будем использовать метод хэширования «SHA256» для создания хэшей блоков.

Технология блокчейн стала популярной, когда она была разработана для цифровой валюты Биткойн, поскольку криптовалюта неизменна и общедоступна, это означает, что когда один пользователь передает биткойны другому пользователю, транзакция для передачи записывается в блок в цепочке блоков узлами в сети биткойнов. Узел — это другой компьютер, на котором запущено программное обеспечение Биткойн, и, поскольку сеть является одноранговой, им может быть кто угодно в мире. Этот процесс называется «майнинг», поскольку владелец узла получает вознаграждение в биткойнах каждый раз, когда он успешно создает действительный блок в блокчейне.

Чтобы успешно создать действительный блок и, следовательно, получить вознаграждение, майнер должен создать криптографический хэш блока, который он хочет добавить в цепочку блоков, который соответствует требованиям к действительному хэшу на тот момент, это достигается путем подсчета количества нулей в начале хэша, если количество нулей равно или превышает уровень сложности, установленный сетью, этот блок действителен. Если хэш недействителен, увеличивается переменная, называемая одноразовым номером, и хеш создается снова. Этот процесс,

называемый «Proof of Work», повторяется до тех пор, пока не будет получен действительный хэш.

Добавим метод MineBlock, который будет искать действительный хэш блока (рис.6).

```
void Block::MineBlock(uint32_t nDifficulty) {
    char cstr[nDifficulty + 1];
    for (uint32_t i = 0; i < nDifficulty; ++i) {
        cstr[i] = '0';
    }
    cstr[nDifficulty] = '\0';

    string str(cstr);

    do {
        _nNonce++;
        _sHash = _CalculateHash();
    } while (_sHash.substr(0, nDifficulty) != str);

    cout << "Block mined: " << _sHash << endl;
}
```

Рисунок 6 – Поиск хэш

Теперь обновим файл «main.cpp» (рис.7).

```
int main() {
    Blockchain bChain = Blockchain();

    cout << "Майнинг блока 1..." << endl;
    bChain.AddBlock(Block(1, "Блок данных 1"));

    cout << "Майнинг блока 2..." << endl;
    bChain.AddBlock(Block(2, "Блок данных 2"));

    cout << "Майнинг блока 3..." << endl;
    bChain.AddBlock(Block(3, "Блок данных 3"));

    return 0;
}
```

Рисунок 7 – Добавление и нахождение блоков в блокчейн

Теперь можно запустить программу и посмотреть как будет работать программа (рис.8).

```
Майнинг блока 1...
Добыт блок: 000000c1b50cb30fd8d9a0f2e16e38681cfcf9caead098cea726854925ab3772
Майнинг блока...
Добыт блок: 0000005081063c8c854d11560cfea4fe734bde515a08565c26aa05448eea184e
Майнинг блока 3...
Добыт блок: 000000ea61810fa85ff636440eb803263daf06b306c607aced9a1f996a421042
```

Рисунок 8 – Работа программы

В данной статье был рассмотрен процесс разработки технологии блокчейн с использованием языка C++.

Библиографический список

1. Андреева П.А., Андреев Р.А. Обзор технологии блокчейн: виды блокчейна и их применение // Интеллектуальные системы в производстве 2019. №1. С. 11-14.
2. Часовских В.П., Лабунец В.Г., Воронов М.П. Технология "блокчейн" (blockchain) в образовании вузов и цифровой экономике // Эко-потенциал 2018. №2(18). С. 99 - 105.
3. Наумов С.Г. Блокчейн, криптовалюта и майнинг // Евразийская адвокатура 2021. №2(51). С. 32.
4. Кумалаков Б., Шакан Я. Блокчейн в образовании: как управлять студенческим зачетом вуза через блокчейн? // Вестник алматинского университета энергетики и связи 2020. №2(49). С. 128-133.
5. Симанович Л.А. Возможности использования технологии блокчейн (blockchain) для электронной торговой площадки // Национальная безопасность и стратегическое планирование 2019. №1(21). С. 134-140.