

ОНАЦЬКИЙ ВІТАЛІЙ

ЧНУ імені Петра Могили

<https://orcid.org/0009-0007-7207-8375>e-mail: vitalii.onatskyi@chmnu.edu.ua

САВІНОВ ВОЛОДИМИР

ЧНУ імені Петра Могили

<https://orcid.org/0000-0002-0862-5879>e-mail: eimkua2@gmail.com

РОЗРОБКА МЕТОДУ ФУНКЦІОНУВАННЯ ДЕЦЕНТРАЛІЗОВАНИХ КОМП'ЮТЕРНИХ СИСТЕМ ПРИ ВИРІШЕННІ ПРОБЛЕМИ СПАДКОЄМНОСТІ

У роботі проведено аналіз складнощів, пов'язаних з передачею майна, та огляд можливостей використання смарт-контрактів як ефективного інструменту для автоматизації цього процесу до спадкоємців у контексті децентралізованих систем. Запропоновано використання смарт-контракту у блокчейні для вирішення проблеми спадкоємності, що відкриває шлях до нових можливостей в сфері спадкування та управління спадковими активами. Вони забезпечують прозорість, автоматизацію та економічність, що робить їх привабливим інструментом для спадкоємності в децентралізованих комп'ютерних системах.

Ключові слова: розподілені системи, децентралізовані системи, блокчейн.

ONATSKYI VITALII, SAVINOV VOLODYMYR

Black Sea National University named after Petro Mohyla

DEVELOPMENT OF A METHOD FOR FUNCTIONING OF DECENTRALIZED COMPUTER SYSTEMS IN SOLVING THE PROBLEM OF INHERITANCE

This article explores the application of smart contracts in inheritance matters within the contemporary digital realm. The first part analyzes the complexities and traditional processes of inheritance in Ukraine, identifying various forms of inheritance based on testamentary and intestate succession laws. The subsequent sections highlight the advantages of utilizing smart contracts, emphasizing transparency, automation, and cost-effectiveness in managing inheritance. Additionally, the article discusses legal considerations and underscores the importance of ensuring the reliability of smart contracts. Overall, smart contracts represent an innovative approach to managing inheritance, offering new opportunities for efficient, transparent, and trustworthy asset distribution among heirs in decentralized systems. For the first time, the use of a smart contract in the blockchain was proposed to solve the problem of succession, which opens the way to new opportunities in the field of inheritance and management of inherited assets. They provide transparency, automation, and cost-effectiveness, making them an attractive tool for continuity in decentralized computing systems. Therefore, smart contracts represent an innovative approach to the management of inherited property, which creates new opportunities for efficient, transparent and trustful distribution of property between heirs. In addition, they can ensure the security and inviolability of the rights of the heirs, since all the terms of the inheritance distribution are fixed in the code of the contract, which cannot be changed, which excludes the possibility of its manipulation by any party. Thanks to blockchain technology, smart contracts can also provide security and necessary documentation to enforce inheritance rights to property, while avoiding the shortcomings of traditional inheritance management systems.

Keywords: distributed system, decentralized systems, blockchain.

Постановка проблеми

У традиційних системах вирішення спадкоємності процес передачі майна та прав наслідування часто потребує значних зусиль, часу та ресурсів. Крім того, існує ризик людських помилок, впливу третіх осіб та недостатньої прозорості у процесі розподілу спадкоємності. У контексті децентралізованих комп'ютерних систем існує потреба у розробці ефективного та безпечного методу автоматизованого вирішення спадкоємності.

Аналіз останніх джерел

Метою роботи є розробка та реалізація методу вирішення спадкоємності в децентралізованих комп'ютерних системах з використанням смарт-контрактів.

Виклад основного матеріалу

В сучасному цифровому світі технології блокчейну стають все більш важливим інструментом для розв'язання складних проблем. Однією з таких проблем є спадкоємність - процес передачі майна від померлого до спадкоємців. Традиційно цей процес може бути складним, дорогим і часом затратним через необхідність участі різних інтересів, виконання формальних процедур тощо.

Розглянемо приклад. Є батько, який на своєму власному шляху розвитку має криптографічний гаманець, що містить у собі певну кількість цифрових активів. Він, як батько, свідомо розуміє цінність та важливість майна, яке він накопичив, та має бажання передати його нащадкам у разі, якщо виникнуть непередбачені обставини, такі як його власна смерть. Проте син, ще молодий і нездатний самостійно керувати такими активами, не може бути безпосереднім одержувачем спадщини, але батько все ж бажає забезпечити, щоб активи були доступні синові у випадку будь-яких непередбачуваних обставин, навіть якщо він сам вже не буде з ним.

З виникненням смарт-контрактів, новий шлях вирішення спадкоємності відкривається - шлях, який може бути ефективним, прозорим і економічно доцільним.

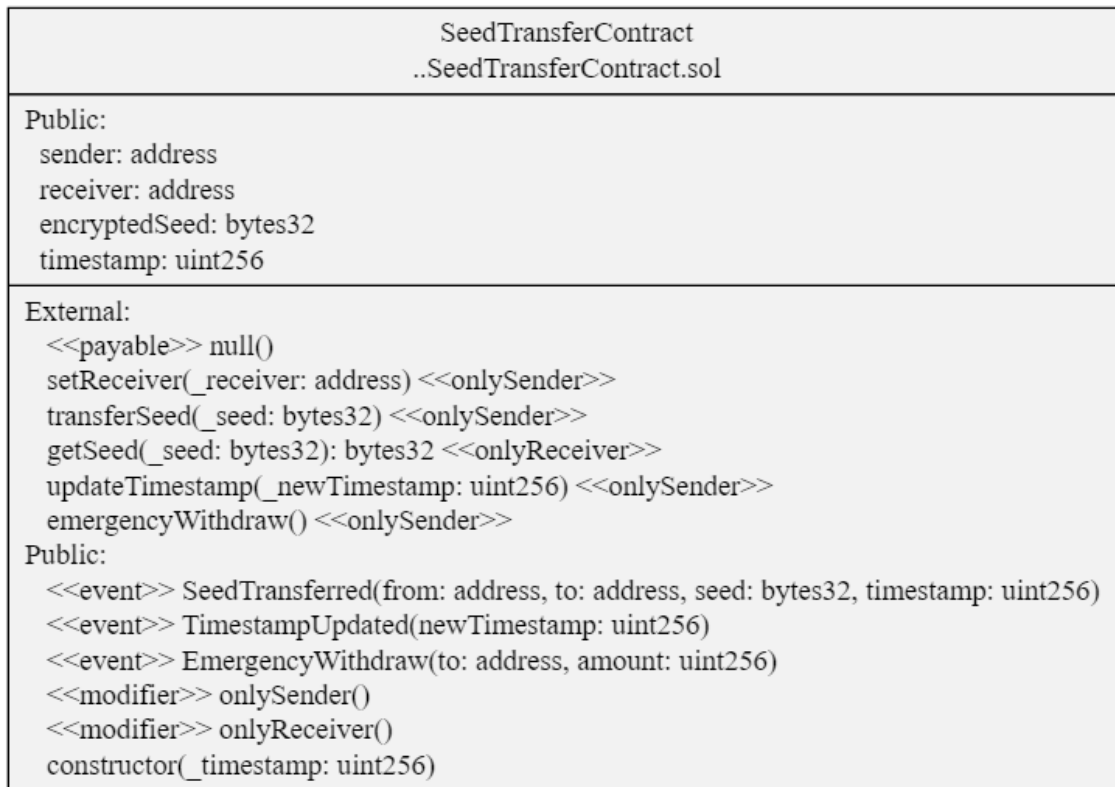


Рис. 2. UML діаграма класів смарт контракту SeedTransferContract

Таблиця 1

Опис кожного методу контракту, його призначення та умови виклику

Метод	Опис
constructor(uint256 _timestamp)	Конструктор контракту, приймає в якості аргументу часову мітку timestamp, яка вказує на момент, до якого не можна передавати насіння(seed).
setReceiver(address receiver)	Встановлює адресу отримувача, якому буде передано зашифровані насіння. Викликається тільки відправником (sender).
transferSeed(bytes32 _seed)	Передає зашифровані насіння відправнику (sender) до отримувача (receiver). Перевіряє часову мітку timestamp, щоб впевнитися, що пересилання вчасне.
getSeed(bytes32 _seed)	Розшифровує отримані насіння відправником (sender) за допомогою його приватного ключа. Викликається тільки отримувачем (receiver).
updateTimestamp(uint256 _newTimestamp)	Оновлює часову мітку timestamp на нове значення _newTimestamp. Перевіряє, що нова мітка часу більша за поточний час. Викликається тільки відправником (sender).
emergencyWithdraw()	У випадку екстрених ситуацій знімає всі кошти з контракту та надсилає їх на адресу отримувача (receiver), якщо поточний час більше або дорівнює timestamp, в іншому випадку надсилає їх на адресу відправника (sender).
receive()	Стандартна функція-приймач, приймає ether, викликається, коли контракт отримує ether без вказання методу.

Смарт-контракти пропонують новий підхід до управління спадковим майном, який базується на принципах децентралізації і прозорості. Їхнє використання дозволяє створювати довіру між учасниками процесу спадкоємності за допомогою криптографії та розподіленої природи блокчейну.

Переваги використання смарт-контрактів у справі спадкоємності очевидні:

Прозорість та недоторканість: Один з основних принципів блокчейну – це прозорість. Інформація, збережена в блокчейні, є публічною і не може бути змінена або видалена. Це означає, що усі учасники процесу спадкоємності можуть бачити всі транзакції та умови, зазначені у смарт-контракті, що забезпечує високий рівень довіри і недоторканості.

Автоматизація виконання умов: Смарт-контракти можуть автоматично самовиконуватися, встановлені в них, якщо всі необхідні умови виконані. Наприклад, вони можуть автоматично розподілити майно між спадкоємцями у випадку смерті власника.

Вартість та час: Використання смарт-контрактів може значно зменшити витрати і час, пов'язаний з процесом спадкоємності. Оскільки вони працюють без посередників, таких як нотаріуси або юристи, витрати на їх використання можуть бути набагато меншими.

Універсальність: Смарт-контракти можуть бути легко адаптовані до різних випадків спадкоємності та різних юрисдикцій, оскільки умови можуть бути програмовані з урахуванням конкретних потреб і правил.

Однією з основних переваг смарт-контрактів є їхня незмінність та автоматичність виконання умов. Коли умови спадкоємства встановлені в смарт-контракті, вони залишаються незмінними і виконуються автоматично під час відповідних подій, таких як смерть спадкоємця. Це дозволяє уникнути спорів і непорозумінь між спадкоємцями та забезпечує розподіл майна відповідно до встановлених правил.

Крім того, смарт-контракти забезпечують прозорість управління спадковим майном. Усі транзакції та умови, встановлені в контракті, є публічно доступними і перевіряємими, що дозволяє всім учасникам бачити процес і впевнитися в його справедливості.

На рис.2 зображено uml діаграму для смарт-контракту SeedTransferContract який дозволяє власнику передавати насіння (англ. seed) отримувачу за допомогою криптографічного шифрування, забезпечуючи безпеку та прозорість у процесі передачі.

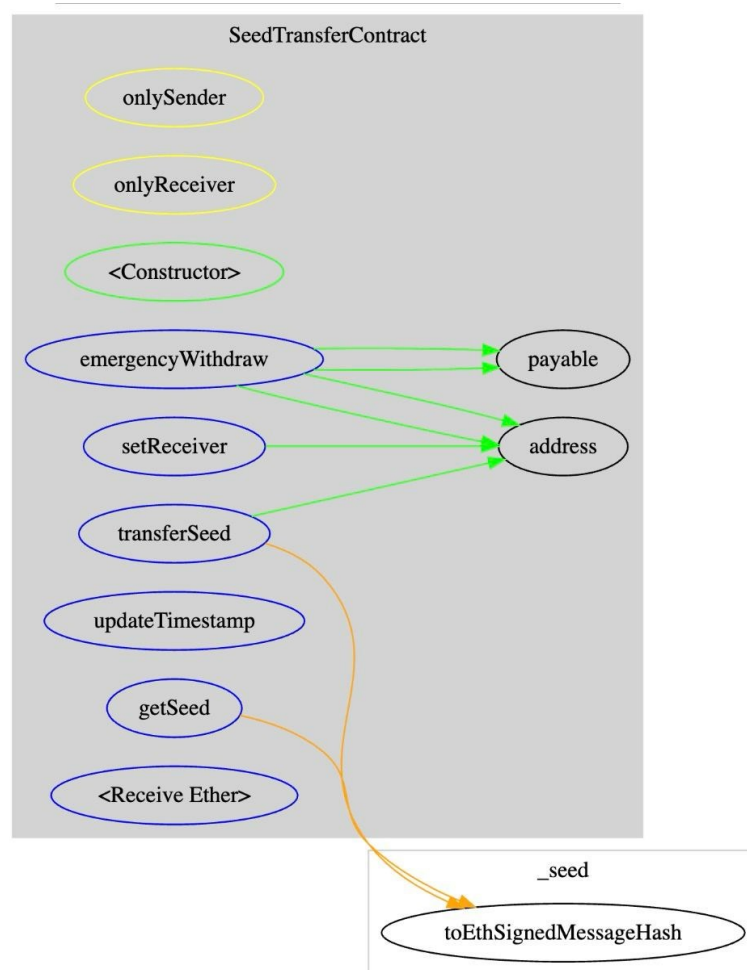


Рис. 3. UML діаграма компонентів смарт-контракту *SeedTransferContract*

Висновки

Вперше запропоновано використання смарт контракту у блокчейні для вирішення проблеми спадкоємності, що відкриває шлях до нових можливостей в сфері спадкування та управління спадковими активами. Вони забезпечують прозорість, автоматизацію та економічність, що робить їх привабливим інструментом для спадкоємності в децентралізованих комп'ютерних системах. Отже, смарт-контракти представляють собою інноваційний підхід до управління спадковим майном, який створює нові можливості для ефективного, прозорого і довірчого розподілу майна між спадкоємцями. Крім того, вони можуть забезпечити безпеку та недоторканість прав спадкоємців, оскільки усі умови спадкового розподілу фіксуються в коді контракту, який неможливо змінити, що виключає можливість його маніпулювання з боку будь-якої сторони. Завдяки технології блокчейн, смарт-контракти також можуть забезпечити безпеку та необхідну документацію для виконання спадкових прав на майно, уникнувши при цьому недоліків традиційних систем управління спадком.

Література

1. Solidity language support and visual security auditor for Visual Studio Code. <https://github.com/Consensys/vscode-solidity-auditor>
2. A set of utilities for exploring Solidity contracts. <https://github.com/Consensys/surya>
3. Що таке смарт-контракти і які принципи їх роботи? <https://lexinform.com.ua/yuridychna-praktyka/shho-take-smart-kontrakty-i-yaki-pryntsyry-yih-roboty/>
4. Блокчейн і децентралізовані системи : навч. посібник для студ. закладів вищ. освіти : в 3 частинах. Ч. 1 / П. Кравченко, Б. Скрябін, О. Дубіна. – Харків : ПРОМАРТ, 2019.
5. Кайхан Эрджиес. Розподілені системи реального часу. Теорія і практика. 2020.

References

1. Solidity language support and visual security auditor for Visual Studio Code. <https://github.com/Consensys/vscode-solidity-auditor>
2. A set of utilities for exploring Solidity contracts. <https://github.com/Consensys/surya>
3. Shcho take smart-kontrakty i yaki pryntsyry yikh roboty? <https://lexinform.com.ua/yuridychna-praktyka/shho-take-smart-kontrakty-i-yaki-pryntsyry-yih-roboty/>
4. Blokchein i detsentralizovani systemy : navch. posibnyk dlia stud. zakladiv vyshch. osvity : v 3 chastynakh. Ch. 1 / P. Kravchenko, B. Skriabin, O. Dubina. – Kharkiv : PROMART, 2019.
5. Kaikhan Erdzhyes. Rozpodileni systemy realnoho chasu. Teoriia i praktyka. 2020.