Міністерство освіти і науки України

Національний університет “Києво-Могилянська академія”

Факультет інформатики

Кафедра інформатики

Магістерська робота

Освітній ступінь: магістр

на тему: **“Моделювання токен економіки”**

Виконав: студент 2 року навчання
Спеціальності
122 Комп’ютерні науки
Декрет Владислав Володимирович
Керівник Ігнатенко О. П.
Доктор фізико-математичних наук

Рецензент Глибовець А. М.
Доктор технічних наук
Магістерська робота захищена
з оцінкою _____
Секретар ЕК

_____
«___» _____2024

Київ-2024

Міністерство освіти і науки України

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ “КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ”

Кафедра інформатики факультету інформатики

ЗАТВЕРДЖУЮ

Зав. кафедри інформатики

к.ф-м.н., доц. Гороховський С.С.

_____

«___» _____2024

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ

ДЛЯ ДИПЛОМНОЇ РОБОТИ СТУДЕНТУ

Декрету Владиславу Володимировичу

Тема роботи: “Моделювання токен економіки”

Керівник роботи: Ігнатенко О.П., доктор фізико-математичних наук

Завдання:

Проект полягає в аналізі програм стимулювання ліквідності (Liquidity Mining Programs), які є механізмом у світі децентралізованих фінансів (DeFi) та криптовалют для заохочення людей надавати ліквідність децентралізованим

біржам або платформам автоматичного маркет-мейкінгу (AMM). Ці програми часто використовуються на блокчейн-мережах, таких як Ethereum або Binance Smart Chain, і набули популярності як спосіб стимулювання ліквідності та заохочення торгівлі на цих платформах. Ось як зазвичай працює програма стимулювання ліквідності:

1. **Децентралізована біржа або AMM**: Програми часто асоціюються з децентралізованими біржами (DEX) або AMM, такими як Uniswap, SushiSwap або PancakeSwap. Ці платформи дозволяють користувачам торгувати різними криптовалютами без потреби у централізованому посереднику.
2. **Надання ліквідності**: Користувачі можуть надавати ліквідність цим платформам, вкладаючи пари токенів у ліквідні пули. Ці пули використовуються для забезпечення торгівлі на платформі.
3. **Винагородні токени**: У програмі стимулювання ліквідності оператор біржі або AMM зазвичай надає винагороди у вигляді своєї корінної криптовалюти або токенів. Ці винагороди даються користувачам, які надають ліквідність платформі. Кількість отриманих винагород часто пропорційна кількості наданої ліквідності.
4. **Стейкінг або блокування**: Користувачі, які хочуть взяти участь у програмі, зазвичай повинні застейкати або заблокувати свої токени провайдерів ліквідності (часто називаються LP токенами) у специфічному контракті або механізмі. Це запобігає користувачам передчасному вилученню своєї ліквідності, оскільки вони можуть втратити свої винагороди.
5. **Механізм розподілу**: Винагороди розподіляються через регулярні інтервали, такі як щодня або щотижня. Розподіл базується на таких факторах, як кількість наданої ліквідності, тривалість надання, та специфічні правила програми.
6. **Фармінг винагород**: Провайдери ліквідності можуть отримувати доходи на своїх активах не тільки від комісій за торгівлю, але й від додаткових винагородних токенів, розподілених через програму стимулювання ліквідності. Цю практику часто називають "фармінгом винагород".

Задача полягає в побудові моделі для аналізу:

1. **Аналіз кількох treatment груп**: Аналіз впливу різних рівнів або типів стимулів на кілька treatment груп може дати уявлення про те, як різні структури стимулів впливають на поведінку пулів.
2. **Аналіз довгострокового впливу**: Продовження терміну аналізу може висвітлити довгострокові впливи стимулів. Чи повертаються пули зрештою до своєї

поведінки до стимулу, або ж вони зберігають деякі здобутки навіть довго після закінчення стимулів?

3. **Аналіз індивідуальної поведінки**: Розуміння індивідуальної поведінки у пулах може виявити патерни або спільні характеристики серед учасників, які непропорційно сприяють збільшенню TVL або комісій.
4. **Кореляція з зовнішніми факторами**: Дослідження того, як зовнішні фактори, такі як загальні умови ринку або специфічні події, впливають на ефективність стимулів, може надати цінний контекст для інтерпретації результатів програми стимулів.
5. **Альтернативні метрики**: Врахування інших метрик, як-от кількість активних учасників у пулі, кількість транзакцій або середній розмір транзакції, може надати більш повне уявлення про продуктивність пулу та його реакцію на стимули.
6. **Моделювання прогнозування**: На основі даних створення моделі для прогнозування впливу майбутніх програм стимулювання може бути цінним інструментом для протоколів, які планують свою наступну програму стимулювання ліквідності.

Досліджуючи ці аспекти детальніше, ми можемо продовжувати удосконалювати та оптимізувати програми стимулювання ліквідності, максимізуючи їх ефективність та користь як для провайдерів ліквідності, так і для ширшого використання екосистеми DeFi.

Зміст ТЧ до курсової роботи:

Аналіз результатів

Висновки

Використана література

Додатки

ГРАФІК ПІДГОТОВКИ ДИПЛОМНОЇ РОБОТИ ДО ЗАХИСТУ

	Назва етапу магістерської роботи	Термін виконання	Примітка
1.	Отримання теми магістерської роботи.	26.10.2023	
2.	Пошук тематичної літератури.	10.11.2023	
3.	Ознайомлення з тематичними матеріалами та побудова структури практичної та теоретичної частин курсової роботи.	25.11.2023	
4.	Написання вступу та змісту роботи.	20.12.2023	
5.	Ознайомлення з існуючими принципами та підходами вирішення проблеми, пошук алгоритмів, що задовольняють вимоги.	25.01.2024	
6.	Створення застосунку ‘моделювання токен економіки’.	19.02.2024	
7.	Тестування роботоспроможності створеного застосунку та внесення змін.	15.03.2024	
8.	Написання аналізу досліджень базуючись на отриманих результатах.	22.04.2024	
9.	Форматування курсової роботи відповідно	23.05.2024	

	до визначених вимог.		
10.	Створення презентації та написання доповіді для захисту магістерської роботи.	28.05.2024	
11.	Узгодження попередньої версії роботи з керівником.	01.06.2024	
12.	Внесення змін до курсової роботи відповідно до зауважень наукового керівника.	03.06.2024	
13.	Захист магістерської роботи.	12.06.2024	

Графік узгоджено «___» _____ 2024 р.

Науковий керівник Ігнатенко О.П.

Виконавець магістерської роботи Декрет В.В.

Зміст

Анотація

При моделюванні токену економіки, зосередимось на дослідженні впливу різних параметрів на формування ціни токену у децентралізованих біржах. У роботі розглядаються такі ключові аспекти, як розмір пулу ліквідності, попит на токен, дохідність та обмеження, що накладаються на інвесторів.

Основною метою є дослідження впливу параметрів на стабільність та зростання ціни токену. Отримані результати можуть бути застосовані для вдосконалення стратегій випуску та торгівлі токенами у децентралізованих фінансових системах.

Вступ

У сучасному світі децентралізовані фінансові системи набувають все більшої популярності завдяки своїй прозорості, безпеці та відсутності посередників. Однією з ключових складових таких систем є токени – цифрові активи, що використовуються для здійснення різноманітних фінансових операцій. Успішне функціонування та зростання вартості токенів значною мірою залежать від правильного моделювання та управління їх економікою.

Ця магістерська робота присвячена моделюванню токену економіки з акцентом на вивченні впливу різних параметрів на формування ціни токену у децентралізованих біржах. Зокрема, розглядаються такі важливі аспекти, як розмір пулу ліквідності, попит на токен та обмеження, що накладаються на інвесторів. Правильне розуміння цих факторів та їх взаємодії дозволить створити стабільну та ефективну токеноміку, що сприятиме довгостроковому зростанню вартості токену.

У рамках дослідження було створено додаток, який дозволяє моделювати та відслідковувати динаміку ціни токену відносно зазначених параметрів. Цей інструмент надає можливість здійснювати детальний аналіз та прогнозування ринкової поведінки токенів, що є надзвичайно важливим для прийняття стратегічних рішень у сфері децентралізованих фінансів.

Мета роботи

Метою даної магістерської роботи є дослідження та моделювання впливу різних параметрів на формування ціни токену у децентралізованих біржах. Досягнення цієї мети передбачає вирішення таких завдань:

1. Аналіз існуючих систем децентралізованих фінансів та дослідження поведінки токену в них.
2. Аналіз обмежень, що накладаються на інвесторів, та їх вплив на ціну токену.
3. Розробка математичної моделі, яка враховує вказані параметри та дозволяє прогнозувати ринкову поведінку токену.
4. Створення програмного забезпечення для відслідковування динаміки ціни токену відносно обраних параметрів.
5. Дослідження впливу розміру пулу ліквідності на стабільність та динаміку ціни токену (перший сценарій).
6. Дослідження впливу попиту на токен на його ринкову вартість (другий сценарій).
7. Дослідження впливу попиту при різних початкових розмірів пулу ліквідності на динаміку ціни токену (третій сценарій).
8. Аналіз результатів для підтвердження ефективності розробленої моделі.

Виконання зазначених завдань дозволить розробити рекомендації щодо оптимального управління токеном економіки та сприятиме підвищенню стабільності та вартості токену на ринку децентралізованих фінансів.

Основні поняття

Токен економіки

Визначення

Токен економіки відноситься до економічної системи або моделі, яка використовується в рамках блокчейн проектів, криптовалют і децентралізованих додатків. Вона охоплює всі аспекти створення, розподілу, використання та управління токенами або цифровими активами, що служать як засіб обміну, стимулювання участі у системі, винагороди за досягнення певних цілей або виконання специфічних дій.

Основні аспекти

1. **Емісія токенів**: нові токени створюються або випускаються. Це може включати в себе фіксовану кількість токенів, інфляційні моделі або алгоритмічні методи, що контролюють пропозицію токенів.
2. **Розподіл токенів**: як токени розподіляються між учасниками екосистеми, включаючи первинні продажі, нагороди за участь у мережі або інші форми розподілу.
3. **Використання токенів**: токени можуть бути використані в екосистемі, такі як голосування, оплата послуг, стейкінг для забезпечення мережі або як засоби обміну.
4. **Управління**: дозволяє власникам токенів впливати на рішення в рамках екосистеми, зміни в протоколі або в напрямках розвитку проекту.
5. **Економічні стимули**: мотивує учасників діяти в інтересах мережі, підтримувати її безпеку, розвиток та зростання.

Мета

Токен економіки спрямований на створення справедливої та стійкої системи, що сприяє залученню та утриманню користувачів, забезпечує ефективне управління ресурсами мережі та стимулює зростання та розвиток екосистеми. Важливою частиною токену економіки є забезпечення балансу між інфляцією та дефляцією активів, а також розробка механізмів, які підтримують довгострокову цінність для власників токенів.

Блокчейн

Визначення

Блокчейн — це технологія, яка дозволяє зберігати дані у вигляді послідовності блоків, що з'єднані між собою. Кожен блок містить інформацію про попередній блок, що формує ланцюжок, створюючи високий рівень безпеки, адже змінити інформацію в одному блоку без зміни всіх попередніх блоків неможливо.

Ключові аспекти

1. **Децентралізація**: На відміну від баз даних, які управляються однією організацією, блокчейн розподілений серед великої кількості вузлів, що забезпечує вищий рівень прозорості та безпеки.
2. **Незмінність**: Інформація, що була записана в блокчейн, не може бути змінена без зміни всіх наступних блоків, що робить дані дуже стійкими до фальсифікацій.
3. **Консенсус**: Для додавання нового блоку до блокчейна всі учасники (або більшість) мережі повинні погодитися, що транзакція є дійсною. Це досягається за допомогою алгоритмів консенсусу, таких-як Proof of Work або Proof of Stake.

4. **Криптографічне шифрування**: Всі дані в блокчейні захищені за допомогою криптографії, що гарантує безпеку та конфіденційність транзакцій.
5. **Смарт-контракти**: це код, який автоматично виконуються, контролюють або документують юридично значущі події та дії згідно з правилами контракту. Це дозволяє значно знизити потребу в посередниках, таких як юристи або банки, і зменшити час та витрати на ведення угод, також зробивши їх більш безпечними та стабільними.

Переваги

1. **Прозорість та аудитабельність**: Записи в блокчейні доступні для всіх учасників і забезпечують високий рівень прозорості. Кожен блок містить відбиток попереднього блоку, часову мітку і транзакційні дані, що робить всю мережу легкою для аудиту і перевірки, але забезпечуючи конфіденційність завдяки криптографії.
2. **Токенізація**: Блокчейн дозволяє токенізувати активи, перетворюючи їх на цифрові токени, які можуть бути куплені, продані або обмінені в цифровому форматі. Це може включати все від нерухомості до художніх творів, що відкриває нові можливості для інвестицій і комерції.
3. **Ланцюги постачання**: Блокчейн може значно покращити ланцюги постачання, забезпечивши детальну простежуваність товарів від виробника до споживача, що допомагає забезпечити автентичність товарів, боротьбу з підробками, оптимізацію запасів і зменшення втрат.
4. **Фінансові послуги**: Використання блокчейну та смарт-контрактів в фінансовому секторі може змінити спосіб, яким ми виконуємо платежі, здійснюємо міжнародні перекази, доступ до кредитів і багато іншого. Завдяки його прозорості та безпеці, блокчейн може забезпечити швидші та дешевші фінансові послуги, знижуючи потребу в посередниках, таких як банки та платіжні системи.
5. **Цифрова ідентифікація**: Блокчейн може бути використаний для створення надійних і безпечних систем цифрової ідентифікації, що дозволяє користувачам

контролювати свою особисту інформацію і надавати доступ до неї лише за власним бажанням, що особливо важливо для захисту від кіберзагроз та витоку даних.

Смарт-контракти

Визначення

Смарт-контракт — це код, що автоматично виконує угоди в детермінований спосіб. Він зберігається та виконується на блокчейні для забезпечення надійності та безпеки. Смарт-контракти можуть оброблювати кошти, виконуючи транзакції, коли виконуються попередньо визначені умови. Вони працюють на простій логіці if else, що спрощує програмування, одночасно прагнучи усунути людську помилку та ненадійність у традиційних контрактах.

Аналогії та приклади

Загальноприйнята аналогія для розуміння смарт-контрактів — це торговий автомат. Так само, як торговий автомат видає продукт після внесення правильної суми грошей, смарт-контракт виконує вказану дію, як тільки умови виконуються. Наприклад, якщо користувач хоче обміняти криптовалютні токени, смарт-контракт може забезпечити цей обмін без необхідності стороннього посередника, забезпечуючи негайні та бездовірні транзакції.

Переваги

1. **Автоматизація**: Як тільки умови виконуються, транзакції оброблюються автоматично.
2. **Детермінованість**: Результати передбачувані та реалізовані без людського втручання.

3. **Безпека**: Виконуються на блокчейні, що робить їх захищеними від зловживань.
4. **Вартісна ефективність**: Відсутність потреби у посередниках, знижуючи витрати на транзакції.
5. **Швидкість**: Транзакції можуть бути завершені набагато швидше, ніж традиційними методами.
6. **Прозорість**: Умови доступні для всіх зацікавлених сторін.

Сфери застосування

1. **Децентралізовані фінанси (DeFi)**: Смарт-контракти дозволяють реалізувати функціональності, такі як автоматизоване забезпечення ліквідності та децентралізовані стабільні монети.
2. **Управління ланцюгами постачання**: Підвищують прозорість та ефективність у відстеженні товарів та матеріалів.
3. **Crowdfunding**: Кошти можуть випускатися на основі досягнення конкретних, перевірених цілей.

Виклики та ризики

1. **Програмні помилки**: Як і у будь-якому програмному забезпеченні, помилки в коді можуть призвести до значних фінансових втрат, наприклад хак DAO.
2. **Зміни у протоколах**: Оновлення блокчейн-платформи можуть вплинути на поведінку смарт-контрактів.
3. **Інтеграція даних реального світу**: Надійне включення даних реального світу в смарт-контракти вимагає оракулів, що вносить елемент довіри до них.
4. **Регуляторні та правові питання**: Смарт-контракти функціонують у правовій сірій зоні, з розвитком регуляцій, які можуть вплинути на їх впровадження та виконання.

Висновок

Смарт-контракти представляють значний зсув у способі управління та виконання угод в цифрову епоху. З постійним прогресом та зростаючою інтеграцією технології блокчейну, вони налаштовані революціонізувати галузі, роблячи транзакції більш безпечними, ефективними та прозорими. Однак, вирішення поточних обмежень та ризиків є суттєвим для їх ширшого прийняття та ефективності.

DeFi

Огляд

DeFi (децентралізовані фінанси) технологія, що має на меті створити нову, доступну для всіх, фінансову систему, що не вимагає довіри до традиційних посередників, таких як банки. Ця система широко використовує криптографію, технологію блокчейну та смарт-контракти. Смарт-контракти, зокрема, виступають як фундаментальна основа для застосунків DeFi.

Роль Ethereum

Більшість проектів DeFi наразі розробляються на блокчейні Ethereum завдяки його міцній мові програмування, Solidity. Ця мова дозволяє створювати розширені смарт-контракти, які містять усю необхідну логіку для застосунків DeFi. Ethereum також може похвалитися найрозвиненішою екосистемою серед платформ для смарт-контрактів, з тисячами розробників, які щодня створюють нові застосунки, та найбільшою кількістю заблокованих у смарт-контрактах коштів, що додатково зміцнює її довіру до цього підходу.

Історичний контекст

Рух DeFi почався з проектів на кшталт MakerDAO, заснованого у 2015 році. MakerDAO дозволяє користувачам блокувати заставу, таку як ETH, та генерувати DAI, стабільну монету, прив'язану до долара США, яку можна використовувати для заощаджень на платформі Oasis Maker. Ця функціональність відтворює одну з основ фінансової системи — позики та позичання, але виконується у бездозвільний та відкритий спосіб.

Основні компоненти DeFi

1. **Позики та позичання**: Платформи на кшталт Compound та Aave дозволяють користувачам інвестувати свої активи та заробляти відсотки або позичати проти застави.
2. **Stable coins**: Окрім алгоритмічних stable coins (DAI), існують централізовані stable coins (USDT та USDC), які, незважаючи на свою популярність, критикують за їхній централізований підхід.
3. **Децентралізовані біржі (DEX)**: На відміну від традиційних бірж, децентралізовані біржі, такі як Uniswap та Kyber, дозволяють обмін активів без посередника в децентралізованій системі.
4. **Деривативи та маржинальна торгівля**: Платформи на кшталт Synthetix та dYdX дозволяють користувачам займатися торгівлею деривативами та використовувати плече, подібно до традиційних ринків.
5. **Страхування**: Проекти, такі як Nexus Mutual, пропонують покриття від збоїв смарт-контрактів та інших ризиків у просторі DeFi.

Децентралізовані vs Традиційні фінанси

1. **Бездозвільний доступ**: DeFi не вимагає процедур Know Your Client, на відміну від традиційних фінансів, які регулюються вимогами нагляду.

2. **Відкрите джерело та співпраця**: Інфраструктура DeFi сприяє прозорості та спільному розвитку, в той час як C-Fi часто включає рішення, що приймаються за закритими дверима.
3. **Стійкість до цензури та нижчі витрати**: DeFi пропонує стійкість до цензури та, загалом, нижчі витрати, усуваючи посередників.

Ризики та виклики

Незважаючи на інноваційний підхід, DeFi не позбавлений ризиків:

1. **Уразливості смарт-контрактів**: Помилки у коді смарт-контрактів можуть призвести до значних втрат у коштах.
2. **Зміни в протоколах**: Оновлення основних платформ, таких як Ethereum, можуть вплинути на існуючі застосунки DeFi.
3. **Системні ризики**: Різке зниження цін може спровокувати ланцюгові ліквідації, підкреслюючи взаємозалежну природу протоколів DeFi.

Висновок

DeFi швидко розвивається, пропонуючи як високі ризики, так і високі винагороди. З дозріванням цього сектора він обіцяє викликати та потенційно трансформувати традиційний фінансовий ландшафт.

Ліквідні пули

Вступ до ліквідних пулів

Ліквідні пули є важливими компонентами децентралізованих бірж DEX, які допомагають забезпечувати торгівлю, надаючи ліквідність, що закріплена в

смарт-контракті. Ці пули складаються з токенів, які дозволяють здійснювати миттєву торгівлю без потреби в традиційних marketmaker-ів.

Навіщо потрібні ліквідні пули?

На традиційних біржах marketmaker-и відіграють ключову роль, забезпечуючи ліквідність, таким чином дозволяючи користувачам виконувати торги без значних затримок. Однак, відтворення цієї моделі в децентралізованих фінансах на платформах, які мають відносно низьку пропускну спроможність транзакцій та високі вартості газу (вимірювання обчислювальної роботи, яку потрібно виконати для здійснення транзакцій або виконання смарт-контрактів), було б неефективним та дорогим.

Перехід до ліквідних пулів

Враховуючи неефективність моделі order-book на блокчейн-платформах, ліквідні пули були розроблені як інноваційне рішення в DeFi. Ці пули спрощують механіку торгівлі, усуваючи потребу в прямому відповіданні покупців і продавців, знижуючи залежність від зовнішніх marketmaker-ів та усуваючи проблеми, пов'язані з обсягом торгів та ліквідністю.

Як працюють ліквідні пули

Типовий ліквідний пул у DeFi містить токен та stable coin, що створює ринок для цієї пари токенів. Перший постачальник ліквідності до пулу встановлює початкове співвідношення кількості токенів відносно кількості stable coins, формуючи ліквідність токену. При цьому добуток кількості stable coins та токенів має бути завжди сталим.

Комісія та стимули

Торги, виконані в межах ліквідного пулу, передбачають комісія (наприклад, 0.3% на Uniswap), який розподіляється між усіма постачальниками ліквідності пропорційно до їхньої частки в пулі. Постачальники заробляють комісію з торгів та можуть забрати свою ліквідність, продавши токени, які вони інвестували, на актуальну ціну.

Різниця між протоколами

Різні DEX використовують різні Automated Market Makers:

1. **Uniswap**: Використовує модель постійного продукту маркетмейкера, яка забезпечує ліквідність незалежно від розміру торгів.
2. **Balancer**: Дозволяє пулам містити до восьми різних токенів, забезпечуючи гнучкість у способах надання та ціноутворення ліквідності.
3. **Curve**: Спеціалізується на пулах для активів, які теоретично повинні мати схожі ціни, таких як стабільні монети, пропонуючи нижчу ковзання та комісії.

Переваги ліквідних пулів

Ліквідні пули дозволяють безперешкодно торгувати без необхідності традиційних marketmaker-ів, забезпечують стабільну ліквідність токену та покращують загальний досвід торгівлі. Вони також дозволяють майнінг ліквідності, де постачальники заробляють додаткові винагороди, стимулюючи участь у покращенні ліквідності.

Виклики та ризики

Хоч ліквідні пули і мають багато переваг, вони також мають ризики:

1. **Нестійка втрата**: Виникає, коли співвідношення цін токенів у пулі змінюється після того, як постачальник продає суттєву кількість своїх токенів, що потенційно може призводити до обвалу ціни. Для того, щоб така ситуація не виникала, токени інвесторів обмежують параметрами cliff, TGE Unlock та vesting, забезпечуючи поступове розблокування токенів для продажу.

2. **Хаки ліквідних пулів**: Уразливості смарт-контрактів можуть призвести до втрати коштів через зломи.

Висновок

Ліквідні пули є фундаментальними для функціонування DEX у DeFi, пропонуючи децентралізований механізм для забезпечення торгівлі без традиційних обмежень order-book-ів та marketmaker-ів. Ці пули адаптуються до потреб простору DeFi, забезпечуючи гнучкість, зниження витрат та збільшення можливостей участі для користувачів.

Cliff, TGE Unlock, Vesting

1. **Cliff:** період часу, протягом якого певна частина токенів залишається заблокованою і не може бути доступною для використання чи торгівлі. Зазвичай цей період встановлюється для захисту інтересів проекту та інвесторів, щоб запобігти раптовому викиду великої кількості токенів на ринок, обвалюючи ціну.
2. **TGE Unlock (Token Generation Event Unlock):** яка частина токенів стають доступними для використання чи торгівлі після періоду cliff. Це перша можливість для інвесторів та інших учасників проекту отримати доступ до своїх токенів.
3. **Vesting:** період часу після cliff, під час якого щомісяця поступово розблоковується токени, що лишились з нерозблокованих після TGE Unlock, дозволяючи уникнути надмірного тиску на ринок і забезпечуючи стабільності ціни токену.

Реалізована та нереалізована дохідності

Метрики для відслідковування ефективності інвестування у нашу DEX:

1. **Реалізована дохідність:** визначаються прибутком від усіх проданих токенів до цього часу.
2. **Нереалізована дохідність:** нереалізований прибуток інвестора, що сумою реалізованої дохідності та добутку кількості ще не проданих токенів (включаючи заблоковані) на теперішню ціну.

Опис системи

Децентралізована біржа з використанням власного токена для сплати комісій

Вступ

Система децентралізованої біржі DEX дозволяє користувачам обмінювати різні криптовалюти без потреби в центральному посереднику.

У цій системі використовується власний токен для сплати комісій, ціна якого формується пулом ліквідності. Додатково, для токена передбачені спеціальні умови для інвесторів, які включають механізми cliff, TGE Unlock та vesting. Система також підтримує маркетингову діяльність та команду розробників.

Основні компоненти системи

1. **Пул ліквідності**: Створює ринок для токена системи та інших криптовалют.
2. **Токен**: Використовується для сплати комісій на біржі.
3. **Інвестори**: Початково скуповують токени по знижці, але мають обмеження на продаж.
4. **Маркетинг**: Залучення користувачів та підвищення популярності біржі.
5. **Команда розробників**: Підтримка та розвиток платформи.

Пул ліквідності

Пул ліквідності є ключовим елементом децентралізованої біржі. Він дозволяє користувачам обмінювати різні криптовалюти, забезпечуючи ліквідність для торгівлі. У цій системі пул ліквідності формується з пари токенів: нашого токена та stable coin.

1. **Створення пулу ліквідності**: Користувачі додають токени та stable coins у пул, забезпечуючи ліквідність та формуючи ціну. За це вони отримують токени ліквідності (LP-токени), які представляють їх частку в пулі.
2. **Формування ціни**: Ціна нашого токена визначається на основі співвідношення токену до stable coins у пулі за допомогою автоматизованого маркетмейкера.
3. **Комісія за торгівлю**: Кожна транзакція в пулі передбачає сплату комісії у вигляді нашого токена, що розподіляються між постачальниками ліквідності.
4. **Знижки на комісії**: Для стимулювання використання токена системи користувачі можуть отримувати знижки на комісії при сплаті саме нашим токеном.

Інвестори

Біржа залучає інвесторів, які скуповують токени на ранніх етапах по знижці. Щоб запобігти раптовому продажу великої кількості токенів і зберегти стабільність ціни, впроваджуються механізми cliff, TGE Unlock та vesting.

Маркетинг

Для успішного розвитку біржі необхідна активна маркетингова діяльність. Вона включає залучення нових користувачів, підвищення популярності проекту та створення довіри до платформи.

1. **Залучення користувачів**: Проведення рекламних кампаній, участь у конференціях, співпраця з лідерами думок.
2. **Винагороди та акції**: Пропозиції бонусів для нових користувачів, програми лояльності для постійних клієнтів.
3. **Інформування**: Публікації в соціальних мережах, блоги, новини про оновлення та події.

Команда розробників

Команда розробників забезпечує підтримку та розвиток платформи, вирішує технічні проблеми та впроваджує нові функції.

1. **Розробка платформи**: Створення та підтримка інфраструктури біржі, розробка смарт-контрактів.
2. **Безпека**: Проведення аудитів безпеки, моніторинг мережі, захист від зломів.
3. **Покращення функціоналу**: Додавання нових функцій, вдосконалення існуючих механізмів, інтеграція з іншими платформами.

Що маркетинг, що команда розробників теж отримують токени, як винагороду за участь у проекті, та мають обмеження на продаж аналогічні інвесторам, хоча умови cliff, TGE Unlock та vesting у них можуть бути інші.

Висновок

Система децентралізованої біржі з використанням власного токена для сплати комісій поєднує в собі механізми ліквідності, підтримку інвесторів та активну маркетингову діяльність. Використання cliff, TGE Unlock та vesting забезпечує стабільність ринку і довгострокову стабільність ціни токена. Професійна команда розробників та ефективні маркетингові стратегії сприяють зростанню та розвитку проекту, залучаючи нових користувачів та інвесторів.

Проте для дослідження динаміки зміни ціни токену не обов’язково реалізовувати цю систему повністю, а отже можемо спростити.

Опис спрощеної системи децентралізованої біржі

Вступ

У нашій спрощеній системі децентралізованої біржі користувачі напряму купують наш токен для сплати комісій, а інвестори та інші учасники системи мають аналогічні обмеження щодо розблокування токенів, хоч і можуть мати різні умови цих обмежень. Ціна нашого токена формується пулом ліквідності, як це зазвичай робиться на DEX.

Основні компоненти системи

1. **Пул ліквідності**: Визначає ціну токена на основі співвідношення токенів.
2. **Користувачі**: Купують токени для сплати комісій.
3. **Інвестори**: Початково мають певну кількість токенів, які збираються реалізувати, але мають обмеження на це.
4. **Інші учасники**: Теж зацікавлені в реалізації токенів, які вони мають спочатку, та мають аналогічні обмеження щодо розблокування токенів, аналогічні інвесторам.

Пул ліквідності

Пул ліквідності формується для нашого токена та stable coins і є ключовим елементом для формування ціни токена. Для чистоти дослідження ми будемо моделювати однакову початкову ціну для всіх сценаріїв та випадків.

Користувачі

Користувачі напряму купують наш токен для сплати комісій на біржі. Це створює постійний попит на токен, що позитивно впливає на його ціну. Додатково реалізуємо поступове збільшення попиту до певного місяця, моделюючи звичайну поведінку попиту в справжніх системах.

1. **Попит на токен**: Попит на токен забезпечує його стабільне використання та зростання ціни.

Інвестори

Інвестори початково мають певну кількість токенів, які збираються реалізувати, і мають обмеження щодо продажу токенів.

1. **Cliff-період**: Протягом встановленого часу інвестори не можуть продавати свої токени.
2. **TGE Unlock**: Після закінчення cliff-періоду певна частина токенів стає доступною для використання.
3. **Vesting-період**: Решта токенів поступово розблоковуються протягом встановленого періоду (наприклад, 2 роки), що дозволяє зберігати стабільність ринку.
4. **Пропозиція на продаж**: Інвестори можуть продавати певний відсоток від розблокованих токенів, створюючи пропозицію на ринку.

Інші учасники

Інші учасники системи, у яких ми об’єднали для зручності маркетингову команду, розробників, тощо, мають аналогічні обмеження щодо розблокування токенів, як і інвестори.

1. **Cliff-період**, **TGE Unlock**, **Vesting-період**.
2. **Пропозиція на продаж**: Інші учасники можуть продавати певний відсоток від розблокованих токенів, створюючи додаткову пропозицію на ринку.

План розробки та використання додатку

1. Ініціалізація та налаштування

1.1. Ініціалізація проекту

- Створити новий Python проект.
- Налаштувати віртуальне середовище для проекту.
- Встановити необхідні бібліотеки, такі як numpy для математичних обчислень, matplotlib для візуалізації даних та pandas.

2. Створення основних класів

2.1. Клас LiquidityPool в liquidity_pool.py

Атрибути:

- stable_coins: кількість stable coins у пулі ліквідності. Цей атрибут змінюється, коли користувачі купують або продають токени. Він є основним фактором, який впливає на ціну токена.
- luquidity_supply: загальна кількість токенів у пулі, виділена компанією (наприклад, 10% від 100 мільйонів).
- current_price: поточна ціна токена. Вона буде змінюватись залежно від кількості стейбл коінів у пулі ліквідності та попиту користувачів.
- multiply_constant: константа перемноження luquidity_supply та stable_coins.

Методи:

- update_pool(amount): метод для зміни кількості стейбл коінів у пулі та для зміни токенів у пулі.
- update_token_price(): метод для оновлення ціни токена на основі поточного стану пулу ліквідності та попиту користувачів.

- get_pool_state(): метод для повернення інформації стосовно пулу ліквідності.

2.2. Клас Investor в investor.py

Атрибути:

- cliff: період, протягом якого інвестори не можуть продавати свої токени.
- tge_unlock: відсоток токенів, який розблоковується під час генерації токенів. Це значення вказує, скільки токенів інвестор може продати відразу після запуску.
- vesting: період поступового розблокування токенів, вимірюється у місяцях або днях. Визначає, як часто і скільки токенів буде розблоковуватись для інвестора протягом часу.
- investors_supply: виділена компанією кількість токенів (наприклад, 15% від 100 мільйонів).
- token_holdings: кількість токенів, якими володіє інвестор. Це значення змінюється залежно від атрибутів зверху і часу.
- monthly_sell_percentage: відсоток токенів від token_holdings, які продають інвестори щомісяця.
- month_passed: лічильник місяців, що пройшли.
- monthly_get_after_cliff: кількість токенів, які розблоковуються щомісяця після cliff періоду.
- tokens_sold: кількість вже проданих токенів.
- realized_profit: реалізований прибуток.
- unrealized_profit: нереалізований прибуток.

Методи:

- update_token_holdings(amount): метод для зміни token_holdings залежно від параметрів.

- sell_tokens(liquidity_pool): метод для продажу токенів щомісяця враховуючи обмеження інвесторів.

2.3. Клас Others в others.py

Атрибути:

- cliff: період, протягом якого учасники не можуть продавати свої токени.
- tge_unlock: відсоток токенів, який розблоковується під час генерації токенів. Це значення вказує, скільки токенів учасник може продати відразу після запуску.
- vesting: період поступового розблокування токенів, вимірюється у місяцях або днях. Визначає, як часто і скільки токенів буде розблоковуватись для учасника протягом часу.
- investors_supply: загальна кількість токенів у учасника, виділена компанією (наприклад, 15% від 100 мільйонів).
- token_holdings: кількість токенів, якими володіє учасник. Це значення змінюється залежно від атрибутів зверху і часу.
- monthly_sell_percentage: відсоток токенів від token_holdings, які продають учасники щомісяця.
- month_passed: лічильник місяців, що пройшли.
- monthly_get_after_cliff: кількість токенів, які розблоковуються щомісяця після cliff періоду.

Методи:

- update_token_holdings(amount): метод для зміни token_holdings залежно від параметрів.
- sell_tokens(liquidity_pool): метод для продажу токенів щомісяця враховуючи обмеження учасників.

2.4. Клас User в user.py

Атрибути:

- demand: попит на токени з боку користувачів. Попит виражений у кількості токенів, які користувачі хочуть придбати за певний період часу.
- token_holdings: кількість токенів, якими володіє користувач. Це значення змінюється при купівлі або продажу токенів.
- demand_growth_rate: темп зростання попиту щомісяця.

Методи:

- update_demand(month): метод для зміни поточного попиту користувачів на токени залежно від темпу зростання попиту.
- buy_tokens(amount): метод для купівлі токенів користувачем. Зменшує кількість стейбл коінів у пулі ліквідності.

2.5. main.py

Змінні:

- others: об'єкт класу Others, що представляє інших учасників системи.
- liquidity_pool: об'єкт класу LiquidityPool, що представляє пул ліквідності.
- investors: список об'єктів класу Investor, що представляє всіх інвесторів.
- users: об'єкт класу User, що представляє всіх користувачів.

Методи:

- main(): Основний метод для моделювання ринкових умов і дослідження впливу різних факторів на ціну токена.
  - Створює папки для збереження графіків.
  - Реалізує перший сценарій:
    - Ініціалізує різні початкові значення для stable_coins.
    - Для кожного значення stable_coins ініціалізує об'єкти LiquidityPool, Investor, User та Others.

        - Імітує щомісячні зміни для кожного початкового значення stable_coins.
        - Зберігає графіки цін токену та інші показники.
    - Реалізує другий сценарій:
        - Ініціалізує різні значення для demand.
        - Для кожного значення demand ініціалізує об'єкти LiquidityPool, Investor, User та Others.
        - Імітує щомісячні зміни для кожного значення demand.
        - Зберігає графіки цін токену та інші показники.
    - Реалізує третій сценарій:
        - Для кожного значення stable_coins та demand ініціалізує об'єкти LiquidityPool, Investor, User та Others.
        - Імітує щомісячні зміни для кожної пари.
        - Зберігає графіки цін токену та інші показники.

- create_investors(): Метод для створення списку інвесторів з різними параметрами.
    - Повертає список об'єктів класу Investor.

- simulate_monthly_changes(months, liquidity_pool, investors, user, others): Метод для імітації щомісячних змін у пулі ліквідності, попиті користувачів, продажах інвесторів та інших учасників протягом 36 місяців.
    - Імітує купівлю токенів користувачами.
    - Оновлює попит користувачів щомісяця.
    - Імітує продаж токенів інвесторами та іншими учасниками.
    - Повертає дані про стан пулу ліквідності та прибутки інвесторів.

- plot_token_price_over_time(all_data, graphic_file_name, graphic_name, x_label, group_label, folder): Метод для побудови графіків зміни ціни токену з часом.
    - Будує графік зміни ціни токену для кожного початкового значення stable_coins або demand.
    - Зберігає графік у відповідну папку.

- plot_combined_graph(all_data, graphic_file_name, graphic_name, x_label, value_label, group_label, folder): Метод для побудови графіків змін кількості токенів або stable_coins.
  - Будує графік змін кількості токенів або stable_coins у пулі ліквідності для кожного початкового значення stable_coins або demand.
  - Зберігає графік у відповідну папку.

- plot_investor_profits(all_data, graphic_file_name, graphic_name, x_label, investor_type, filter_value, filter_type, folder): Метод для побудови графіків прибутку інвесторів.
  - Будує графік реалізованого та нереалізованого прибутку інвесторів для кожного початкового значення stable_coins або demand.
  - Зберігає графік у відповідну папку.

3. Моделювання системи

3.1. Налаштування початкових умов

Визначити початкові значення для кількості токенів, стейбл коінів у пулі ліквідності, параметрів інвесторів, відсотку комісії та попиту користувачів. Дослідити два сценарії: різна початкова кількість stable coins та різні початкові demand користувачів.

Перший сценарій

1. Загальна кількість токенів - 100 000 000.
2. Моделюємо 5 різних інвесторів з різною пропозицією на продаж токенів: 0.05, 0.1, 0.2, 0.3, 0.4. Параметри cliff = 6, tge_unlock = 10%, vesting = 18 та по 4 000 000 токенів для кожного інвестора.
3. Параметри cliff = 6, tge_unlock = 10%, vesting = 18 та різниця 80 000 000 та початкового liquidity_supply токенів для учасників (оскільки з різною початковою кількістю stable coins, має відрізнятись і початкове liquidity_supply для того, щоб початкова ціна для усіх випадків була однаковою).

4. Пропозиція учасників складає 10%.
5. Попит користувачів складає 50 000 (у доларах) та зростання на 30% перші 12 місяців.
6. Кількість stable coins у пулі ліквідності реалізує різні випадки для цього сценарію дослідження і складає такий масив: 1 500 000, 2 000 000, 2 500 000, 3 500 000, 4 500 000. Для однакової початкової ціни 0.1 для кожного випадку, кількість токенів виділених на пул ліквідності має складати відповідно: 15 000 000, 20 000 000, 25 000 000, 35 000 000, 45 000 000.

Другий сценарій

1. Загальна кількість токенів - 100 000 000.
2. Моделюємо 5 різних інвесторів з різною пропозицією на продаж токенів: 0.05, 0.1, 0.2, 0.3, 0.4. Параметри cliff = 6, tge_unlock = 10%, vesting = 18 та по 4 000 000 токенів для кожного інвестора.
3. Параметри cliff = 6, tge_unlock = 10%, vesting = 18 та 65 000 000 токенів для учасників.
4. Пропозиція учасників складає 10%.
5. Попит користувачів, зі зростанням у 30% перші 12 місяців, реалізує різні випадки для цього сценарію дослідження і складає такий масив: 10 000, 30 000, 50 000, 70 000, 100 000 (у доларах).
6. Пул ліквідності складає 15% від кількості токенів (15 000 000), з кількістю стейбл коінів 1 500 000 (початкова ціна 0.1).

Третій сценарій

1. Загальна кількість токенів - 100 000 000.

2. Моделюємо 5 різних інвесторів з різною пропозицією на продаж токенів: 0.05, 0.1, 0.2, 0.3, 0.4. Параметри cliff = 6, tge_unlock = 10%, vesting = 18 та по 4 000 000 токенів для кожного інвестора.
3. Параметри cliff = 6, tge_unlock = 10%, vesting = 18 та різниця 80 000 000 та початкового liquidity_supply токенів для учасників (оскільки з різною початковою кількістю stable coins, має відрізнятись і початкове liquidity_supply для того, щоб початкова ціна для усіх випадків була однаковою).
4. Пропозиція учасників складає 10%.
5. Попит користувачів, зі зростанням у 30% перші 12 місяців, реалізує різні випадки для цього сценарію дослідження і складає такий масив: 10 000, 30 000, 50 000, 70 000, 100 000 (у доларах).
6. Кількість stable coins у пулі ліквідності також реалізує різні випадки для цього сценарію дослідження і складає такий масив: 1 500 000, 2 000 000, 2 500 000, 3 500 000, 4 500 000. Для однакової початкової ціни 0.1 для кожного випадку, кількість токенів виділених на пул ліквідності має складати відповідно: 15 000 000, 20 000 000, 25 000 000, 35 000 000, 45 000 000.

3.2. Графіки

Як результат моделювання системи отримуємо такі графіки, для перших двох сценаріїв:

- Порівняння зміни ціни токену від часу кожної ітерації в одному графіку.
- Для кожного випадку по графіку для порівняння зміни реалізованої дохідності від часу для кожного інвестора.
- Для кожного випадку по графіку для порівняння зміни нереалізованої дохідності від часу для кожного інвестора.
- Порівняння зміни кількості токенів пулу ліквідності кожної ітерації від часу.

- Порівняння зміни кількості stable coins пулу ліквідності кожної ітерації від часу.

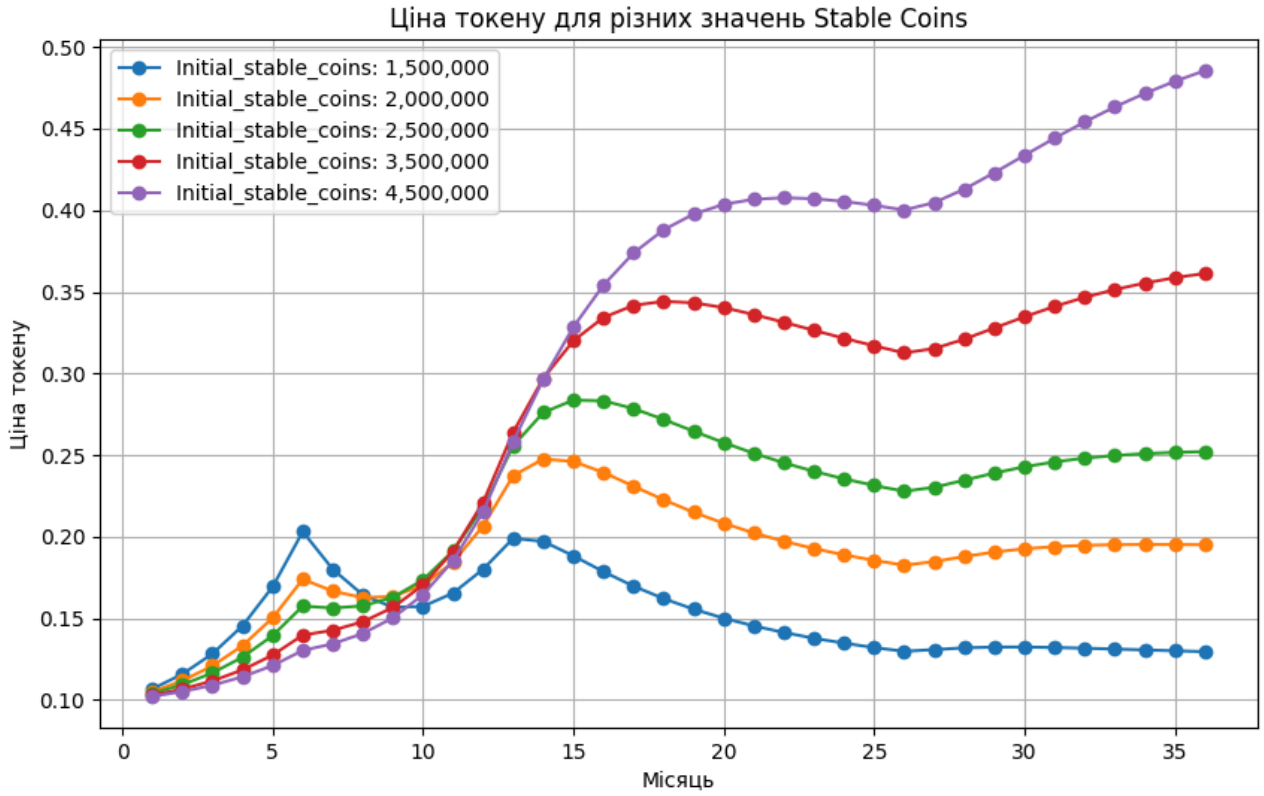
Як результат для третього сценарію отримуємо:

- Для кожного випадку початкової розмірності пулу ліквідності по графіку для порівняння зміни ціни токену від часу відносно різних demand.
- Для кожного випадку початкової розмірності пулу по графіку для порівняння різних випадків для зміни кількості токенів у пулі ліквідності від часу відносно різних demand.
- Для кожного випадку початкової розмірності пулу по графіку для порівняння різних випадків для зміни кількості stable coins у пулі ліквідності від часу відносно різних demand.

Аналіз результатів

Аналіз першого сценарію

Спочатку розглянемо графік ціни токену від часу для різних значень розміру пулу ліквідності:



Можемо поділити графік на три частини візуально, а саме на такі періоди часу:

1. **1-6 місяці:** період cliff, коли усі токени інвесторів заблоковані, а отже вони не можуть продавати їх і впливати на ціну.
2. **7-25 місяці:** період vesting, який наступає відразу після cliff, за цей період розблоковується кожен місяць по однаковій частині токенів від різниці початкових токенів та TGE Unlock.
3. **26-36 місяці:** період, коли інвестори вже не отримують токени кожен місяць, проте продовжують продавати ті токени, які у них лишились.

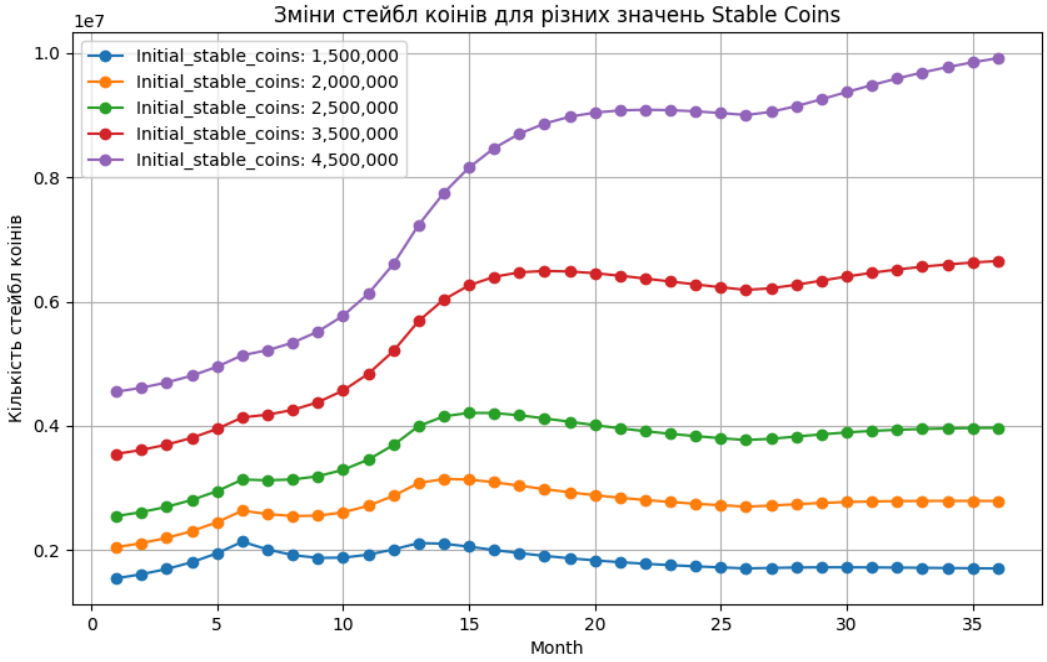
Також можемо помітити, що у перші 6 місяців ціна росте нелінійно. Тут відслідковується поступовий ріст попиту на 30% щомісяця, моделюючи ріст популярності нашої біржі на ринку. У середньому популярність платформи зростає приблизно рік, а потім тримається одного значення, тому поступовий ріст попиту триває саме 12 місяців, а потім тримається сталого значення усі подальші місяці.

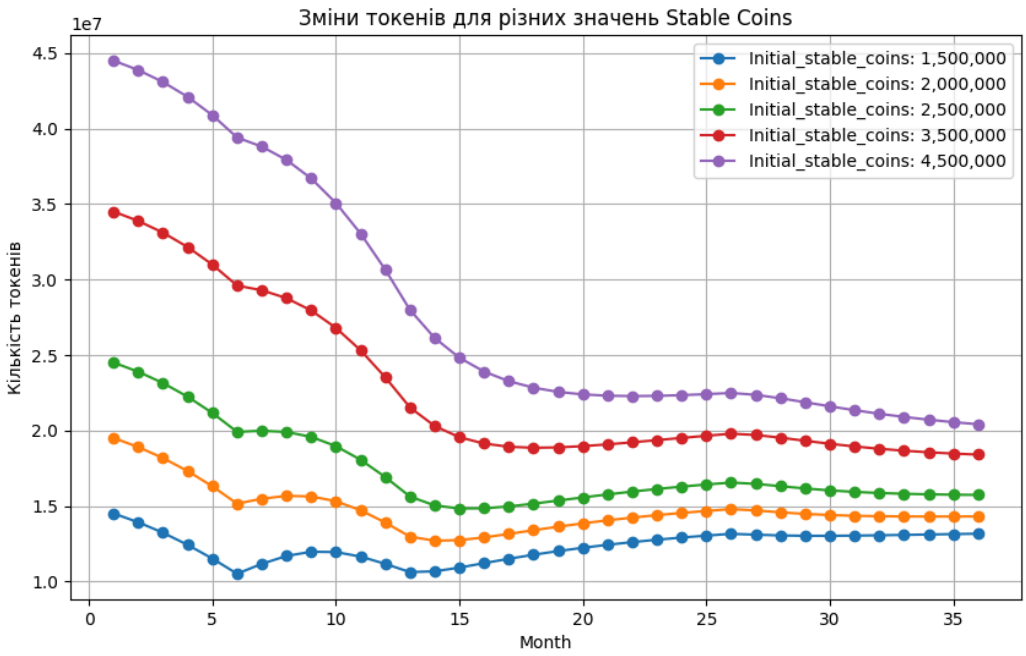
Враховуючи фактори перелічені вище, ми бачимо, що у підпроміжку 6-12 місяця, спочатку, пропозиція інвесторів та інших учасників перевищила попит користувачів. Проте пізніше покупка токенів почала перевищувати, поки не зупинила свій ріст попит 12-го місяця.

Важливо також пам’ятати, що хоч пропозиція інвесторів відсотково стала, але через різну кількість доступних токенів, що розблоковуються щомісяця, продається кожного разу різна кількість.

Загалом, порівнюючи різні початкові розміри пулу ліквідності, можемо зробити висновок, що чим менше розмір пулу, тим менша стійкість до змін.

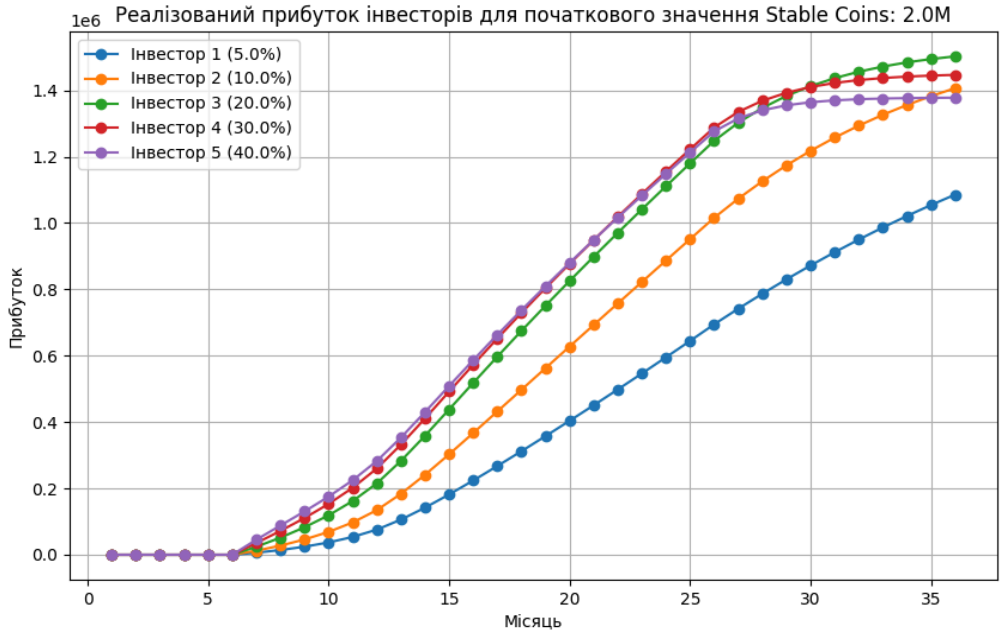
Розглянемо графіки кількості токенів та стейбл коінів:

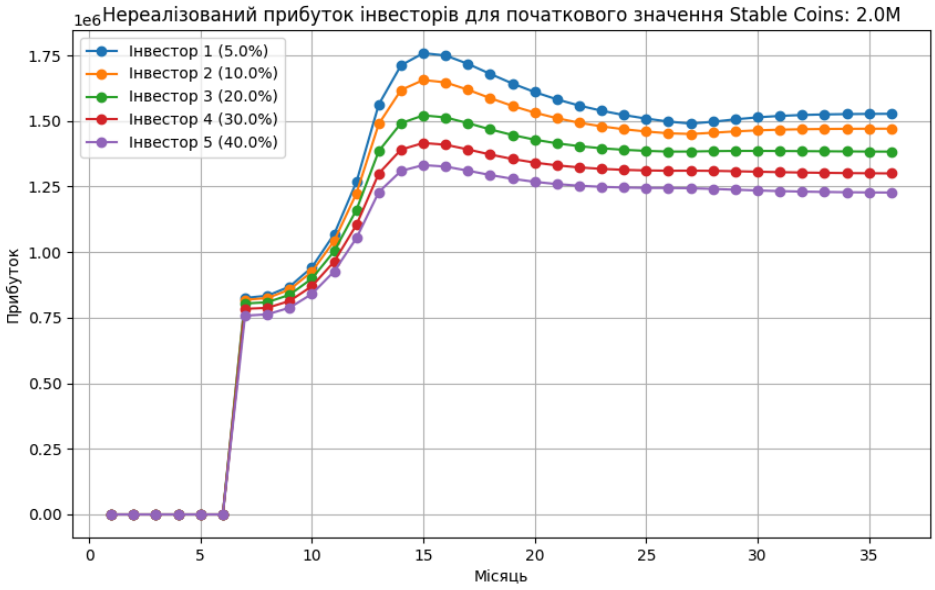




Бачимо, що коли один зростає, то інший завжди спадає, це пов`язано з їх гіперболічною залежністю один від одного, оскільки їх добуток завжди має бути константою.

Наступний графік, який варто порівняти, це графік реалізованої та нереалізованої дохідності інвесторів для конкретного випадку (наприклад для 2 мільйонів початкових стейбл коінів):



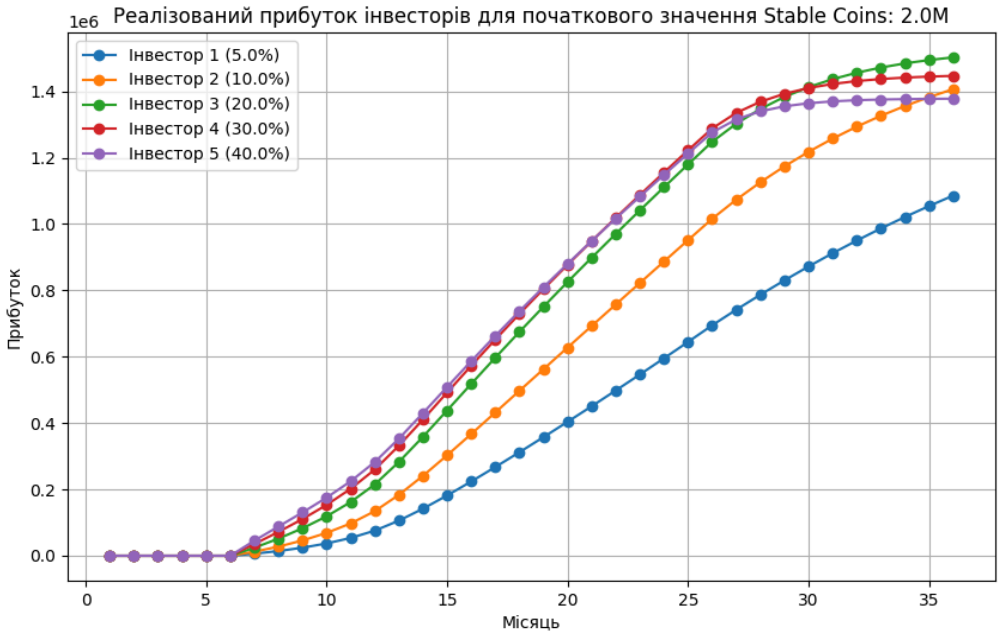


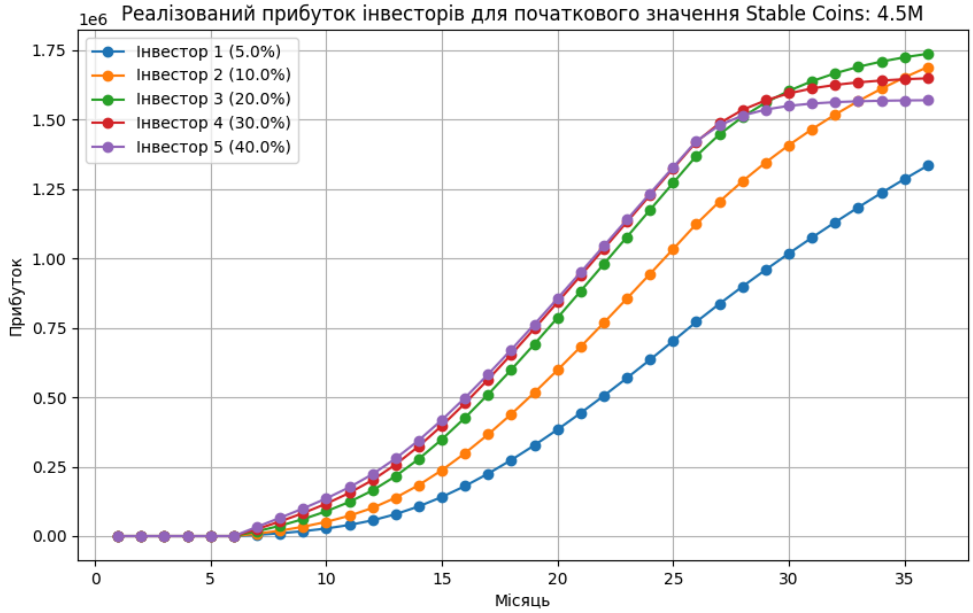
У даних графіках ми можемо побачити різницю дохідності інвесторів, які мають різні пропозиції: 5%, 10%, 20%, 30%, 40%.

Графік реалізованої дохідності показує, який дохід має кожен інвестор у певний період часу. Спостерігаємо, що жадібний інвестор (з пропозицію 40%) не отримує найбільший прибуток та займає передостаннє місце наприкінці 36-го місяця. Лідирує третій інвестор, який має пропозицію 20%, що обігнав інших у 30-ому місяці.

Аналізуючи графік нереалізованої дохідності бачимо, що для всіх інвесторів було найбільш вигідно продати усі свої токени саме 15-го місяця. Уцілому найменш жадібний інвестор має найбільший нереалізований дохід.

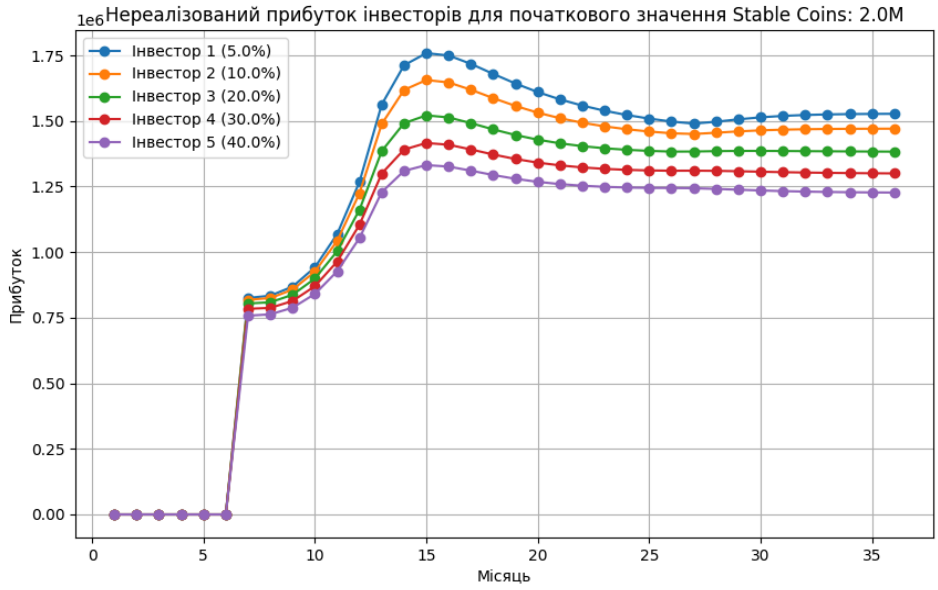
Тепер порівняємо графіки реалізованої дохідності для різних розмірів пулів ліквідності (2*10^6 та 4.5*10^6):

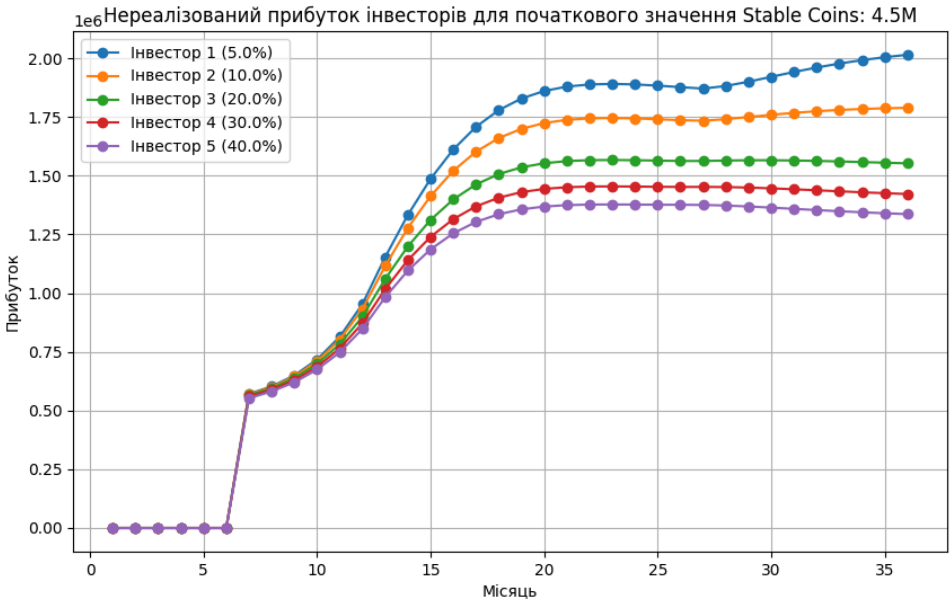




Бачимо різницю у тому, що другий інвестор, який у графіку для 2 мільйонів stable coins займав третє місце, у 4.5 займає друге. Також, інвестор 1 вже ближче до своїх колег по графіку. Тобто чим більший пул ліквідності, тим потужніша дохідність менш жадібних інвесторів.

Останнє порівняння для даного сценарію буде аналогічне минулому, але для нереалізованих дохідностей:

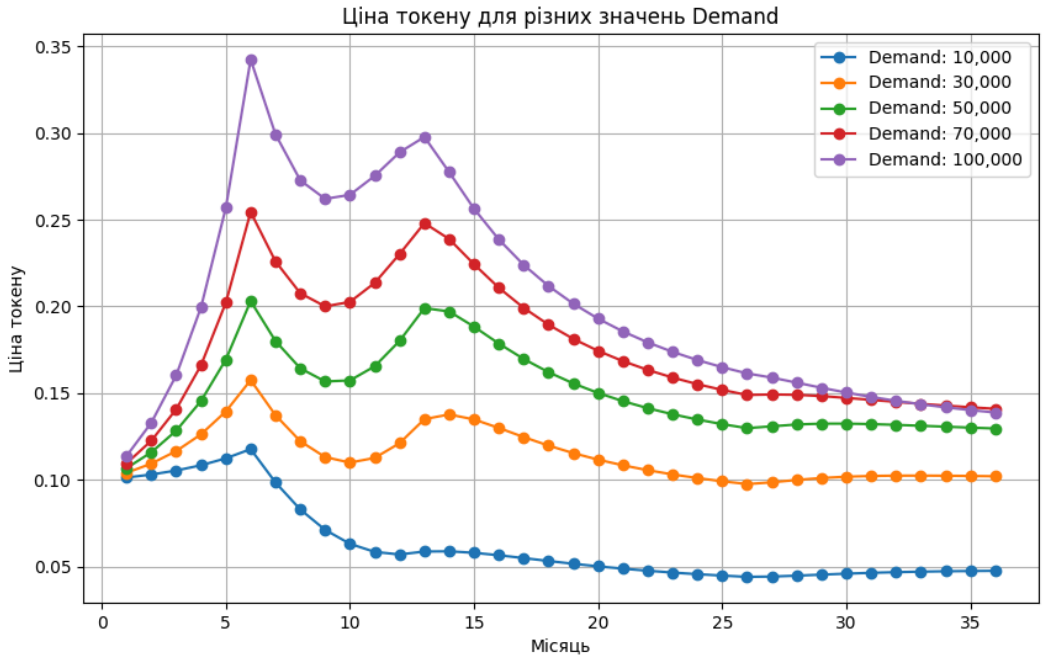




Порівнюючи дані графіки, бачимо, що на відміну від 2 мільйонів stable coins, де усі інвестори мають найбільший нереалізований дохід на 15-ий місяць, у випадку 4.5 мільйонів усе відрізняється. Перші два інвестори мають найбільший можливий прибуток 36-го місяця, коли інші - десь на 23-25 місяць. Також, чим більший пул, тим більша різниця між нереалізованими доходами.

Аналіз другого сценарію

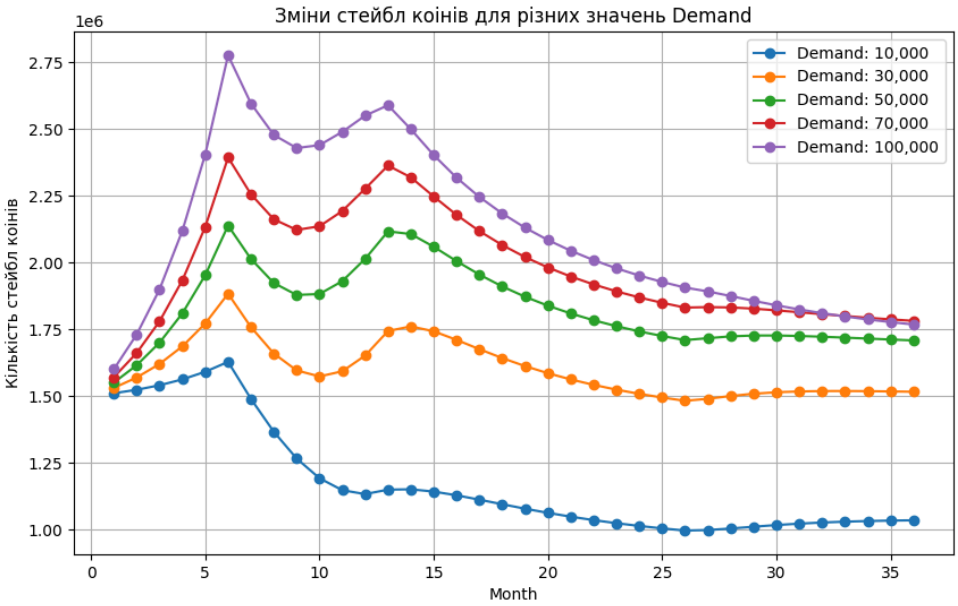
Спочатку розглянемо графік ціни токену від часу для різних значень попиту користувачів:

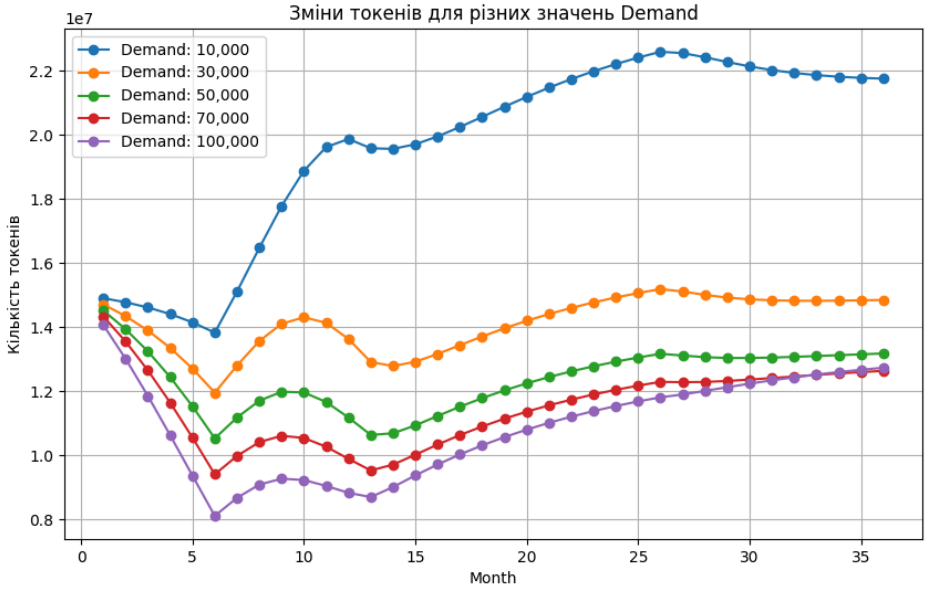


Аналізуючи графік, можемо його знову поділити на 3 періоди: 0-6, 7-25, 26-36, як і в минулому сценарії, з тих самих причин. Також можемо аналогічно знайти підперіод 6-12, де спочатку пропозиція перевищує, але після попит набирає сили до 12-го місяця. Але даний сценарій дає змогу дослідити цей період більш детально, показуючи, що для початкового попиту 10 000 перевага була не такою великою, як у випадку з початковим 70 000.

З цікавого, бачимо, що ціна для меншого початкового попиту 70 000 з 33 місяця перевищує ціну більшого початкового попиту 100 000, та і в цілому бачимо, що графіки поступово зближуються під кінець.

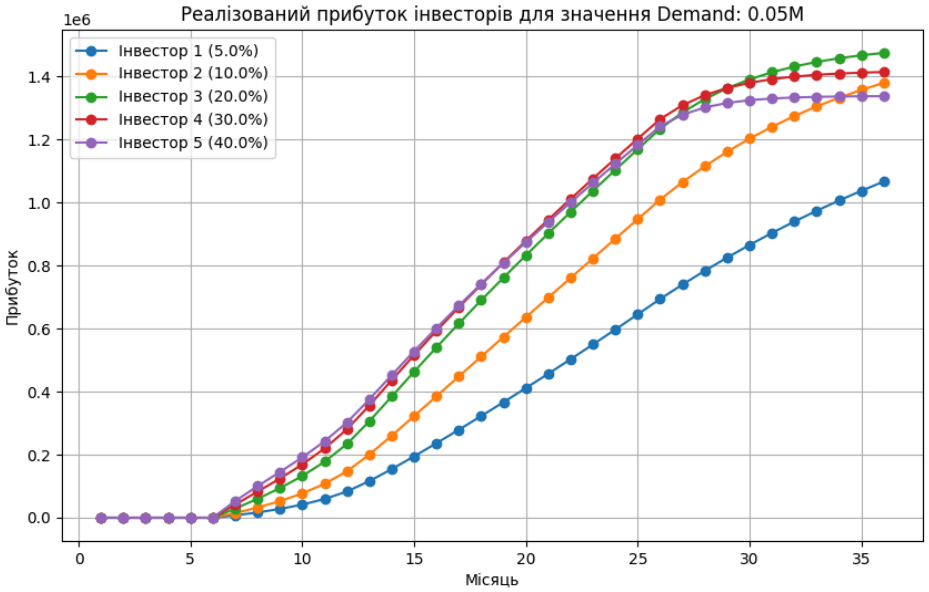
Розглянемо кількість токенів та stable coins:

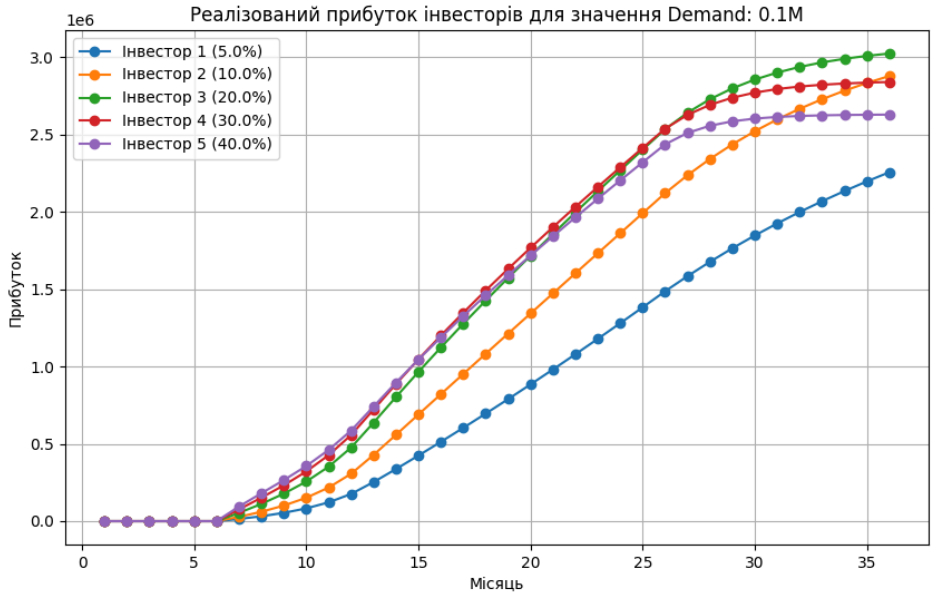




Бачимо аналогічний першому сценарію гіперболічний зв’язок. Також бачимо однаковий перетин у 33-ому місяці, що і в графіку ціни токену від часу.

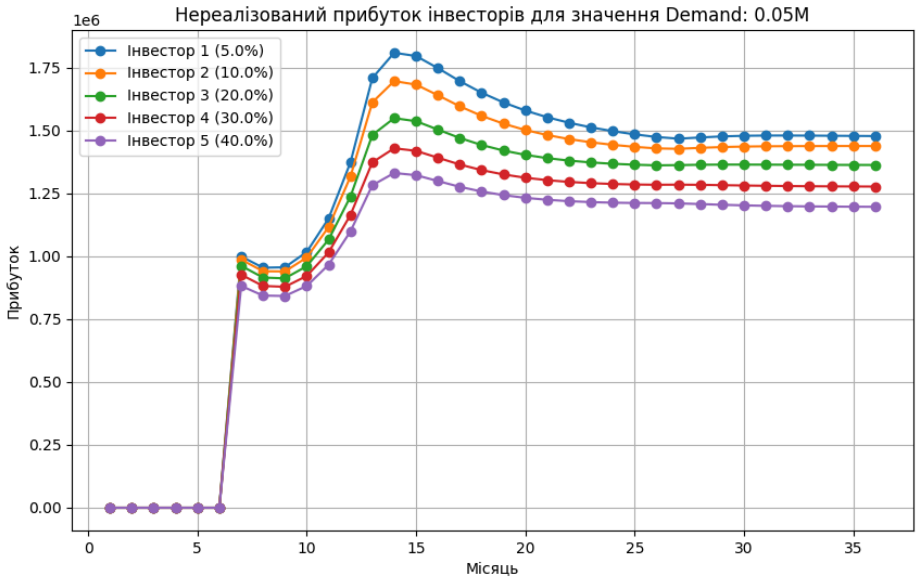
Розглянемо та відразу і порівняємо дохідності для різного початкового попиту користувачів:

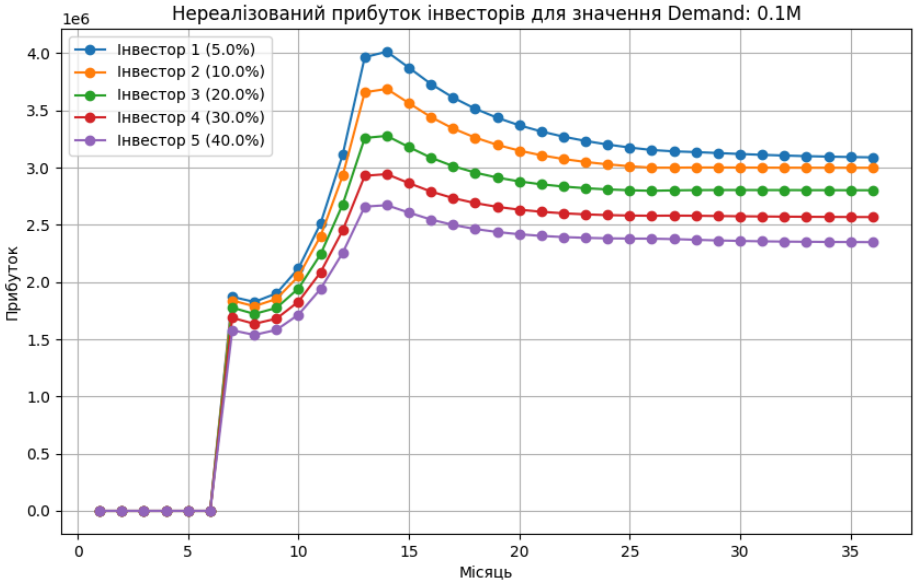




Аналогічно першому сценарію бачимо як лідера інвестора з пропозицією 20% в обох графіках, коли те, що відрізняє ці два графіки, це позиція другого інвестора. Загалом можемо дійти висновку, що чим більший попит, тим вигідніше бути більш заощадливим інвестором.

Наостанок для цього сценарію розглянемо нереалізовані дохідності для різного початкового попиту:



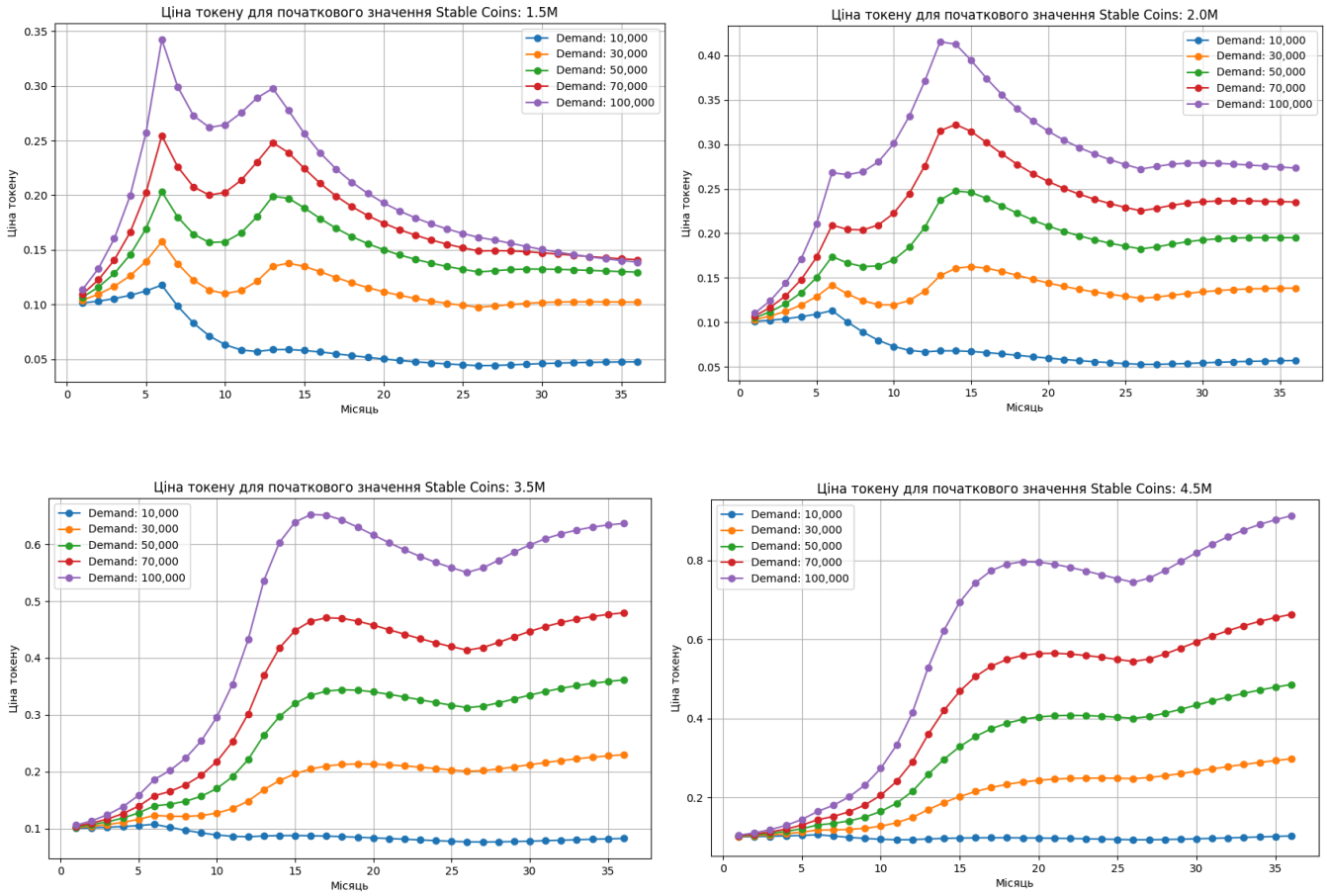


Порівнюючи нереалізовані дохідності, бачимо, що графік з більшим попитом більш різкий, тобто ріст та падіння у ньому більш відчутні. Як висновок можемо сказати, що усім інвесторам найвигідніше було б продати усі токени на 14-ий місяць.

Аналіз третього сценарію

У даному сценарії ми поєднуємо два минулих в одному, для дослідження поведінки ціни токена з різним початковим розміром пулу ліквідності та для кожного різного початкового попиту.

Відразу порівняємо уже бачений нами графік порівняння ціни для різних попитів з 1.5*10^6 кількістю stable coins, з 2.0*10^6, з 3.5*10^6 та 4.5*10^6:



Аналізуючи ці графіки можемо зробити висновок:

- Чим більша початкова кількість stable coins, тим стабільніший графік до різких змін.
- Чим більша початкова кількість stable coins, тим більша різниця у ціні між різними початковими попитами, при цьому, якщо для 1.5*10^6 та 2.0*10^6

графіки збігаються один до одного, то вже для 3.5*10^6 та 4.5*10^6 вони починають розходитись.

- Вже з графіку для 2.0*10^6 і далі зникає різкий спад після періоду cliff, тобто початок продажу токенів не перебиває попит користувачів у ефективності підняття ціни.
- Якщо у графіку 1.5*10^6 після 13-го місяця йде різкий спад після припинення збільшення попиту, то чим більший розмір пулу, тим більш плавний цей спад і після періоду vesting з’являється знову ріст від кількості stable coins рівних або більше 2*10^6.

Висновок

У даній роботі було досліджено формування токен економіки для децентралізованої біржі DEX та був реалізований інструмент для моделювання його поведінки відносно таких змінних:

- Інвестори: cliff, TGE Unlock, vesting та пропозиція.
- Інші користувачі - аналогічно інвесторам.
- Користувачі: попит та зміна попиту.
- Пул ліквідності: кількість токенів та кількість stable coins.

Було досліджено різні початкові розміри пулу ліквідності та побудовано для них графіки:

- Ціни токену від часу.
- Кількості токенів від часу.
- Кількості stable coins від часу.
- Реалізовану дохідність для кожного розміру пулу ліквідності.
- Нереалізовану дохідність для кожного розміру пулу ліквідності.

Було досліджено різний початковий попит користувачів та побудовано для них графіки аналогічно першому сценарію.

Було досліджено третій сценарій, який поєднує в собі перший та другий, для нього побудовано такі графіки:

- Ціни токену від часу.
- Кількості токенів від часу.
- Кількості stable coins від часу.

Для першого сценарію можу сформувати такі висновки:

- Чим менше розмір пулу, тим менша стійкість до змін.
- Збалансована витрата інвесторів більш прибуткова ніж жадібна.
- Чим більше початкове значення кількості stable coins, тим більш вигідно бути заощадливим інвестором, проте і в міру.

- Чим більший пул, тим більша різниця між нереалізованими дохідностями.
- Пул впливає на екстремум в графіку нереалізованої дохідності.

Аналізуючи другий сценарій формуємо такі висновки:

- Найбільший попит не завжди формує найвищу ціну.
- Більший початковий попит впливає більш позитивно на більш ощадливих інвесторів.
- Чим більший попит, тим більш різка зміна нереалізованої дохідності, екстремум при цьому не змінюється і залишається в 14-ому місяці.

Поєднуючи перший та другий сценарії у третьому, можемо підсумувати:

- Чим більша початкова кількість stable coins, тим більш стабільний графік до різких змін.
- Чим більший розмір пулу, тим більша різниця у ціні між різними початковими попитами, при тому, що вони можуть при менших значеннях поступово збігатися, але при більших вже будуть розбігатись.
- Чим більша кількість stable coins, тим більше зникає різкий спад після періоду cliff, тобто початок продажу токенів не перебиває попит користувачів у ефективності підняття ціни.
- Чим більший розмір пулу, тим більш плавний спад після 12-го місяця, коли попит перестав поступово рости.
- Залежно від розміру пулу ліквідності, після періоду vesting падіння перетворюється на ріст.

Загалом, інструмент, який був запрограмований для досліджень, дозволяє розробити рекомендації щодо оптимального управління токеном економіки та сприятиме підвищенню стабільності та вартості токену на ринку децентралізованих фінансів.

Використана література

1. Voshmgir, Shermin. Token Economy: How the Web3 reinvents the Internet. Token Kitchen, 2020.
2. Bashir, Imran. Mastering Blockchain: Unlocking the Power of Cryptocurrencies, Smart Contracts, and Decentralized Applications. Packt Publishing, 2020.
3. Mulligan, Cathy, editor. Blockchain Economics: Implications of Distributed Ledgers. World Scientific Publishing Company, 2020.
4. Lewis, Antony. The Basics of Bitcoins and Blockchains: An Introduction to Cryptocurrencies and the Technology that Powers Them. Mango Publishing, 2018.
5. Kleppmann, Martin. Designing Data-Intensive Applications: The Big Ideas Behind Reliable, Scalable, and Maintainable Systems. O'Reilly Media, 2017.

Додатки

investor.py

```
class Investor:

    def __init__(self, cliff, tge_unlock, vesting, investors_supply, monthly_sell_percentage):

        self.cliff = cliff  # Період, протягом якого інвестори не можуть продавати свої токени

        self.tge_unlock = tge_unlock  # Відсоток токенів, який розблоковується під час генерації токенів.

        self.vesting = vesting  # Період поступового розблокування токенів після cliff періоду


        self.started_investors_supply = investors_supply  # Початкова кількість токенів виділена інвесторам загалом

        self.investors_supply = investors_supply  # Кількість токенів виділена інвесторам загалом

        self.token_holdings = tge_unlock * investors_supply  # Наявна кількість токенів

        self.tokens_sold = 0  # Кількість вже проданих токенів


        self.monthly_sell_percentage = monthly_sell_percentage  # Відсоток токенів, які продають інвестори щомісяця

        self.months_passed = 0  # Лічильник місяців, що пройшли

        # Кількість токенів, які розблоковують щомісяця після cliff періоду

        self.monthly_get_after_cliff = (1 - tge_unlock) * investors_supply / vesting


        self.realized_profit = 0  # Реалізований прибуток

        self.unrealized_profit = 0  # Нереалізований прибуток


    def update_token_holdings(self):

        if self.months_passed > self.cliff:
```

```
            self.token_holdings += self.monthly_get_after_cliff
            self.investors_supply -= self.monthly_get_after_cliff

    def update_profits(self, liquidity_pool, ammount_to_sell):
        # Реалізований прибуток: добуток кількості вже проданих токенів на теперішню ціну
        self.realized_profit += ammount_to_sell * liquidity_pool.current_price

        # Нереалізований прибуток: добуток кількості ще не проданих токенів (включаючі заблоковані) на теперішню ціну
        self.unrealized_profit = (self.started_investors_supply - self.tokens_sold) * liquidity_pool.current_price + self.realized_profit

    def sell_tokens(self, liquidity_pool):
        self.months_passed += 1

        if self.investors_supply >= self.monthly_get_after_cliff:
            self.update_token_holdings()
        else:
            print("All tokens from investors supply were given")

        if self.months_passed <= self.cliff:
            print("Cliff period for investor: tokens cannot be sold.")
            return

        amount_to_sell = self.token_holdings * self.monthly_sell_percentage
        stable_coins_to_receive = amount_to_sell * liquidity_pool.current_price

        print(f"Stable coins to receive: {stable_coins_to_receive}")
```

```
        print(f"Token holdings before sale: {self.token_holdings}")

        if liquidity_pool.update_pool(stable_coins_to_receive, is_addition=False):
            self.token_holdings -= amount_to_sell
            self.tokens_sold += amount_to_sell

        print(f"Token holdings after sale: {self.token_holdings}")
        print(f"Realized profit: {self.realized_profit}")

        self.update_profits(liquidity_pool, amount_to_sell)
```

others.py

```
class Others:
    def __init__(self, cliff, tge_unlock, vesting, others_supply, monthly_sell_percentage):
        self.cliff = cliff  # Період, протягом якого інвестори не можуть продавати свої токени
        self.tge_unlock = tge_unlock  # Відсоток токенів, який розблоковується під час генерації токенів.
        self.vesting = vesting  # Період поступового розблокування токенів після cliff періоду
        self.investors_supply = others_supply  # Кількість токенів виділена інвесторам загалом
        self.token_holdings = tge_unlock * others_supply  # Наявна кількість токенів
        self.monthly_sell_percentage = monthly_sell_percentage  # Відсоток токенів, які продають інвестори щомісяця
        self.months_passed = 0  # Лічильник місяців, що пройшли

        # Кількість токенів, які розблоковують щомісяця після cliff періоду
        self.monthly_get_after_cliff = (1 - tge_unlock) * others_supply / vesting

    def update_token_holdings(self):
        if self.months_passed > self.cliff:
```

```
            self.token_holdings += self.monthly_get_after_cliff

    def sell_tokens(self, liquidity_pool):
        self.months_passed += 1

        self.update_token_holdings()

        if self.months_passed <= self.cliff:
            print("Cliff period for others: tokens cannot be sold.")
            return

        available_to_sell = self.token_holdings
        amount_to_sell = available_to_sell * self.monthly_sell_percentage
        stable_coins_to_receive = amount_to_sell * liquidity_pool.current_price

        print(f"Stable coins to receive: {stable_coins_to_receive}")

        if liquidity_pool.update_pool(stable_coins_to_receive, is_addition=False):
            self.token_holdings -= amount_to_sell
        else:
            print("Failed to sell tokens: Not enough stable coins in the liquidity pool.")
```

liquidity_pool.py

```
class LiquidityPool:
    def __init__(self, stable_coins, liquidity_supply):
        self.stable_coins = stable_coins  # Кількість стейбл коінів у пулі ліквідності
        self.liquidity_supply = liquidity_supply  # Кількість токенів у пулі ліквідності
```

```
        self.multiply_constant = stable_coins * liquidity_supply  # Константа перемноження стейбл коінів та токенів
        self.current_price = self.update_token_price()  # Ціна на токен

    def update_pool(self, amount_stable_coins, is_addition=True):
        if is_addition:
            new_stable_coins = self.stable_coins + amount_stable_coins
        else:
            new_stable_coins = self.stable_coins - amount_stable_coins
            if new_stable_coins <= 0:
                print("Not enough stable coins in the liquidity pool.")
                return False

        new_liquidity_supply = self.multiply_constant / new_stable_coins
        if new_liquidity_supply <= 0:
            print("Not enough tokens in the liquidity pool.")
            return False

        self.stable_coins = new_stable_coins
        self.liquidity_supply = new_liquidity_supply
        self.update_token_price()

        print(f"Updated pool state: {self.get_pool_state()}")

        return True

    def update_token_price(self):
        self.current_price = self.stable_coins / self.liquidity_supply
```

```
        return self.current_price

    def get_pool_state(self):
        return {
            "stable_coins": self.stable_coins,
            "liquidity_supply": self.liquidity_supply,
            "current_price": self.current_price
        }
```

user.py

```
from liquidity_pool import LiquidityPool


class User:
    def __init__(self, demand, demand_growth_rate=0):
        self.demand = demand  # Кількість стейбл коінів, на яку купують користувачі щомісяця
        self.token_holdings = 0  # Кількість токенів, якими володіють користувачі
        self.demand_growth_rate = demand_growth_rate  # Темп зростання попиту

    def update_demand(self, month):
        if month <= 12:  # Зміна попиту тільки для перших 12 місяців
            self.demand *= (1 + self.demand_growth_rate)

    def buy_tokens(self, liquidity_pool: LiquidityPool):
        amount_to_spend = self.demand
        tokens_to_buy = amount_to_spend / liquidity_pool.current_price
        if liquidity_pool.update_pool(amount_to_spend, is_addition=True):
            self.token_holdings += tokens_to_buy
```

```
        else:
            print("Failed to buy tokens: Not enough stable coins in the liquidity pool.")
```

main.py

```
import matplotlib.pyplot as plt
import pandas as pd
import os
from liquidity_pool import LiquidityPool
from investor import Investor
from user import User
from others import Others



def main():
    all_data_stable_coins = []
    all_data_demand = []
    all_data_stable_coins_and_demand = []

    os.makedirs('graphics/stable_coins', exist_ok=True)
    os.makedirs('graphics/demand', exist_ok=True)
    os.makedirs('graphics/stable_coins_and_demand', exist_ok=True)

    initial_stable_coins_values = [1.5 * 1000 * 1000, 2.0 * 1000 * 1000, 2.5 * 1000 * 1000, 3.5 * 1000 * 1000, 4.5 * 1000 * 1000]
    demand_values = [0.01 * 1000 * 1000, 0.03 * 1000 * 1000, 0.05 * 1000 * 1000, 0.07 * 1000 * 1000, 0.1 * 1000 * 1000]

    for initial_stable_coins in initial_stable_coins_values:
        liquidity_pool = LiquidityPool(stable_coins=initial_stable_coins, liquidity_supply=initial_stable_coins * 10)
```

```
        investors = create_investors()

        user = User(demand=0.05 * 1000 * 1000, demand_growth_rate=0.3)

        others = Others(cliff=6, tge_unlock=0.1, vesting=18, others_supply=(80 * 1000 * 1000) - (initial_stable_coins * 10), monthly_sell_percentage=0.1)

        data = simulate_monthly_changes(36, liquidity_pool, investors, user, others)

        df = pd.DataFrame(data)

        df["initial_stable_coins"] = initial_stable_coins

        all_data_stable_coins.append(df)


    plot_token_price_over_time(all_data_stable_coins, "token_price_stable_coins", "Ціна токену для різних значень Stable Coins", x_label="month", group_label="initial_stable_coins", folder="stable_coins")

    plot_combined_graph(all_data_stable_coins, "stable_changes_combined", "Зміни стейбл коінів для різних значень Stable Coins", x_label='month', y_label='Кількість стейбл коінів', value_label='stable_coins', group_label='initial_stable_coins', folder='stable_coins')

    plot_combined_graph(all_data_stable_coins, "token_changes_combined", "Зміни токенів для різних значень Stable Coins", x_label='month', y_label='Кількість токенів', value_label='liquidity_supply', group_label='initial_stable_coins', folder='stable_coins')


    for initial_stable_coins in initial_stable_coins_values:

        plot_investor_profits(all_data_stable_coins, f'investors_realized_profit_stable_coins_{initial_stable_coins/1000000}M', f'Реалізований прибуток інвесторів для початкового значення Stable Coins: {initial_stable_coins/1000000}M', x_label='month', investor_type='realized', filter_value=initial_stable_coins, filter_type='initial_stable_coins', folder='stable_coins', investors=create_investors())

        plot_investor_profits(all_data_stable_coins, f'investors_unrealized_profit_stable_coins_{initial_stable_coins/1000000}M', f'Нереалізований прибуток інвесторів для початкового значення Stable Coins: {initial_stable_coins/1000000}M', x_label='month', investor_type='unrealized', filter_value=initial_stable_coins, filter_type='initial_stable_coins', folder='stable_coins', investors=create_investors())


    for demand in demand_values:

        liquidity_pool = LiquidityPool(stable_coins=1.5 * 1000 * 1000, liquidity_supply=15 * 1000 * 1000)

        investors = create_investors()
```

```
        user = User(demand=demand, demand_growth_rate=0.3)

        others = Others(cliff=6, tge_unlock=0.1, vesting=18, others_supply=65 * 1000 * 1000,
monthly_sell_percentage=0.1)

        data = simulate_monthly_changes(36, liquidity_pool, investors, user, others)

        df = pd.DataFrame(data)

        df["demand"] = demand

        all_data_demand.append(df)


    plot_token_price_over_time(all_data_demand, "token_price_demand", "Ціна токену для різних значень
Demand", x_label="month", group_label="demand", folder="demand")

    plot_combined_graph(all_data_demand, "stable_changes_combined", "Зміни стейбл коінів для різних значень
Demand", x_label='month', y_label='Кількість стейбл коінів', value_label='stable_coins', group_label='demand',
folder='demand')

    plot_combined_graph(all_data_demand, "token_changes_combined", "Зміни токенів для різних значень
Demand", x_label='month', y_label='Кількість токенів', value_label='liquidity_supply', group_label='demand',
folder='demand')


    for demand in demand_values:

        plot_investor_profits(all_data_demand, f'investors_realized_profit_demand_{demand/1000000}M',
f'Реалізований прибуток інвесторів для значення Demand: {demand/1000000}M', x_label='month',
investor_type='realized', filter_value=demand, filter_type='demand', folder='demand', investors=create_investors())

        plot_investor_profits(all_data_demand, f'investors_unrealized_profit_demand_{demand/1000000}M',
f'Нереалізований прибуток інвесторів для значення Demand: {demand/1000000}M', x_label='month',
investor_type='unrealized', filter_value=demand, filter_type='demand', folder='demand',
investors=create_investors())


    for initial_stable_coins in initial_stable_coins_values:

        data_for_current_stable_coin = []

        for demand in demand_values:

            liquidity_pool = LiquidityPool(stable_coins=initial_stable_coins, liquidity_supply=initial_stable_coins *
10)

            investors = create_investors()

            user = User(demand=demand, demand_growth_rate=0.3)
```

```
        others = Others(cliff=6, tge_unlock=0.1, vesting=18, others_supply=(80 * 1000 * 1000) -
(initial_stable_coins * 10), monthly_sell_percentage=0.1)

        data = simulate_monthly_changes(36, liquidity_pool, investors, user, others)

        df = pd.DataFrame(data)

        df["initial_stable_coins"] = initial_stable_coins

        df["demand"] = demand

        data_for_current_stable_coin.append(df)

        all_data_stable_coins_and_demand.append(df)



    plot_token_price_over_time(data_for_current_stable_coin,
f"token_price_stable_coins_{initial_stable_coins/1000000}M", f"Ціна токену для початкового значення Stable
Coins: {initial_stable_coins/1000000}M", x_label="month", group_label="demand",
folder="stable_coins_and_demand")

    plot_combined_graph(data_for_current_stable_coin,
f"stable_changes_combined_{initial_stable_coins/1000000}M", f"Зміни стейбл коінів для початкового значення
Stable Coins: {initial_stable_coins/1000000}M", x_label='month', y_label='Кількість стейбл коінів',
value_label='stable_coins', group_label='demand', folder='stable_coins_and_demand')

    plot_combined_graph(data_for_current_stable_coin,
f"token_changes_combined_{initial_stable_coins/1000000}M", f"Зміни токенів для початкового значення
Stable Coins: {initial_stable_coins/1000000}M", x_label='month', y_label='Кількість токенів',
value_label='liquidity_supply', group_label='demand', folder='stable_coins_and_demand')




def create_investors():

  return [

    Investor(cliff=6, tge_unlock=0.1, vesting=18, investors_supply=4 * 1000 * 1000,
monthly_sell_percentage=0.05),

    Investor(cliff=6, tge_unlock=0.1, vesting=18, investors_supply=4 * 1000 * 1000,
monthly_sell_percentage=0.1),

    Investor(cliff=6, tge_unlock=0.1, vesting=18, investors_supply=4 * 1000 * 1000,
monthly_sell_percentage=0.2),

    Investor(cliff=6, tge_unlock=0.1, vesting=18, investors_supply=4 * 1000 * 1000,
monthly_sell_percentage=0.3),
```

```
        Investor(cliff=6, tge_unlock=0.1, vesting=18, investors_supply=4 * 1000 * 1000,
monthly_sell_percentage=0.4)
    ]


def simulate_monthly_changes(months, liquidity_pool, investors, user, others):
    data = []

    for month in range(1, months + 1):
        user.buy_tokens(liquidity_pool=liquidity_pool)
        user.update_demand(month)
        for investor in investors:
            investor.sell_tokens(liquidity_pool=liquidity_pool)
        others.sell_tokens(liquidity_pool=liquidity_pool)

        pool_state = liquidity_pool.get_pool_state()
        pool_state["month"] = month
        for i, investor in enumerate(investors):
            pool_state[f'investor_{i+1}_realized_profit'] = investor.realized_profit
            pool_state[f'investor_{i+1}_unrealized_profit'] = investor.unrealized_profit
            pool_state[f'investor_{i+1}_monthly_sell_percentage'] = investor.monthly_sell_percentage
        data.append(pool_state)

    return data


def plot_token_price_over_time(all_data, graphic_file_name, graphic_name, x_label, group_label, folder=""):
    plt.figure(figsize=(10, 6))
```

```
    for df in all_data:
        plt.plot(df[x_label], df["current_price"], marker='o', linestyle='-',
                 label=f'{group_label.capitalize()}: {df[group_label].iloc[0]:,.0f}')

    plt.title(graphic_name)
    plt.xlabel("Місяць")
    plt.ylabel("Ціна токену")
    plt.legend()
    plt.grid(True)
    plt.savefig(f'graphics/{folder}/{graphic_file_name}.png')
    plt.close()


def plot_combined_graph(all_data, graphic_file_name, graphic_name, x_label, y_label, value_label, group_label, folder=""):
    plt.figure(figsize=(10, 6))

    for df in all_data:
        plt.plot(df[x_label], df[value_label], marker='o', linestyle='-', label=f'{group_label.capitalize()}: {df[group_label].iloc[0]:,.0f}')

    plt.title(graphic_name)
    plt.xlabel(x_label.capitalize())
    plt.ylabel(y_label)
    plt.legend()
    plt.grid(True)
    plt.savefig(f'graphics/{folder}/{graphic_file_name}.png')
```

```
    plt.close()



def plot_investor_profits(all_data, graphic_file_name, graphic_name, x_label, investor_type, filter_value,
filter_type, folder="", investors=None):
    plt.figure(figsize=(10, 6))

    for df in all_data:
        if df[filter_type].iloc[0] == filter_value:
            for i, investor in enumerate(investors):
                plt.plot(df[x_label], df[f'investor_{i+1}_{investor_type}_profit'], marker='o', linestyle='-',
                         label=f'Інвестор {i+1} ({investor.monthly_sell_percentage * 100}%)')

    plt.title(graphic_name)
    plt.xlabel("Місяць")
    plt.ylabel("Прибуток")
    plt.legend()
    plt.grid(True)
    plt.savefig(f'graphics/{folder}/{graphic_file_name}.png')
    plt.close()



if __name__ == "__main__":
    main()
```