САМОСТІЙНА РОБОТА № 1

Розробка та дослідження програмної моделі блокчейну засобами Python або Node.js

1. Мета роботи

Закріпити теоретичні знання та сформувати практичні навички розробки програмної моделі блокчейну. Навчитися реалізовувати криптографічне хешування, структуру блоків, формування ланцюга блоків, перевірку цілісності даних та механізми консенсусу Proof of Work і Proof of Stake.

2. Програмне забезпечення

Для виконання роботи необхідно використовувати одне з таких середовищ:

- Python 3.10 або новішої версії;
- Node.js 18 або новішої версії;
- середовище розробки Visual Studio Code або PyCharm.

Дозволяється використовувати стандартні криптографічні бібліотеки.

3. Теоретичні відомості

Блокчейн – це розподілена структура даних, у якій інформація зберігається у вигляді послідовності блоків. Кожен блок містить дані, власний криптографічний хеш та хеш попереднього блоку.

Основними компонентами програмної моделі блокчейну є:

1. **Block** – структура окремого блоку.
2. **Chain** – послідовність взаємопов'язаних блоків.
3. **Hash** – криптографічний відбиток вмісту блоку.
4. **Nonce** – числове значення, яке змінюється під час майнінгу.
5. **Difficulty** – параметр складності майнінгу.
6. **Proof of Work** – механізм консенсусу, що передбачає виконання обчислювальної роботи.
7. **Proof of Stake** – механізм консенсусу, у якому право участі у формуванні блоків залежить від частки володіння (stake).

Для обчислення хешу рекомендується використовувати SHA-256.

Приклад обчислення хешу мовою Python:

```
import hashlib

def calculate_hash(data):
    return hashlib.sha256(
        data.encode()
    ).hexdigest()

print(calculate_hash("Blockchain"))
```

Приклад структури блоку:

```
class Block:
    def __init__(self, index, data, previous_hash):
        self.index = index
        self.timestamp = time.time()
        self.data = data
        self.previous_hash = previous_hash
        self.nonce = 0
        self.hash = self.calculate_hash()

    def calculate_hash(self):
        block_data = (
            str(self.index) +
            str(self.timestamp) +
            str(self.data) +
            str(self.previous_hash) +
```

```
            str(self.nonce)
        )
        return hashlib.sha256(
            block_data.encode()
        ).hexdigest()
```

4. Загальне завдання

На основі обраного варіанта необхідно розробити програму, яка моделює роботу блокчейн-системи.

Програма повинна містити такі функціональні компоненти:

Завдання 1. Реалізація структури блоку.

Розробити клас `Block`, який містить:

- порядковий номер блоку;
- часову мітку створення;
- інформацію або транзакції;
- хеш попереднього блоку;
- власний хеш;
- параметр nonce.

Реалізувати метод обчислення SHA-256 хешу блоку.

Завдання 2. Реалізація блокчейну.

Розробити клас `Blockchain`, який забезпечує:

- створення початкового блоку Genesis Block;
- додавання нових блоків;
- збереження посилання на попередній блок;
- виведення всіх блоків;
- перевірку цілісності ланцюга.

Завдання 3. Реалізація Proof of Work.

Розробити алгоритм майнінгу, який:

- приймає параметр складності `difficulty`;
- змінює значення `nonce`;
- повторно обчислює хеш блоку;
- шукає хеш, що відповідає заданій складності;
- визначає кількість ітерацій та час майнінгу.

У цій роботі складність визначається кількістю нульових символів на початку хешу.

Завдання 4. Реалізація Proof of Stake.

Створити програмну модель набору валідаторів. Кожен валідатор повинен мати:

- ідентифікатор;
- ім'я;
- розмір stake;
- статус активності;
- кількість успішно створених блоків.

Реалізувати алгоритм випадкового вибору валідатора з урахуванням розміру його частки. Передбачити механізм штрафування (slashing) за спробу підтвердити некоректний блок.

Завдання 5. Тестування.

Перевірити:

- правильність обчислення хешів;
- додавання блоків;
- зміну даних у вже створеному блоці;
- виявлення порушення цілісності;
- роботу майнінгу;
- вибір валідатора;
- застосування штрафів.

5. Індивідуальні варіанти завдань

Кожен студент виконує один варіант відповідно до свого порядкового номера у списку академічної групи.

Для всіх варіантів обов'язковими є реалізація блокчейну, SHA-256, Proof of Work, Proof of Stake та перевірка цілісності. Індивідуальний варіант визначає предметну область даних, які зберігатимуться в блоках, і додаткову програмну логіку.

Варіант 1. Блокчейн для обліку фінансових транзакцій

Розробити блокчейн для збереження фінансових операцій між користувачами.

Необхідно реалізувати:

- створення транзакцій із зазначенням відправника, отримувача та суми;
- перевірку достатності балансу відправника;
- обчислення балансів користувачів на основі ланцюга транзакцій;
- об'єднання декількох транзакцій в один блок;
- перевірку неможливості повторного використання вже витрачених коштів.

Додатково: реалізувати виведення історії операцій конкретного користувача.

Варіант 2. Блокчейн для системи електронного голосування

Розробити систему збереження результатів голосування на основі блокчейну.

Необхідно реалізувати:

- створення переліку кандидатів;
- реєстрацію виборців за унікальними ідентифікаторами;
- створення транзакцій голосування;
- перевірку, чи голосував користувач раніше;
- запис результатів голосування до блоків;
- підрахунок голосів після завершення голосування.

Додатково: реалізувати перевірку неможливості повторного голосування одним виборцем.

Варіант 3. Блокчейн для системи відстеження товарів

Розробити блокчейн для контролю переміщення товарів між учасниками логістичного ланцюга.

Необхідно реалізувати:

- створення записів про товари;
- збереження унікального ідентифікатора товару;
- реєстрацію виробника, складу та отримувача;
- запис інформації про зміну місця перебування товару;
- перевірку послідовності переміщень;
- отримання повної історії руху товару.

Додатково: реалізувати пошук товару за його ідентифікатором.

Варіант 4. Блокчейн для системи обліку навчальних досягнень

Розробити систему збереження інформації про оцінки студентів.

Необхідно реалізувати:

- створення записів про студентів;
- збереження дисципліни, оцінки та дати оцінювання;
- додавання результатів навчання до блоків;
- пошук оцінок конкретного студента;
- обчислення середнього бала;
- виявлення спроби зміни раніше збереженої оцінки.

Додатково: реалізувати формування електронного звіту про успішність студента.

Варіант 5. Блокчейн для системи реєстрації медіафайлів

Розробити систему контролю цілісності цифрових файлів.

Необхідно реалізувати:

- реєстрацію файлів;
- обчислення SHA-256 хешу кожного файлу;
- збереження назви, розміру, дати створення та хешу;

- перевірку відповідності поточного хешу початковому;
- виявлення модифікації файлів;
- збереження історії реєстрацій у блокчейні.

Додатково: реалізувати перевірку файлу за його хешем.

Варіант 6. Блокчейн для системи обліку енергетичних ресурсів

Розробити блокчейн для збереження інформації про виробництво та споживання електроенергії.

Необхідно реалізувати:

- реєстрацію виробників та споживачів;
- створення транзакцій передачі електроенергії;
- зазначення кількості переданої енергії;
- перевірку доступного енергетичного ресурсу;
- розрахунок загального обсягу споживання;
- збереження історії операцій.

Додатково: реалізувати визначення найбільшого споживача електроенергії.

Варіант 7. Блокчейн для системи реєстрації цифрових сертифікатів

Розробити систему збереження та перевірки цифрових сертифікатів.

Необхідно реалізувати:

- створення сертифікатів із унікальними номерами;
- збереження інформації про власника та організацію-видавця;
- встановлення дати видачі та терміну дії;
- перевірку справжності сертифіката;
- реєстрацію відкликання сертифіката;
- виявлення підроблених або змінених записів.

Додатково: реалізувати пошук сертифіката за унікальним номером.

Варіант 8. Блокчейн для системи обліку оренди нерухомості

Розробити систему реєстрації договорів оренди.

Необхідно реалізувати:

- реєстрацію об'єктів нерухомості;
- збереження інформації про орендодавця та орендаря;
- створення записів про договори оренди;
- збереження суми та терміну оренди;
- реєстрацію платежів;
- перевірку цілісності історії договорів.

Додатково: реалізувати визначення активних договорів оренди.

Варіант 9. Блокчейн для системи обліку благодійних внесків

Розробити систему прозорого обліку благодійних платежів.

Необхідно реалізувати:

- створення благодійних проєктів;
- реєстрацію внесків;
- збереження суми, дати та ідентифікатора проєкту;
- обчислення загальної суми зібраних коштів;
- контроль цільового використання коштів;
- формування історії внесків.

Додатково: реалізувати статистику внесків за кожним проєктом.

Варіант 10. Блокчейн для системи обліку цифрових активів

Розробити систему реєстрації передачі цифрових активів між користувачами.

Необхідно реалізувати:

- створення цифрових активів з унікальними ідентифікаторами;
- реєстрацію власників;
- передачу активів між користувачами;
- перевірку права власності перед передачею;

- формування історії зміни власників;
- виявлення повторної передачі вже переданого активу.

Додатково: реалізувати отримання переліку активів конкретного користувача.

6. Контрольні експерименти

Під час тестування необхідно провести такі експерименти:

№	Експеримент	Очікуваний результат
1	Створення блокчейну	Створено Genesis Block
2	Додавання нового блоку	Блок додано до ланцюга
3	Зміна даних блоку	Цілісність порушено
4	Перевірка ланцюга	Виявлено некоректний блок
5	Майнінг із difficulty = 2	Знайдено відповідний хеш
6	Майнінг із difficulty = 3	Збільшено обсяг обчислень
7	Вибір валідатора	Обрано валідатора відповідно до stake
8	Некоректний блок	Застосовано slashing

У звіті потрібно навести фактичні результати експериментів, скриншоти та пояснення.