

Використання блокчейн технологій у фінансовому секторі

Васильєва Юлія Володимирівна, студентка
ORCID ID 0000-0002-8058-678X

Науковий керівник: Цеслів Ольга Володимирівна, к.т.н., доцент
КПІ ім. Ігоря Сікорського, Україна

З появою блокчейн технології, особливо її впровадженням у криптовалютах, відкрилися новітні шляхи для реформування фінансового сектору. Ця інноваційна технологія, яка забезпечує децентралізоване зберігання даних із високим рівнем безпеки, тепер активно інтегрується у різноманітні фінансові служби[1,2].

Блокчейн пропонує значне вдосконалення фінансових операцій через забезпечення їх прозорості, оптимізацію витрат на обробку даних та значне підвищення рівня безпеки. Незмінність даних, яку гарантує блокчейн, знижує ризики шахрайства, тоді як автоматизація процесів сприяє підвищенню загальної ефективності банківських та фінансових операцій[3,4].

Одним із яскравих прикладів використання блокчейну є його застосування у міжнародних банківських переказах. Так, блокчейн дозволяє значно скоротити час та фінансові витрати на транзакції, що зазвичай потребують складної взаємодії між різними фінансовими установами. Ця технологія забезпечує ефективність та прозорість платежів, знижуючи залежність від традиційних міжбанківських систем та їхніх високих комісій[5].

Для наочності, розглянемо реалізацію простого блокчейну на мові програмування Java. Цей приклад включає створення основного класу Block, який міститиме важливі елементи такі як дані транзакції, унікальний хеш блоку, а також хеш попереднього блоку для забезпечення послідовності ланцюга. Використання алгоритму SHA-256 для генерації хешу забезпечує надійність та безпеку даних. Процес створення ланцюга блоків (blockchain) включає ініціалізацію першого блоку (так званого genesis block) та додавання нових блоків, забезпечуючи зв'язок між ними через хеші. Валідація цього ланцюга полягає у перевірці послідовності хешів усіх блоків, що є критично важливим для підтримання цілісності та безпеки блокчейну[6].

Код:

```
import java.security.MessageDigest;
public class Block {
    private String hash;
    private String previousHash;
    private String data; // дані, наприклад, деталі транзакції
    private long timeStamp;
    public Block(String data, String previousHash ) {
        this.data = data;
        this.previousHash = previousHash;
        this.timeStamp = System.currentTimeMillis();
        this.hash = calculateHash();
    }
    public String calculateHash() {
        String calculatedhash = applySha256(
            previousHash +
            Long.toString(timeStamp) +
            data
        );
        return calculatedhash;
    }
    public static String applySha256(String input){
        try {
            MessageDigest digest = MessageDigest.getInstance("SHA-256");
            byte[] hash = digest.digest(input.getBytes("UTF-8"));
            StringBuffer hexString = new StringBuffer();
            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);
    }
}
public String getHash() {
    return hash;
}
public String getPreviousHash() {
    return previousHash;
}
}
import java.util.ArrayList;
public class Blockchain {
    public static ArrayList<Block> blockchain = new ArrayList<Block>();
    public static Boolean isChainValid() {
        Block currentBlock;
        Block previousBlock;
        for(int i = 1; i < blockchain.size(); i++) {
            currentBlock = blockchain.get(i);
            previousBlock = blockchain.get(i-1);
            if(!currentBlock.getHash().equals(currentBlock.calculateHash())){
                System.out.println(«Current hashes not equal»);
                return false;
            }
            if(!previousBlock.getHash().equals(currentBlock.getPreviousHash())) {
                System.out.println(«Previous hashes not equal»);
                return false;
            }
        }
        return true;
    }
    public static void main(String[] args) {
        blockchain.add(new Block(«First block», «0»));
        blockchain.add(new Block(«Second block», blockchain.get(blockchain.size()-1).getHash()));
        blockchain.add(new Block(«Third block», blockchain.get(blockchain.size()-1).getHash()));
        System.out.println(«Blockchain is Valid: « + isChainValid()»);
        for (Block block : blockchain) {
            System.out.println(«Block: «);
            System.out.println(«Hash: « + block.getHash());
            System.out.println(«Previous Hash: « + block.getPreviousHash());
            System.out.println();
        }
    }
}

```

Результати коду:

Blockchain is Valid: true

Block:

Hash: 8b693703d6a0e82a4179ce6e72f3ccaf598e23e57691b483a8a82ed9a546a2d1

Previous Hash: 0

Block:

Hash: 7517b63739422e75cfd58f90c3a6a9028cfd3a2e8e21ee0c8b998baac6ae20df

Previous Hash: 8b693703d6a0e82a4179ce6e72f3ccaf598e23e57691b483a8a82ed9a546a2d1

Block:

Hash: 5731885d340bb8572208af70b7106fddb6369115e021f239b2f6f0e9bbaa4715

Previous Hash: 7517b63739422e75cfd58f90c3a6a9028cfd3a2e8e21ee0c8b998baac6ae20df

Результати виконання програми відображають ланцюг блоків, де кожен наступний блок містить відомості про попередній блок, тим самим створюючи неперервний ланцюг. Це ілюструє, як в блокчейні забезпечується незмінність записів: зміна інформації в одному блоку вимагала б змін у всіх наступних блоках, що практично неможливо без виявлення. Такий підхід може бути застосований у фінансовому секторі для створення прозорих та безпечних систем міжнародних платежів, кредитування, страхування та інших банківських послуг. Завдяки цьому, блокчейн не лише підвищує ефективність фінансових операцій, але й забезпечує важливий рівень безпеки та відкритості, який є фундаментальним для довіри та стабільності у фінансовому світі [6]. Розвиток та впровадження блокчейн технологій має величезний потенціал для реформування та модернізації фінансового сектору. Ця технологія відкриває шлях для нових методів ведення бізнесу, сприяє ефективності взаємодії з клієнтами

та має потенціал значно впливати на глобальні фінансові системи, змінюючи традиційні підходи та регуляторні рамки.

Література:

1. Nakamoto, S. (2008). Bitcoin: A Peer-to-Peer Electronic Cash System.
2. Antonopoulos, A. M. (2014). Mastering Bitcoin: Unlocking Digital Cryptocurrencies.
3. Tapscott, D., & Tapscott, A. (2016). Blockchain Revolution: How the Technology Behind Bitcoin Is Changing Money, Business, and the World.
4. Mougayar, W. (2016). The Business Blockchain: Promise, Practice, and Application of the Next Internet Technology.
5. Drescher, D. (2017). Blockchain Basics: A Non-Technical Introduction in 25 Steps.
6. Bashir, I. (2018). Mastering Blockchain: Distributed ledger technology, decentralization, and smart contracts explained, 2nd Edition.