Харківський національний університет імені В. Н. Каразіна

ННІ Комп’ютерних наук та штучного інтелекту

Кафедра інтелектуальних програмних систем і технологій

ЗВІТ З ЛАБОРАТОРНОЇ РОБОТИ №2

на тему «Аналіз структури транзакцій Bitcoin. Програмна реалізація

примітивів блок і транзакція»

з дисципліни «Технології блокчейн та криптовалютні операційні платформи»

Виконав: студент групи КС32

*Жимеренко Степан*

Перевірила:

PhD, доц. Родінко М.Ю.

Харків – 2024

**Завдання**:

1. Розробити програмний примітив, що представляє транзакцію, тобто, клас Transaction, який містить приблизно наступні поля:

● вхідна адреса відправника (input);

● вихідна адреса(-и) отримувача(-ів) (output(s));

● сума переказу транзакції (amount);

● часова мітка створення транзакції (txTimestamp);

● геш транзакції (txHash);

● цифровий підпис.

Додати відповідні конструктори, сетери, гетера та toString(), метод для верифікації геш-значення та цифрового підпису транзакції.

2. Розробити програмний примітив, що представляє блок, тобто, клас

Block, який містить приблизно наступні поля (див. рис. 6):

● версія протоколу (version);

● геш-значення заголовку попереднього блока (prevHash);

● мітка часу (timestamp);

● складність видобутку блока, тобто, майнінгу (difficulty target);

● випадкове значення (nonce);

● корінь дерева Меркла (MerkleRoot);

● транзакції у вигляді дерева Меркла (геш-дерева);

● цифровий підпис.

Додати відповідні конструктори, сетери, гетера та toString().

Окрім, цього необхідно створити метод для генерації геш-значення заголовку блока за усіма його атрибутами, метод для верифікації блока (перевірки коректності геш-значення заголовку блока, кореня дерева та цифрового підпису).

Лістинг моєї програми можна побачити за цим посиланням:

Ця програма є реалізацією простого блокчейну на мові Java, і я вирішив створити її для того, щоб зрозуміти, як працюють основні концепції блокчейну, такі як транзакції, блоки та верифікація даних.

На початку програми я генерую пару ключів для трьох користувачів: Аліси, Боба та Чарлі, використовуючи алгоритм RSA. Це важливий етап, адже ключі знадобляться для підписування транзакцій та забезпечення їхньої цілісності.

Далі я створюю сам блокчейн, ініціалізуючи новий об'єкт класу Blockchain. Згодом я генерую кілька транзакцій. Наприклад, Аліса надсилає 50 одиниць своїм друзям Бобу та Чарлі. Я підписую цю транзакцію за допомогою приватного ключа Аліси, щоб підтвердити, що саме вона є відправником. Також Боб проводить транзакцію на користь Чарлі, і я підписую цю транзакцію його приватним ключем.

Після створення транзакцій я зберігаю їх у списку та вивожу на екран всю інформацію про них, включаючи перевірку їхньої валідності. Кожна транзакція проходить верифікацію, і я переконуюсь, що її підпис дійсний.

Наступним кроком є створення нового блоку, який міститиме згенеровані транзакції. Я встановлюю складність майнінгу та намагаюся видобути блок, поки хеш не відповідатиме цільовому значенню. Після успішного майнінгу я підписую блок приватним ключем Аліси.

Перш ніж додати блок до блокчейну, я перевіряю його валідність, а також підтверджую правильність підпису. Якщо всі умови дотримано, блок успішно додається до блокчейну.

На завершення програми я виводжу весь блокчейн та перевіряю його цілісність. Ця перевірка включає в себе верифікацію всіх блоків у ланцюзі, щоб впевнитись, що дані не були змінені. Програма завершує свою роботу, демонструючи, як різні компоненти блокчейну працюють разом для забезпечення безпеки та прозорості.

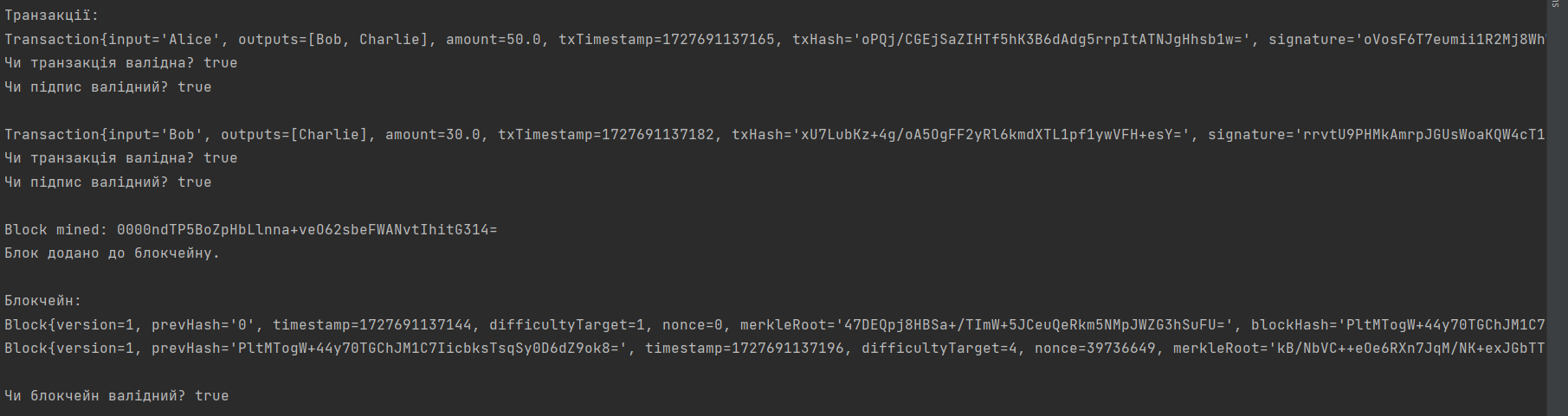
Ця реалізація дозволяє мені глибше зрозуміти, як функціонують транзакції та блоки, а також як забезпечується безпека у дистрибутивних системах. Я відчуваю, що з цим досвідом можу краще орієнтуватися в світі блокчейн-технологій.

Рисунок 1 − Результат компіляції програми

Коли я запустив свою програму, спочатку вона вивела інформацію про дві транзакції. Перша транзакція, в якій Аліса відправила 50 одиниць своїм друзям Бобу та Чарлі, відображала всі її деталі: адреса відправника, вихідні адреси, сума, час транзакції, хеш та підпис. Програма підтвердила, що ця транзакція валідна, і підпис дійсний.

Наступна транзакція, де Боб передав 30 одиниць Чарлі, також була представлена в тому ж форматі. Вона пройшла верифікацію так само, як і попередня, підтверджуючи свою валідність.

Після того, як я отримав підтвердження валідності транзакцій, програма перейшла до процесу майнінгу блоку. Вона успішно знайшла хеш для нового блоку, який відповідав заданій складності, і я отримав повідомлення, що блок був видобутий.

Далі блок був доданий до блокчейну, і я зміг побачити всю інформацію про блоки в ланцюгу. Перший блок, генезис-блок, містив дані, які я очікував, і не мав підпису, оскільки це початковий блок. Другий блок уже мав усі необхідні деталі, включаючи хеш, підпис та інші параметри. Я зрозумів, що він був створений на основі транзакцій, які я раніше створив.

Наприкінці програма перевірила валідність всього блокчейну і вивела, що він валідний. Це означало, що жоден з блоків не було змінено і дані залишалися незмінними, підтверджуючи цілісність всього ланцюга. Цей результат дав мені змогу усвідомити, як блокчейн може забезпечити безпеку та прозорість у системах, де важливо довіряти інформації.