МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ

Факультет кібербезпеки комп’ютерної та програмної інженерії

Кафедра безпеки інформаційних технологій

**Blockchain-технології**

**Лабораторна робота №8**

Тема:  Смарт контракти

Тривалість заняття: 90 хв.

Київ 2023

**Тема:** Смарт-контракти.

**Мета:** Навчитись створювати смарт-контракти.

**Теоретичні відомості**

Смарт-контракт ([англ.](https://uk.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D1%96%D0%B9%D1%81%D1%8C%D0%BA%D0%B0_%D0%BC%D0%BE%D0%B2%D0%B0) smart contract — «розумний контракт») — різновид угоди в формі закодованих математичних алгоритмів, де укладення, зміни, виконання і розривання можна виконати лише з використанням комп'ютерних програм ([блокчейн](https://uk.wikipedia.org/wiki/%D0%91%D0%BB%D0%BE%D0%BA%D1%87%D0%B5%D0%B9%D0%BD" \o "Блокчейн)-платформ) у рамках мережі [Інтернет](https://uk.wikipedia.org/wiki/%D0%86%D0%BD%D1%82%D0%B5%D1%80%D0%BD%D0%B5%D1%82" \o "Інтернет).

В реальному світі широке охоплення в реалізації смарт-контрактів здобула DAO, [розподілена автономна організація](https://uk.wikipedia.org/w/index.php?title=%D0%A0%D0%BE%D0%B7%D0%BF%D0%BE%D0%B4%D1%96%D0%BB%D0%B5%D0%BD%D0%B0_%D0%B0%D0%B2%D1%82%D0%BE%D0%BD%D0%BE%D0%BC%D0%BD%D0%B0_%D0%BE%D1%80%D0%B3%D0%B0%D0%BD%D1%96%D0%B7%D0%B0%D1%86%D1%96%D1%8F&action=edit&redlink=1" \o "Розподілена автономна організація (ще не написана)) для венчурного фінансування, яка була запущена у травні 2016 року.

Першою країною, яка законодавчо закріпила розумні контракти стала [Білорусь](https://uk.wikipedia.org/wiki/%D0%91%D1%96%D0%BB%D0%BE%D1%80%D1%83%D1%81%D1%8C) у 2017 році.

Смарт-контракт — це «комп'ютеризований протокол транзакцій, який виконує умови контракту». Блокчейн, на основі смарт-контракта, видно всім користувачам зазначеного блоку. Проте, це призводить до помилок, в тому числі дірок в системі безпеки, які можуть побачити всі користувачі, але їх не можна виправити швидко. Таким чином, була успішно виконана атака на DAO в червні 2016 року вартістю 50 млн USD у Ethereum, в той час, як розробники намагалися прийти до вирішення цієї проблеми. Обробка задачі на блокчейні вимагала часу, за який хакер може отримати доступ до ресурсів, і зняти кошти з DAO контракту.

Смарт-контракти Ethereum розробляються на одній з мов, спроектованих для трансляції в байт-код віртуальної машини Ethereum — Solidity (схожий на С або JavaScript), Vyper і Serpent (схожі на Python), LLL (низькорівнева версія Lisp), Mutan (заснований на Go).

Сторони підписують розумний контракт, використовуючи методи, аналогічні підписанню перерахунку коштів в криптовалютні мережі, що існують в наш час. Після підписання сторонами, контракт зберігається в блокчейні і вступає в силу. Для забезпечення автоматизованого виконання зобов'язань договору обов'язково потрібно середовище існування (ноди блокчейна Ethereum), яке дозволяє повністю автоматизувати виконання пунктів контракту. Це означає, що розумні контракти зможуть існувати тільки всередині середовища, що має безперешкодний доступ виконуваного коду до об'єктів розумного контракту.

Всі умови контракту повинні мати програмний опис і ясну логіку виконання. У зв'язку з цим перші розумні контракти мають завдання формалізації найбільш простих взаємин, що складаються з невеликої кількості умов. Маючи безперешкодний доступ до об'єктів контракту, розумний контракт відстежує за вказаними умовами досягнення або порушення пунктів і приймає самостійні рішення, ґрунтуючись на запрограмованих умовах. Таким чином, основний принцип розумного контракту полягає в повній автоматизації і правдивості виконання договірних відносин.

Для того, щоб розумні контракти могли існувати, потрібні певні умови:

* Використання широко поширених методів електронного підпису на основі публічних і приватних ключів (асиметричне шифрування).
* Існування відкритих, децентралізованих і довірчих сторонам контракту баз даних для виконуваних транзакцій, робота яких повністю виключає людський фактор. Як приклад: блокчейн в Bitcoin.
* Децентралізація середовища виконання розумного контракту. Як приклад: Ethereum, Codius, Counterparty.
* Правдивість джерела цифрових даних. Як приклад: кореневі центри сертифікації SSL в базах сучасних інтернет-браузерів.

Об'єкти розумного контракту

Підписанти — сторони розумного контракту, які беруть або відмовляються від умов з використанням електронних підписів. Прямим аналогом є підпис відправника коштів в мережі Bitcoin, яка підтверджує внесення транзакції в ланцюжок блоків.

Предмет договору. Предметом договору може бути тільки об'єкт, що знаходиться всередині середовища існування розумного контракту, або ж повинен забезпечуватися безперешкодний, прямий доступ розумного контракту до предмету договору без участі людини.

Умови. Умови розумного контракту повинні мати повний математичний опис, який можливо запрограмувати в середовищі існування розумного контракту. Саме в умовах описується логіка виконання пунктів предмета договору.

Сабо передбачає, що смарт-контракт — це інфраструктура, яка може бути реалізована шляхом тиражованих реєстрів активів та виконання контрактів з використанням криптографічного хеш-ланцюга і Візантійської відмовостійкої винагороди. Кожен вузол у мережі рівноправних вузлів виступає як заголовок реєстру і умовна порука, що виконує зміни права власності та автоматично відзначає правила, що регулюють ці операції, і перевіряє ту ж роботу інших вузлів. Аскемос реалізував цей підхід в 2002 році, використовуючи схему як мову сценаріїв контракту.

Криптовалюти, такі як bitcoin впровадили спеціальні випадки таких реєстрів, де майно гроші. Bitcoin і багато з його спін-оффів містять механізми для того, щоб мати більше можливостей спільної власності і виконання контрактів. Код підтримки це прихована частина протоколу Bitcoin, на основі імовірної та анонімної (за рахунок доказу роботи на основі візантійського) винагороди.

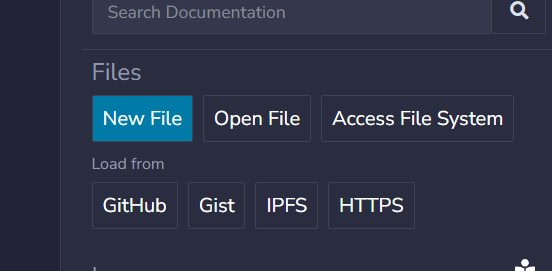
Одна з пропозицій для використання Bitcoin для реєстрації активів і виконання контракту називається «кольорові монети». Реєстрація доменного імені реєстру здійснюється в Namecoin; реєструємо назви для потенційно довільних форм власності, поряд з виконанням контракту, реалізовані в Crypti, Ripple, Mastercoin і Ethereum. NXT реалізує право власності на доказі власності, частки в базовій валюті.

Смарт-контракти можуть бути реалізовані за допомогою рікардіанському шаблону проектування контракту.

Додатки можуть включати в себе фінансові інструменти, такі як облігації, акції та похідні, договорів по забезпеченню, а також інші документи і угоди, де вузли можуть стежити за розвитком подій, на яких зумовлені розумні правила контракту. Переваги смарт-контракту як еквіваленту звичайного фінансового інструменту гіпотетично включають зведення до мінімуму ризику контрагента, скорочення термінів розрахунків, а також підвищення прозорості. UBS експериментував з «розумними» облігаціями, які використовують bitcoin блокчейн, в якому платіжні потоки гіпотетично могли б бути повністю автоматизовані, створюючи самостійно оплачуваний інструмент.

**Завдання**

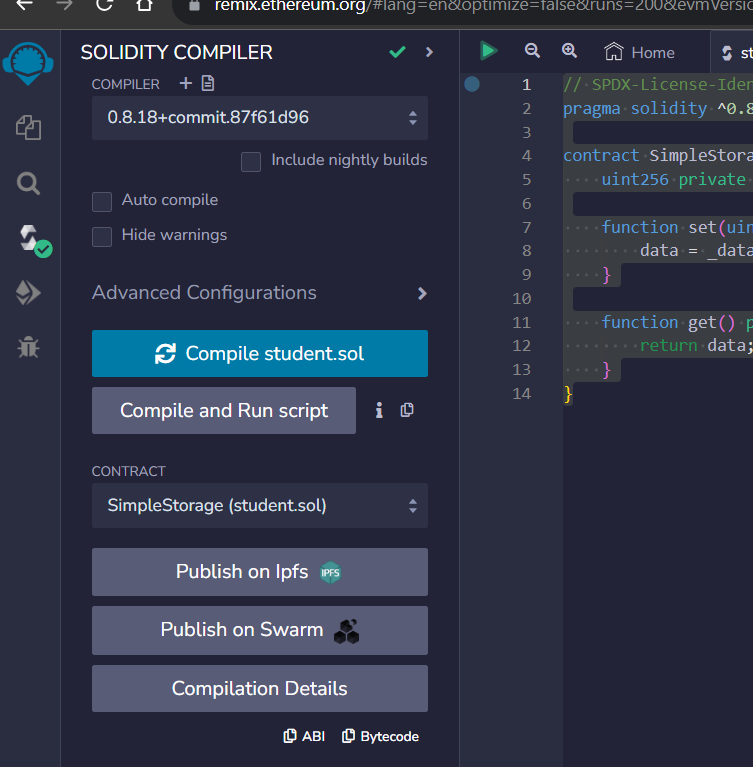
1. Перейдіть за цим посиланням <https://remix.ethereum.org/#lang=en&optimize=false&runs=200&evmVersion=null&version=soljson-v0.8.18+commit.87f61d96.js>
2. Натисніть на кнопку New File, та назвіть файл своєю фамілією



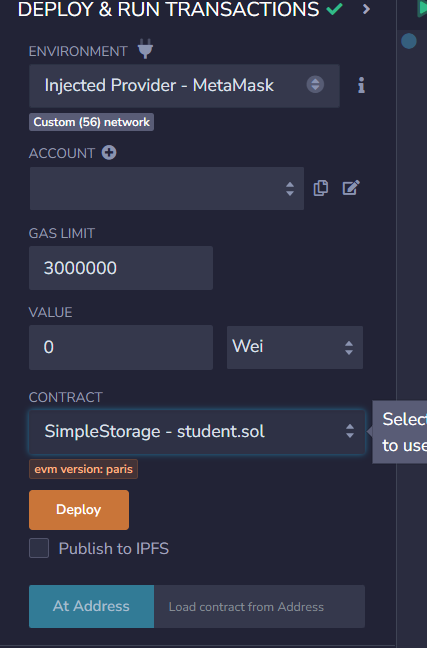
1. В створений вами файл вставте наступний текст:
2. // SPDX-License-Identifier: MIT
3. pragma solidity ^0.8.0;
4. contract SimpleStorage {
5. uint256 private data;
6. function set(uint256 \_data) public {
7. data = \_data;
8. }
9. function get() public view returns (uint256) {
10. return data;
11. }

}

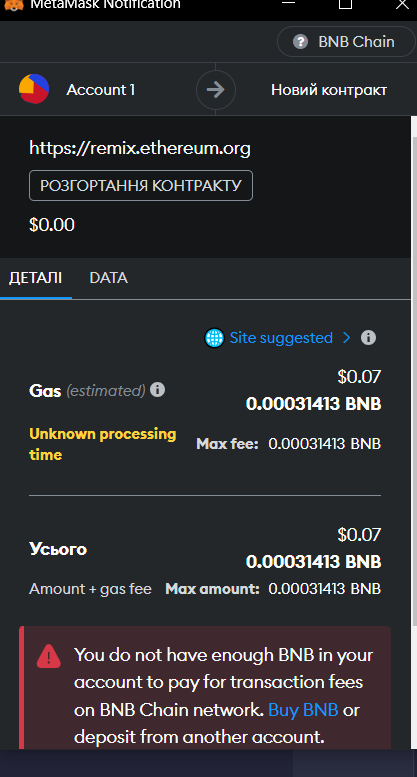
4. Далі перейдіть на вкладку SOLIDITY COMPILER, виберіть файл який було вами створено, та скомпілюйте його.



1. Перейдіть на вкладку "Deploy & Run Transactions" у Remix.
2. У розділі "Environment", виберіть середовище виконання. Ви можете використовувати JavaScript VM (вбудований у Remix, для локальної розробки), Injected Web3 (для підключення до вашого гаманця, такого як MetaMask), або інші варіанти. В нашому випадку вибираємо наступне, як на рисунку



1. У розділі "Deploy", виберіть смарт-контракт, який був скомпільований на попередньому кроці. Також вам потрібно буде у вікні яке відкриється з’єднати це середовище розробки з вашим гаманцем створеним в попередній лабораторній роботі, Для цього просто натисніть кнопку з’єднати.
2. Натисніть "Deploy", щоб розгорнути смарт-контракт. Якщо ви використовуєте Injected Web3, MetaMask запросить ваше підтвердження для виконання транзакції.
3. Після успішного розгортання смарт-контракту ви отримаєте адресу контракту і зможете взаємодіяти з ним через Remix.
4. На рисунку нижче зображено успішно виконану операцію з розгортання смарт-контракту.



1. Дайте відповідь на контрольні питання.
2. Занесіть пророблену вами роботу до звіту з використанням скріншотів.
3. Зробіть висновки та занесіть їх до звіту.

**Контрольні питання**

1. Що таке смарт-контракт?
2. За яким принципом працює смарт-контракт?
3. Які є об’єкти смарт-контракту?
4. Які умови потрібні для того, щоб смарт-контракти могли існувати?
5. Яким чином можуть бути реалізовані смарт контракти?