**Міністерство блокчейну і криптовалюти України**

**Практичне завдання № 2**

**На Тему:  
«**Реалізація власного блокчейну**»**

**Виконав:**

L1MD5\_171

L1MD5\_285

L1MD5\_150

L1MD5\_100

L1MD5\_95

**Перевірив:**

Distributedlab

2022 р.

**Тема:** Реалізація блокчейну.

**Мета:** Для реалізації власного блокчейну сформувати теоретичні відомості, визначити сферу застосування.

**Хід роботи**

**Сфера застосування блокчейну**

Перш ніж визначити сферу застосування блокчейну, пригадаємо його одну із найбільших переваг, а саме децентралізацію. Тобто це інформація, яка рівномірно розподілена по всіх вузлах мережі, дублюючись у кожному із них. При такому підході, доки хоча б один вузол працездатний, інформація не втратиться, як це могло би бути, наприклад, з інтернет мережею.

Враховуючи, таку вище наведену особливість блокчейну, децентралізацію можна застосувати у такій, на перший погляд незвичній сфері, як у мережі спортивних залів.

**Плюси:**

* Незалежність залів один від одного
* Незалежність від валюти країни, окрім USD.
* При розширені мережі спортивного залу за кордоном, клієнтові не доведеться розмінювати фіат, щоб заплатити за абонемент. Тобто, якщо клієнт відвідував спортивний заклад і купував його монети за UAH, то поїхавши згодом за кордон, де є спортивні заклади такої ж мережі, він має змогу користуватися монетами, придбаними за UAH.
* Можливість формування єдиної клієнтської бази даних.
* Можливість формування самообслуговування клієнтів.

**Мінуси:**

* Залежність спортивного закладу від електроенергії та інтернету
* Залежність клієнтів від інтернету та обов’язкової наявності смартфонів.
* Повільніша обробка транзакцій у порівнянні із платіжними системами Visa, Mastercard и PayPal.

**Технічне завдання та реалізація**

Метою технічного завдання є об’єднання мережі спортивних закладів із створенням спільної системи оплати та клієнтської бази даних.

Клієнти не мають змоги майнити дану монету спортивного закладу, а лише обмінюватись монетами, без права на їх продаж. Тобто накопичення монет задля їх подальшої реалізації по дорожчій ціні неможливе, а отже якщо клієнт придбав монету, то в очікуваному результаті буде відвідувати спортивний заклад, обмінюючи монету на послуги.

Під час реалізації найпростішого блокчейну для мережі спортивних закладів, було використано мову програмування С# із середовищем розробки Visual Studio Code 2022.

Функціонал розробленого блокчейну включає в себе:

* Створення 10 блоків з випадковою кі-стю монет
* Верифікація блоків

Після процесу розробки було проведено також тестування програмного коду:

* Тестування блокчейну без змін
* Тестування блокчейну зі зміною суми в одному з блоків
* Тестування блокчейну зі зміною суми в одному з блоків і регенерацією хеша

**Структура блокчейну**

**Опис коду**

[Зображення, що містить текст

Автоматично згенерований опис](https://github.com/HighError/Course-Distributed-Lab/blob/main/Code/Images/1.png)

Клас Chain = один блок.

В собі цей клас має такі параметри:

* Кі-сть монет
* Хеш
* Хеш попереднього блоку

Зображення, що містить текст

Автоматично згенерований опис

Метод CreateBlock класу Chain створює новий блок.

Цей метод приймаю кі-сть монет та хеш попереднього блоку.

Зображення, що містить текст

Автоматично згенерований опис

Для генерації хешу, використовується метод GetHash, який за допомогою методу GetMd5Hash (див. нижче) шифрує текст, який представляє з себе конкатенацію кі-сті монет, попереднього хешу та індексу в md5.

[Зображення, що містить текст

Автоматично згенерований опис](https://github.com/HighError/Course-Distributed-Lab/blob/main/Code/Images/4.png)

Метод GetMd5Hash шифрує заданий текст в md5.

[[Зображення, що містить текст

Автоматично згенерований опис](https://github.com/HighError/Course-Distributed-Lab/blob/main/Code/Images/5.png)](https://github.com/HighError/Course-Distributed-Lab/blob/main/Code/Images/5.png)

В основному класі знаходиться метод VerificationBC, який приймає блоки.

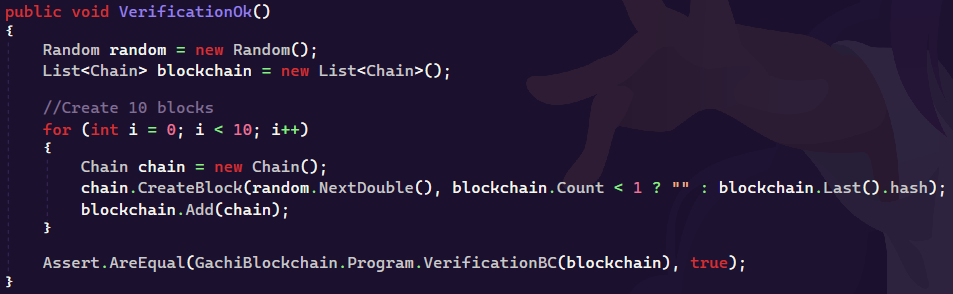
Даний метод перевірає цілісність блокчейну.

Якщо хоча б в одному блоці буде порушене його хеш значення або хеш значення поперенього блоку, метод поверне false і в такому випадку, блокчейн не буде верифікованим.

**Тестування**

В данному проекті ми реалізували 3 тестування.

тестування.

[](https://github.com/HighError/Course-Distributed-Lab/blob/main/Code/Images/6.png)

Перше тестування - VerificationOk перевіряє чи проходить перевірку блокчейн, який не піддавався змінам після його створення.

**Очікуваний результат:** **True**

Зображення, що містить текст

Автоматично згенерований опис

Друге тестування - VerificationError1 перевіряє чи пройде перевірку блокчейн, який піддався зміні кі-сті монет у одному з блоків.

**Очікуваний результат: False**

Зображення, що містить текст

Автоматично згенерований опис

Третє тестування - VerificationError2 перевіряє чи пройде перевірку блокчейн, який піддався зміні кі-сті монет у одному з блоків та в якому перегенерували хеш.

**Очікуваний результат: False**

**Висновок**

Під час реалізації даного практичного завдання, ми обрали та описали сферу під реалізацію блокчейну, який було реалізовано на мові програмування С#, із урахуванням функціоналу та вимог. Після реалізації було проведено тестування на працездатність програмного продукту.