

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА**

Факультет комп'ютерних наук та кібернетики
Кафедра теоретичної кібернетики

Кваліфікаційна робота
на здобуття ступеня бакалавра
за спеціальністю 122 Комп'ютерні науки
на тему:

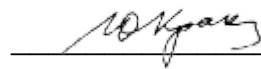
РОЗРОБКА БОТА ДЛЯ ТОРГІВЛІ КРИПТОВАЛЮТОЮ

Виконав: студент 4-го курсу
Боявець Владислав Володимирович



(підпис)

Науковий керівник:
професор, доктор фіз.-мат. наук
Крак Юрій Васильович



(підпис)

Засвідчую, що в цій роботі
немає запозичень з праць інших авторів
без відповідних посилань.

Студент



(підпис)

Роботу розглянуто й допущено до захисту
на засіданні кафедри теоретичної
кібернетики

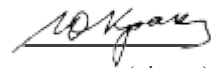
« 1 » Червня 2022 р.,

протокол № 11

Завідувач кафедри

доктор фіз.-мат. наук, професор

Юрій КРАК



(підпис)

Київ-2022

РЕФЕРАТ

Обсяг роботи 47 сторінок, 13 ілюстрації, 22 джерела посилань.

БЛОКЧЕЙН, КРИПТОВАЛЮТА, РИНОК, ТЕХНІЧНИЙ АНАЛІЗ,
BITCOIN, TELEGRAM, BINANCE, JAVA

Об'єктом роботи є дослідження технології blockchain, порівняння з існуючими технологіями зберігання даних, а також дослідження криптовалют та цілого криптовалютного-ринку.

Метою роботи є розробка торгового бота «Crypto Currency Tracking Bot» з метою економити час трейдерам, а також планами на подальше покращення певного функціоналу відповідно до потреб користувачів.

Методи розроблення: аналіз технологій blockchain, принципів роботи мережі blockchain, криптовалют та криптовалютних ринків.

Інструменти розроблення. Для розробки реалізації були використані інтерактивні середовища розробки IntelliJ Idea Ultimate Edition. Було використано мову програмування Java та фреймворк Spring. Для створення UI використано TelegramBot API.

Результат роботи. Виконано опис технології blockchain, її історія, пояснення принципів її роботи. Також була досліджена тема криптовалют та криптовалютного ринку. Розроблено торговий бот для моніторингу, аналізу та створення торгових угод на ринку. Програмний код було викладено в мережу на правах загального доступу та зроблено доступним усім розробникам. Було продемонстровано його роботу та надано інструкцію його використання.

ЗМІСТ

РЕФЕРАТ	2
ЗМІСТ	3
Розділ 1. Технологія blockchain і її використання	6
1.1 Історія блокчейну та коротка характеристика	6
1.2 Що таке blockchain?	7
1.3 Як працює blockchain?	8
Розділ 2. Криптовалюта та її майбутнє	11
2.1 Що таке криптовалюта?	11
2.2 Які бувають типи криптовалют?	12
2.3 Чи законні криптовалюти?	13
2.4 Порівняння фіатних та крипто валют	14
2.5 Переваги та недоліки криптовалюти	15
2.6 Перспективи розвитку криптовалютного ринку	18
Розділ 3. Підготовка до розробки торгового бота	20
3.1 Мова програмування та фреймворк	20
3.2 База даних та її переваги	21
3.3 Telegram Bot API і його реалізація	23
3.4 Що таке технічний аналіз та як з ним працювати	25
3.5 Допоміжні бібліотеки використані для розробки	26
3.6 Про біржу Binance та Binance API	27
3.7 Розгортання бота в хмарному середовищі	29
Розділ 4. Реалізація торгового бота	31
4.1 Опис ключових етапів реалізації	31
4.2 Демонстрація роботи бота	42
ВИСНОВКИ	45
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ	46

ВСТУП

Оцінка сучасного стану об'єкта дослідження. У сьогоднішній день технологія блокчейн стає все більше і більше популярною. Це відбувається через створення криптовалют і зростання їх ціни, що в свою чергу викликає великий інтерес у людей. Це приводить в ринок криптовалют надзвичайно велику кількість грошей. Дуже багато інвесторів переводять свої заощадження з фондового ринку в ринок криптовалют. А саме головне - це його волатильність, що неодмінно приваблює трейдерів.

Актуальність роботи та підстави для її виконання. Криптовалютний ринок зростає кожного дня. Мільйони людей інвестують свої гроші в Bitcoin та йому подібні. Тисячі компаній створюють свої власні валюти, з вірою у їх ріст та рекламу компанії. З одного боку це все дуже добре для трейдерів, тому що в них з'являється надзвичайного багато матеріалу, з яким можна працювати. З іншого - моніторинг ринку забирає надзвичайно велику кількість часу. Саме для цього створюються боти, які можуть спостерігати за ціною і повідомляти трейдера коли вона досягне того чи іншого значення. Це надзвичайно сильно економить людям час, який вони можуть витратити на свою сім'ю, своє здоров'я чи навіть зробити щось корисне для навколишнього середовища.

Мета й завдання роботи. Метою дипломної роботи є розробка спеціального бота, з трьома основними функціями:

1. Звичайний моніторинг ринку і повідомлення користувача про різкий рух ціни.
2. Моніторинг ринку за допомогою технічного аналізу, що дасть можливість спрогнозувати подальший рух ціни вгору або вниз.

3. Автоматичне створення торгових угод на покупку та продажу певної криптовалюти на основі даних з пункту 2.

Можливі сфери застосування. Реалізація даного бота з функціями моніторингу доступна абсолютно кожному telegram користувачу за посиланням [@crypto_trkr_bot](https://t.me/crypto_trkr_bot).

Що ж, тепер давайте розглянемо необхідну теорію, щоб краще розуміти з чим нам належить працювати.

Розділ 1. Технологія blockchain і її використання

1.1 Історія блокчейну та коротка характеристика

Технологія блокчейн була описана в 1991 році вченим-дослідником Стюартом Хабером і Скоттом Сторнеттою. Вони хотіли запровадити практичне в обчислювальному відношенні рішення для цифрових документів із відміткою часу, щоб вони не могли бути датовані або підроблені. Вони розробляють систему, що використовує концепцію криптографічно захищеного ланцюга блоків для зберігання документів із мітками часу.

У 1992 році Merkle Trees (Дерево Меркла) були включені в дизайн, що робить блокчейн більш ефективним, дозволяючи збирати кілька документів в один блок. Merkle Trees використовуються для створення «захищеного ланцюжка блоків». Він зберігає серію записів даних, і кожен запис даних пов'язаний з попереднім. Найновіший запис у цьому ланцюжку містить історію всього ланцюжка. Однак ця технологія не використовувалася, а патент втратив чинність у 2004 році.

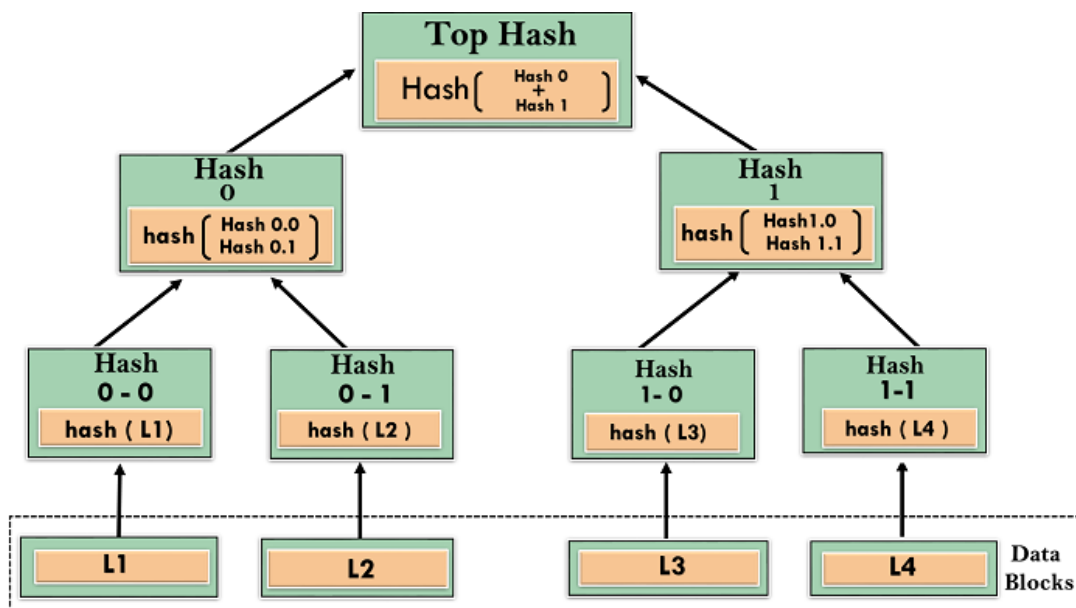


Рис. 1.1 - Merkle Tree (Дерево Меркла)

У 2004 році вчений-комп'ютерник і активіст з криптографії Хел Фінні представив систему під назвою Reusable Proof Of Work (RPoW) як прототип для цифрової готівки. Це був значний ранній крок в історії криптовалют. Система RPoW працювала, отримуючи незамінний або незамінний токен підтвердження роботи на основі Hashcash, створюючи підписаний RSA токен, який можна було передавати від людини до людини.

RPoW вирішив проблему подвійних витрат, зберігаючи право власності на токени, зареєстровані на надійному сервері. Цей сервер був розроблений, щоб дозволити користувачам у всьому світі перевірити його правильність та цілісність у режимі реального часу.

1.2 Що таке blockchain?

Blockchain (в перекладі з англійської - ланцюг блоків, від block – блок, chain – ланцюг) – це децентралізована база даних, зростаючий список записів, які називаються блоками, пов'язаних між собою за допомогою криптографічних алгоритмів. Спільний для всіх, незмінний реєстр, призначений для запису транзакцій, обліку активів і побудови довірчих відносин. Кожен блок містить в собі криптографічний хеш попереднього блоку, мітка часу створення блоку та транзакції (в основному утворюють дерево Меркла (рис. 1.1)). За задумом блокчейн дані, які були записані до блокчейну неможливо замінити. Це вдалось досягти, за допомогою зміни значення хешу блока, при зміні даних, що в свою чергу викликає конфлікт у майбутніх ланках ланцюга.

Часто люди задаються питанням "А як же технологія blockchain може виявитись корисною і чим саме відрізняється від вже існуючих альтернативних технологій?"

Blockchain являє собою особливий тип баз даних (розподіленої), тому його можна використовувати у тих же самих випадках, що і будь-які інші види баз даних. Звичайно в певних випадках зовсім немає сенсу використовувати дану технологію, так як звичайна база даних може чудово справлятися з поставленою задачею, що є значно простішим в реалізації.

Основною перевагою є безпечна передача важливої інформації будь-яким людям, які можуть бути не зовсім чесними по відношенню до вас. Або ви боїтеся, що тим чи іншим способом ваші дані можуть зазнати змін, які зіграють вам не на руку. Технологія blockchain унеможливорює подібні ситуації.

1.3 Як працює blockchain?

Blockchain – це цифровий реєстр. В більшості випадків він децентралізований, розподілений та загальнодоступний. В цей реєстр вносяться певного роду транзакції, які групуються в блоки. Кожен користувач зберігає абсолютну копію ланцюга блоків на своєму комп'ютері. Завдяки цьому кожен клієнт може валідувати транзакції самостійно, без третьої сторони (що в звичайному житті відіграють нотаріуси, банки і т.п.). Всі учасники дотримуються одного єдиного протоколу валідації та зв'язку блоків. Насправді транзакції в блокчейні реально підмінити, але завдяки масовій співпраці кожного комп'ютера в системі, підроблені блоки одразу виявляються та відхиляються, а ланцюг стає не валідним.

Блок – набір підтверджених транзакцій, хеш-код яких утворює дерево Меркла (рис. 1.1). Перший блок створюється руками і називається генезисним. Кожен наступний блок містить в собі хеш попереднього блоку. Саме так вони зв'язуються. Такі пов'язані блоки утворюють собою ланцюг. Це дає змогу перевірити цілісність ланцюга, пройшовши по всіх блоках з

початку до кінця. Існує шанс, що два нових блока створюються одночасно. Це породжує тимчасове розгалуження ланцюга. Тоді істиною вважається ланцюг, який містить найдовшу послідовність блоків. Звідси випливає, що клієнти блокчейну матимуть трішки різні ланцюги, адже кожен зберігає лише відому йому комбінацію блоків. Тому кожного разу, при одержанні нової істинної версії ланцюга вони розширюють або перезаписують свій локальний ланцюг і передають зміни наступним користувачам. Технологія концентрується на розширенні новими блоками, аніж перезаписі вже існуючих. Тому чим більше блоків над вже існуючим, тим більше він стає істинним.

Майнер – особа, а точніше її комп'ютер, який підтримує життєдіяльність мережі blockchain. Задачею кожного майнера є пошук нових блоків. Подібно до того, як корисні копалини добувають з надр землі за допомогою великої техніки, схоже і для блокчейн також використовуються великі групи відеокарт. Вони вирішують певні математичні головоломки, створені алгоритмом блокчейна. Саме майнери роблять мережу надійною, перевіряючи всю інформацію про транзакції. Майнери завантажують всю історію блокчейну і збирають нові транзакції в блок. Якщо блок зібраних транзакцій прийнятий і перевірений іншими майнерами, то майнер отримує винагороду за блок.

Децентралізація – явище, завдяки якому втрачається єдиний центр управління. У blockchain дані містяться в одноранговій (peer-to-peer) мережі, що зменшує ризики централізованого утримання даних. Блокчейн використовує криптографію з відкритим ключем, що підвищує рівень анонімності та безпеки користувача. Відкритий ключ є його адресою, що заміняє ім'я, фамілію чи інші ознаки, які ідентифікують особу. Приватний ключ – свого роду пароль, який надає власнику доступ до його цифрових активів та можливість керувати ними. Ланцюг копіюється в кожному вузлі мережі. Так як всі ланцюги однопікані, кожен

з них проходить перевірку на істинність. Транзакції передаються в мережу за допомогою певної програми. Під час майнінгу блоку, найближчі вузли перевіряють коректність транзакцій. Якщо все вірно вони їх підтверджують, тим самим додають до нового блоку, після чого розповсюджують його сусіднім вузлам.

Розділ 2. Криптовалюта та її майбутнє

2.1 Що таке криптовалюта?

Криптовалюта – це цифрова або віртуальна валюта, яка захищена криптографією, завдяки чому її майже неможливо підробити або використати кілька разів. Більшість криптовалют є децентралізованими мережами, заснованими на технології блокчейн, створеної за допомогою розподіленої мережі комп'ютерів. Визначальною особливістю криптовалют є те, що вони, як правило, не випускаються жодним центральним органом влади, що робить їх теоретично несприятливими до державного втручання або маніпуляцій.

Криптовалюти – це цифрові або віртуальні валюти, що ґрунтуються на криптографічних системах. Вони дозволяють здійснювати безпечні онлайн-платежі без використання сторонніх посередників. «Крипто» відноситься до різних алгоритмів шифрування та криптографічних методів, які захищають ці записи, таких як шифрування з еліптичною кривою, пари відкритих та закритих ключів і функції хешування.

Криптовалюти можна добувати або купувати на криптовалютних біржах. Не всі сайти електронної комерції дозволяють робити покупки за допомогою криптовалют. Насправді криптовалюти, навіть такі популярні, як біткойн, майже не використовуються для роздрібних транзакцій. Проте різко зростаюча вартість криптовалют зробила їх популярними як інструменти торгівлі. В обмеженій мірі вони також використовуються для транскордонних переказів.

Криптовалюти – це нова парадигма грошей. Вони обіцяють оптимізувати існуючу фінансову архітектуру, щоб зробити її швидшою та дешевшою. Їхня технологія та архітектура децентралізують існуючі

грошові системи та дають можливість сторонам, які здійснюють операції, обмінювати вартість та гроші незалежно від посередницьких установ, таких як банки.

2.2 Які бувають типи криптовалют?

Біткоїн - найпопулярніша і найдорожча криптовалюта. Анонімна людина на ім'я Сатоші Накамото винайшла його та представив світу за допомогою білої книги в 2008 році. Сьогодні на ринку представлені тисячі криптовалют.

Кожна криптовалюта стверджує, що має різні функції та специфікації. Наприклад, ефір Ethereum продає себе як газ для базової платформи смарт-контрактів. XRP Ripple використовується банками для полегшення переказів між різними регіонами.

Біткоїн, який був доступний для людства в 2009 році, залишається найбільш широко торговою та покритою криптовалютою. Станом на листопад 2021 року в обігу перебувало понад 18,8 мільйонів біткойнів із загальною ринковою капіталізацією близько 1,2 трильйона доларів. Максимальний ліміт біткойнів 21 мільйон.

На хвилі успіху біткойна було запущено багато інших криптовалют, відомих як «альткойни». Деякі з них є клонами або форками біткойна, а інші є новими валютами, створеними з нуля. Серед них Solana, Litecoin, Ethereum, Cardano і EOS. До листопада 2021 року сукупна вартість усіх існуючих криптовалют перевищила 2,1 трильйона доларів — біткоїн становив приблизно 41% від загальної вартості.

2.3 Чи законні криптовалюти?

Фіатні валюти отримують свої повноваження як засоби транзакцій від уряду або монетарних органів. Наприклад, кожна доларова банкнота підтримується Федеральним резервом.

Але криптовалюти не підтримуються жодними державними чи приватними організаціями. Тому було важко обґрунтувати їх правовий статус у різних фінансових юрисдикціях по всьому світу. Не допомагає той факт, що криптовалюти в основному функціонують за межами більшості існуючих фінансових інфраструктур. Правовий статус криптовалют впливає на їх використання в щоденних транзакціях і торгівлі. У червні 2019 року Група фінансових заходів (FATF) рекомендувала, щоб банківські перекази криптовалют підпорядковувалися вимогам свого Правила подорожей, яке вимагає відповідності AML.

Станом на грудень 2021 року Сальвадор був єдиною країною у світі, яка дозволила біткойн як законний платіжний засіб для грошових операцій. У решті світу регулювання криптовалют залежить від юрисдикції.

Проте вже 17 лютого 2022 року Президент України Володимир Зеленський підписав закон «Про віртуальні активи», який ухвалила Верховна Рада України. Тож тепер криптовалюта офіційно легалізована в Україні.

Японський Закон про платіжні послуги визначає біткойн як законну власність. Криптовалютні біржі, що працюють в країні, підлягають збору інформації про клієнта та деталей, що стосуються банківського переказу. Китай заборонив біржі криптовалют і майнінг у своїх кордонах.

Криптовалюти є легальними в Європейському Союзі. Похідні інструменти та інші продукти, які використовують криптовалюти, повинні кваліфікуватися як «фінансові інструменти». У червні 2021 року

Європейська комісія оприлюднила положення про ринки крипто-активів (MiCA), яке встановлює гарантії регулювання та встановлює правила для компаній або постачальників, які надають фінансові послуги за допомогою криптовалют. У Сполучених Штатах найбільший і найскладніший фінансовий ринок. У світі крипто-похідні, такі як ф'ючерси на біткойн, доступні на Чиказькій товарній біржі. Комісія з цінних паперів і бірж (SEC) заявила, що Bitcoin та Ethereum не є цінними паперами.

Хоча криптовалюти вважаються формою грошей, служба внутрішніх доходів (IRS) розглядає їх як фінансовий актив або власність. І, як і в більшості інших інвестицій, якщо ви отримуєте прибуток від продажу або торгівлі криптовалютами, уряд хоче отримати частину прибутку. 20 травня 2021 року Міністерство фінансів США оголосило про пропозицію, згідно з якою платники податків повідомлятимуть IRS про будь-яку операцію з криптовалютою на суму понад 10 000 доларів США. Як саме IRS оподатковуватиме надходження — як приріст капіталу чи звичайний дохід — залежить від того, скільки часу платник податків зберігав криптовалюту.

2.4 Порівняння фіатних та крипто валют

Обидві ці форми валюти значною мірою отримують свою цінність від широкого визнання, яке вони мають у всьому світі. Більше визнання означає більше довіри. Вони також поділені — так само, як рупію можна розділити на 100 пайс, 1 біткойн можна розділити лише на 0,00000001 BTC. Як і фіатні гроші, криптовалюту можна використовувати для оплати послуг або покупки речей. Їх також можна подарувати, використати як запас цінності.

На відміну від фіатних грошей, криптовалюта не регулюється центральними органами влади та не підтримується урядами. Це робить

віртуальну валюту менш надійною, ніж реальну (тверді гроші або цифрові гроші на банківських рахунках). Криптовалюта також набагато мінливіша, ніж фіатні гроші. Нестабільність в першу чергу обумовлена спекулятивним характером торгівлі, коли інвестори зосереджені на швидкому створенні багатства шляхом резервування прибутку.

Криптовалюти не вимагають посередника для підтвердження транзакції, як банк у випадку фіатних грошей. Крипто-транзакції перевіряються за допомогою технології блокчейн, завдяки якій вся торгова діяльність фіксується постійно, що підвищує безпеку кожної біржі.

2.5 Переваги та недоліки криптовалюти

Криптовалюти були введені з наміром змінити фінансову інфраструктуру. Однак, як і в кожній революції, тут є компроміси. На сучасному етапі розвитку криптовалют існує багато відмінностей між теоретичним ідеалом децентралізованої системи з криптовалютами та його практичною реалізацією.

Деякі переваги та недоліки криптовалют полягають у наступному.

Переваги

- Криптовалюти представляють нову децентралізовану парадигму грошей. У цій системі централізовані посередники, такі як банки та грошово-кредитні установи, не потрібні для забезпечення довіри та поліцейських операцій між двома сторонами. Таким чином, система з криптовалютами виключає можливість єдиної точки збою, наприклад, великого банку, що спричинить каскад криз у всьому

світі, наприклад той, який був спровокований у 2008 році через збій установ у Сполучених Штатах.

- Криптовалюти обіцяють спростити переказ коштів безпосередньо між двома сторонами, без потреби в довіреній третій стороні, як-от банк або компанія, що займається кредитними картками. Такі децентралізовані передачі забезпечуються використанням відкритих і закритих ключів, а також різних форм заохочення, таких як підтвердження роботи або підтвердження участі.
- Оскільки вони не використовують сторонніх посередників, перекази криптовалюти між двома сторонами, які здійснюють транзакцію, відбуваються швидше, ніж стандартні грошові перекази. Гарним прикладом таких децентралізованих переказів є флеш-кредити в децентралізованих фінансах. Ці позики, які обробляються без забезпечення застави, можуть бути виконані за лічені секунди та використовуються в торгівлі.
- Інвестиції в криптовалюту можуть приносити прибуток. Ринки криптовалют різко зросли в ціні за останнє десятиліття, колись досягнувши майже 2 трильйони доларів. Станом на 20 грудня 2021 року вартість біткойн на криптовалютних ринках становила понад 862 мільярди доларів.
- Економіка грошових переказів тестує один з найпомітніших випадків використання криптовалюти. Наразі такі криптовалюти, як біткойн, служать проміжними валютами для оптимізації грошових переказів через кордони. Таким чином, фіатна валюта конвертується в біткойн (або іншу криптовалюту), передається через кордони і згодом конвертується в фіатну валюту призначення. Цей метод спрощує процес переказу грошей і робить його дешевшим.

Недоліки

- Хоча вони стверджують, що є анонімною формою транзакції, насправді криптовалюти є псевдонімами. Вони залишають цифровий слід, який можуть розшифрувати такі агентства, як Федеральне бюро розслідувань (ФБР). Це відкриває можливості урядів чи федеральних органів відстежувати фінансові операції звичайних громадян.
- Криптовалюти стали популярним інструментом для злочинців для таких злочинних дій, як відмивання грошей і незаконні покупки. Випадок Dread Pirate Roberts, який керував маркетплейсом для продажу наркотиків у темній мережі, вже добре відомий. Криптовалюти також стали улюбленими хакерами, які використовують їх для діяльності з програмним забезпеченням-вимагачем.
- Теоретично криптовалюти призначені для децентралізації, їх багатство розподіляється між багатьма сторонами на блокчейні. Насправді власність дуже сконцентрована. Наприклад, дослідження Массачусетського технологічного інституту показало, що лише 11 000 інвесторів володіли приблизно 45% зростання вартості біткойна.
- Однією з переваг криптовалют є те, що будь-хто може добувати їх за допомогою комп'ютера з під'єднанням до Інтернету. Однак майнінг популярних криптовалют вимагає значної енергії, іноді стільки ж енергії, скільки споживають цілі країни. Дорогі витрати на енергію в поєднанні з непередбачуваністю майнінгу сконцентрували видобуток серед великих компаній, чії доходи сягають мільярдів доларів. Згідно з дослідженням MIT, на 10% майнерів припадає 90% його потужностей.
- Хоча блокчейни криптовалют дуже безпечні, інші криптосховища, такі як біржі та гаманці, можуть бути зламані. Багато

криптовалютних бірж і гаманців були зламани протягом багатьох років, іноді призводячи до викрадення «монет» на мільйони доларів.

- Криптовалюти, які торгуються на публічних ринках, страждають від волатильності цін. У грудні 2017 року біткойн зазнав стрімких сплесків і обвалів, піднявшись до \$17 738 у грудні 2017 року, а потім впав до \$7 575 протягом наступних трьох місяців. Тому деякі економісти вважають криптовалюти недовговічною примхою або спекулятивною бульбашкою.

2.6 Перспективи розвитку криптовалютного ринку

Ми бачили, що протягом останнього року біткойн вразив кілька нових найвищих цін за всю історію — за якими слідували значні падіння — і більше інституційного бай-іну від великих компаній. Ethereum, друга за величиною криптовалюта, також наприкінці минулого року досягла свого нового історичного максимуму. Урядові чиновники США та адміністрація Байдена все частіше виявляють інтерес до нових правил щодо криптовалюти.

За весь час інтерес людей до криптовалют різко зріс: це гаряча тема не лише серед інвесторів, а й у популярній культурі, завдяки всім, від давніх інвесторів, таких як Ілон Маск, до цієї дитини з вашої середньої школи на Facebook.

Багато в чому 2021 рік став «проривом», — каже Дейв Абнер, керівник глобального розвитку Gemini, популярної біржі криптовалют.

Але галузь знаходиться тільки в зародковому стані і постійно розвивається. Це велика частина того, чому кожен новий максимум біткойна може легко супроводжуватися великими падіннями. Важко передбачити, куди рухатимуться справи в довгостроковій перспективі, але

в найближчі місяці експерти стежать за темами від регулювання до інституційного прийняття крипто-платежів, щоб спробувати краще зрозуміти ринок.

Розділ 3. Підготовка до розробки торгового бота

3.1 Мова програмування та фреймворк

Розробку торгового бота я буду проводити за допомогою мови програмування Java.

Java — об'єктно-орієнтована мова програмування, випущена 1995 року компанією «Sun Microsystems» як основний компонент платформи Java. З 2009 року мовою займається компанія «Oracle», яка того року придбала «Sun Microsystems». В офіційній реалізації Java-програми компілюються у байт-код, який при виконанні інтерпретується віртуальною машиною для конкретної платформи.

В основі проекту лежатиме Spring framework.

Spring Framework — це програмний каркас (фреймворк) з відкритим кодом та контейнери з підтримкою інверсії управління для платформи Java.

Основні особливості Spring Framework можуть бути використані будь-яким додатком Java, але є розширення для створення веб додатків на платформі Java EE. Незважаючи на це, Spring Framework не нав'язує якоїсь конкретної моделі програмування, Spring Framework став популярним в спільноті Java як альтернатива, або навіть доповнення моделі Enterprise JavaBean (EJB).

Також я використовуватиму додаткові бібліотеки Spring:

- **Spring Boot** дозволяє легко створювати автономні, виробничі програми на основі Spring, які можна «просто запускати». Розробники дотримуються впевненого погляду на платформу Spring і сторонні бібліотеки, щоб користувачі могли розпочати роботу з

мінімальною суетою. Більшість програм Spring Boot потребують мінімальної конфігурації Spring.

- **Spring Data JPA**, частина більшого сімейства Spring Data, дозволяє легко реалізувати репозиторії на основі JPA. Цей модуль має справу з розширеною підтримкою рівнів доступу до даних на основі JPA. Це полегшує створення програм на основі Spring, які використовують технології доступу до даних.

Реалізація рівня доступу до даних програми була громіздкою протягом досить тривалого часу. Для виконання простих запитів, а також для розбиття на сторінки та аудиту потрібно написати занадто багато шаблонного коду. Spring Data JPA має на меті значно покращити реалізацію рівнів доступу до даних, зменшуючи зусилля до фактично необхідного обсягу. Як розробник, ви пишете інтерфейси свого репозиторію, включаючи спеціальні методи пошуку, і Spring автоматично забезпечує реалізацію.

- **Spring boot devtools**. Метою модуля є спроба покращити час розробки під час роботи з додатком Spring Boot. Він допомагає вносити зміни в уже запущений сервер без його перезавантаження.

3.2 База даних та її переваги

Для зберігання інформації про користувачів бота, а також всієї необхідної інформації для моніторингу криптовалют я використовуватиму PostgreSQL.

PostgreSQL — є безкоштовною системою керування реляційними базами даних (RDBMS) з відкритим вихідним кодом, яка робить акцент на розширюваності та відповідності SQL. Спочатку він називався

POSTGRES, посилаючись на його походження як наступник бази даних Ingres, розробленої в Каліфорнійському університеті в Берклі. У 1996 році проект був перейменований на PostgreSQL, щоб відобразити його підтримку SQL. Після огляду в 2007 році команда розробників вирішила зберегти назву PostgreSQL і псевдонім Postgres.

PostgreSQL містить транзакції з властивостями атомарності, узгодженості, ізоляції, довговічності (ACID), автоматично оновлюваними представленнями, матеріалізованими представленнями, тригерами, зовнішніми ключами та збереженими процедурами. Він призначений для обробки різноманітних робочих навантажень, від окремих машин до сховищ даних або Інтернету. послуги з великою кількістю одночасних користувачів. Це база даних за замовчуванням для macOS Server, а також доступна для Windows, Linux, FreeBSD і OpenBSD.

Об'єктно-реляційна база даних – це система керування базами даних подібна до реляційної, але з об'єктно-орієнтованою моделлю бази: об'єкти, класи та наслідування підтримуються в схемі даних та мові запитів. На додачу, як і в реляційних БД, вона підтримує розширення моделі даних новими типами даних та методами.

Об'єктно-реляційна база даних знаходиться в проміжному положенні між реляційними базами даних і об'єктно-орієнтованими базами даних (об'єктними базами даних). В об'єктно-реляційних баз даних, підхід, по суті, як у реляційних баз даних: дані зберігаються в базі даних і маніпулюють всі разом із запитом мовою запитів; на іншому полюсі знаходиться OODBMSes, в якій база даних є ніби стійким об'єктом, сховищем для програмного забезпечення, написаного об'єктно-орієнтованою мовою програмування, з програмуванням API для зберігання та вилучення об'єктів, і мало або взагалі не має конкретної підтримки запитів.

3.3 Telegram Bot API і його реалізація

Для побудови UI інтерфейсу свого бота я буду використовувати Telegram. Він пропонує бек-енд розробникам вже готову реалізацію клієнтського додатку, тому можна буде сконцентруватись лише на бізнес логіці.

Боти — це сторонні програми, які працюють всередині Telegram. Користувачі можуть взаємодіяти з ботами, надсилаючи їм повідомлення, команди та вбудовані запити. Ви керуєте своїми ботами за допомогою HTTPS-запитів до API бота Telegram.

Telegram пропонує наступний функціонал своїх ботів:

- Отримуйте персоналізовані сповіщення та новини. Бот може діяти як розумна газета, надсилаючи вам відповідний вміст, щойно його публікують.
- Інтеграція з іншими сервісами. Бот може збагатити чати Telegram вмістом із зовнішніх сервісів.
- Приймати платежі від користувачів Telegram. Бот може пропонувати платні послуги або працювати як віртуальна вітрина.
- Створюйте власні інструменти. Бот може надавати вам сповіщення, прогноз погоди, переклади, форматування чи інші послуги.
- Бот для розмітки, бот для наклейок, бот для голосування, бот для лайків
- Створюйте одно- та багатокористувацькі ігри. Бот може запропонувати багатий досвід HTML5, від простих аркад і

головоломок до 3D-шутерів і стратегічних ігор у реальному часі.

- Створюйте соціальні послуги. Бот може з'єднувати людей, які шукають партнерів по розмові на основі спільних інтересів або близькості.
- Робіть практично будь-що інше. Крім посуду — боти жахливо миють посуд. =)

По суті, Telegram Vots — це спеціальні облікові записи, для налаштування яких не потрібен додатковий номер телефону. Користувачі можуть взаємодіяти з ботами двома способами:

- Надсилайте повідомлення та команди ботам, відкриваючи з ними чат або додаючи їх до груп.
- Надсилайте запити безпосередньо з поля введення, ввівши @username бота та запит. Це дозволяє надсилати вміст від вбудованих ботів безпосередньо в будь-який чат, групу чи канал.

Повідомлення, команди та запити, надіслані користувачами, передаються програмному забезпеченню, запущеному на ваших серверах. Наш сервер-посередник обробляє за вас усе шифрування та спілкування з Telegram API. Ви спілкуєтеся з цим сервером через простий HTTPS-інтерфейс, який пропонує спрощену версію API Telegram. Ми називаємо цей інтерфейс нашим API бота.

Для більшості мов програмування існують бібліотеки з готовою реалізацією Telegram Bot API. Java не є виключенням. Я буду використовувати бібліотеку під назвою Pengrad/Java Telegram Bot API.

3.4 Що таке технічний аналіз та як з ним працювати

Головною особливістю мого бота буде його вміння користуватися технічним аналізом.

Технічний аналіз — у економіці — прогнозування зміни ціни в майбутньому на основі аналізу зміни ціни у минулому. У його основі лежить аналіз часових рядів ціни — «чартів» (від англ. *chart*). Окрім цінових рядів, в технічному аналізі використовується інформація про об'єми торгів та інші статистичні дані. Найчастіше методи технічного аналізу використовуються для аналізу цін, що змінюються вільно, наприклад, на біржах.

У технічному аналізі розроблено безліч різноманітних інструментів та методів, але всі вони засновані на одному загальному припущенні — шляхом аналізу часових рядів за допомогою виділення трендів (на основі паттернів і технічних індикаторів) можливо спрогнозувати поведінку ціни в майбутньому.

Технічний аналіз і його традиційний опонент фундаментальний аналіз — основні школи аналізу цінних паперів.

Існують відмінності в методах технічного аналізу на Forex та на біржовому фондовому ринку. Наприклад, на валютному ринку операції укладаються між банками і об'єми операцій не публікуються, кожен банк може публікувати лише свої котирування, операції відбуваються цілодобово, включаючи вихідні дні. На біржах ціни і об'єми операцій публікують спеціальні комісії, торгівля ведеться в рамках часу торговельних сесій. Проте, загальні принципи технічного аналізу на всіх ринках однакові.

Для реалізації аналізу я буду використовувати бібліотеку Ta4j.

Ta4j — бібліотека Java з відкритим кодом для технічного аналізу. Вона надає основні компоненти для створення, оцінки та виконання торгових стратегій.

Особливості Ta4j:

- 100% чиста Java - працює на будь-якій платформі Java версії 8 або новішої
- Більше 130 технічних індикаторів (Aroon, ATR, ковзні середні, параболічний SAR, RSI тощо)
- Потужний механізм для створення власних торгових стратегій
- Утиліти для запуску та порівняння стратегій
- Мінімальна залежність від сторонніх розробників
- Проста інтеграція

3.5 Допоміжні бібліотеки використані для розробки

Project Lombok – це бібліотека Java, яка автоматично підключається до вашого редактора та створює інструменти, оживляючи вашу Java.

З ним більше не доведеться писати методи як `getter` або `equals`. З однією анотацією ваш клас має повнофункціональний конструктор та все необхідне для роботи з ним

Java WebSockets – нічого особливого, просто дозволяє реалізувати `web socket client` на стороні серверної часті Java.

Unirest Java – це легка клієнтська бібліотека HTTP від Mashape. Поряд з Java він також доступний для Node.js, .Net, Python, Ruby тощо.

Вбудований HTTP клієнт від Java є не дуже зручним, тому я буду використовувати дану альтернативу.

Emoji Java — це легка бібліотека, яка допомагає використовувати Emojis у програмах Java.

3.6 Про біржу Binance та Binance API

Працювати мій бот буде з однією із самих популярних криптовалютних бірж.

Binance — це біржа криптовалют, яка є найбільшою біржею у світі за щоденним обсягом торгівлі криптовалютами. Вона була заснована в 2017 році і зареєстрована на Кайманових островах.

Binance був заснований Чанпен Чжао, розробником, який раніше створював програмне забезпечення для високочастотної торгівлі. Спочатку Binance базувався в Китаї, але пізніше переніс свою штаб-квартиру з Китаю, після посилення регулювання криптовалют китайським урядом.

У 2021 році Міністерство юстиції та Податкова служба США розслідували Binance за звинуваченнями у відмиванні грошей та податкових правопорушеннях. Управління фінансового контролю Великобританії наказало Binance припинити всю регульовану діяльність у Сполученому Королівстві в червні 2021 року.

Binance пропонує для своїх користувачів відкрите та абсолютно безплатне API. Це одне з найкращих та найбільших API, з якими мені доводилось працювати.

Ось кілька розділів з їхньої документації:

- Wallet Endpoints

- Sub-Account Endpoints
- Market Data Endpoints
- Websocket Market Streams
- Spot Account/Trade
- Margin Account/Trade
- User Data Streams
- Savings Endpoints
- Staking Endpoints
- Mining Endpoints
- Futures
- Futures Algo Endpoints
- Portfolio Margin Endpoints
- BLVT Endpoints
- BSwap Endpoints
- Fiat Endpoints
- C2C Endpoints
- Crypto Loans Endpoints
- Pay Endpoints
- Convert Endpoints
- Rebate Endpoints
- NFT Endpoints
- Binance Code Endpoints
- Error Codes

Як ми бачимо тут є досить багато інформації, з якою було б цікаво працювати. В своєму боті я буду використовувати розділ Websocket Market Streams для моніторингу ринку в живому часі та Spot Account/Trade для здійснення торгових угод.

3.7 Розгортання бота в хмарному середовищі

Розгортання програмного забезпечення (англ. *Software deployment*) — це усі дії, що роблять програмну систему готовою до використання. Даний процес є частиною життєвого циклу програмного забезпечення.

Загалом процес розгортання складається з кількох взаємопов'язаних дій із можливими переходами між ними. Ця активність може відбуватися як з боку виробника, так і з боку споживача. Оскільки кожна програмна система є унікальною, то усі процеси та процедури під час розгортання важко передбачити. Тому, «розгортання» можна трактувати як загальний процес відповідно до певних вимог та характеристик. Розгортання може здійснюватися програмістом і в процесі розробки програмного забезпечення.

У нас інколи англomовний термін *software deployment* перекладають як «*впровадження*», оскільки згідно з давніми, радянськими стандартами, орієнтованими на планову економіку, впровадження мало бути кінцевим етапом будь-якого виробництва, що породило деякі неоднозначності у літературі. У вітчизняній літературі ці терміни часто вживаються або як взаємозамінні синоніми, або як окремі назви етапів. Ці два терміни дещо відрізняються за своєю суттю. *Впровадження* — вузький термін. Тому часто фірми, пов'язані з розробкою та встановленням програмного забезпечення, вказують, що вони здійснюють *впровадження* і *розгортання* програмного забезпечення.

Для розгортання свого бота я буду використовувати сервіс Heroku.

Heroku — хмарна PaaS-платформа, що підтримує ряд мов програмування. Компанією Heroku володіє Salesforce.com. Heroku, одна з перших хмарних

платформ, з'явилась в червні 2007 року і спочатку підтримувала тільки мову програмування Ruby, але зараз список підтримуваних мов також містить в собі Java, Node.js, Scala, Golang, Clojure, Python і PHP. На серверах Heroku використовуються операційні системи Debian або Ubuntu (яка також заснована на Debian).

Чому я обрав саме Heroku? По-перше, Heroku надає попередньо встановлену операційну систему та сервери, отже, усуває потребу в налаштуванні інфраструктури, а отже, навіть людина, яка новачок у хмарній платформі Heroku, може легко зупинитися на розгортанні, не турбуючись про проблеми з інфраструктурою управління.

По-друге, користувач може легко вибрати план підписки відповідно до власних потреб, а Heroku — це продукт, який оплачуєте те, що ви використовуєте, за іншим принципом. Отже, якщо вам раптом знадобиться велика кількість кешованих даних, просто додайте Redis до свого плану, і все готово.

Розділ 4. Реалізація торгового бота

4.1 Опис ключових етапів реалізації

Отже, настав час втілювати проект в життя. В цьому розділі я буду описувати ключові етапи розробки. Весь код можна переглянути натиснувши [сюди](#).

Перш за все потрібно створити Spring boot проект. Якщо ви користуєтесь IntelliJ Idea Ultimate edition це зробити буде дуже просто. Потрібно буде обрати тип Spring initializr і заповнити кілька інтуїтивно зрозумілих полей (рис. 4.1) Також можна скористатися сайтом <https://start.spring.io/> де після обрання всіх необхідних параметрів можна скачати zip архів з кістяком для майбутнього проекту.

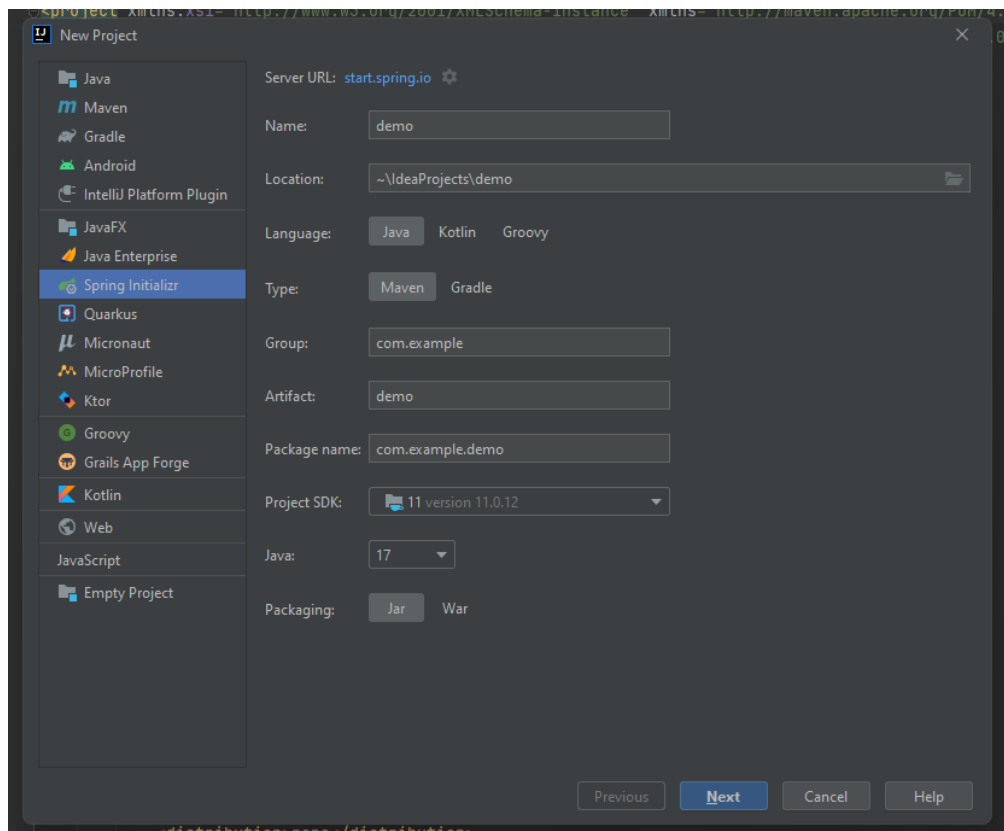
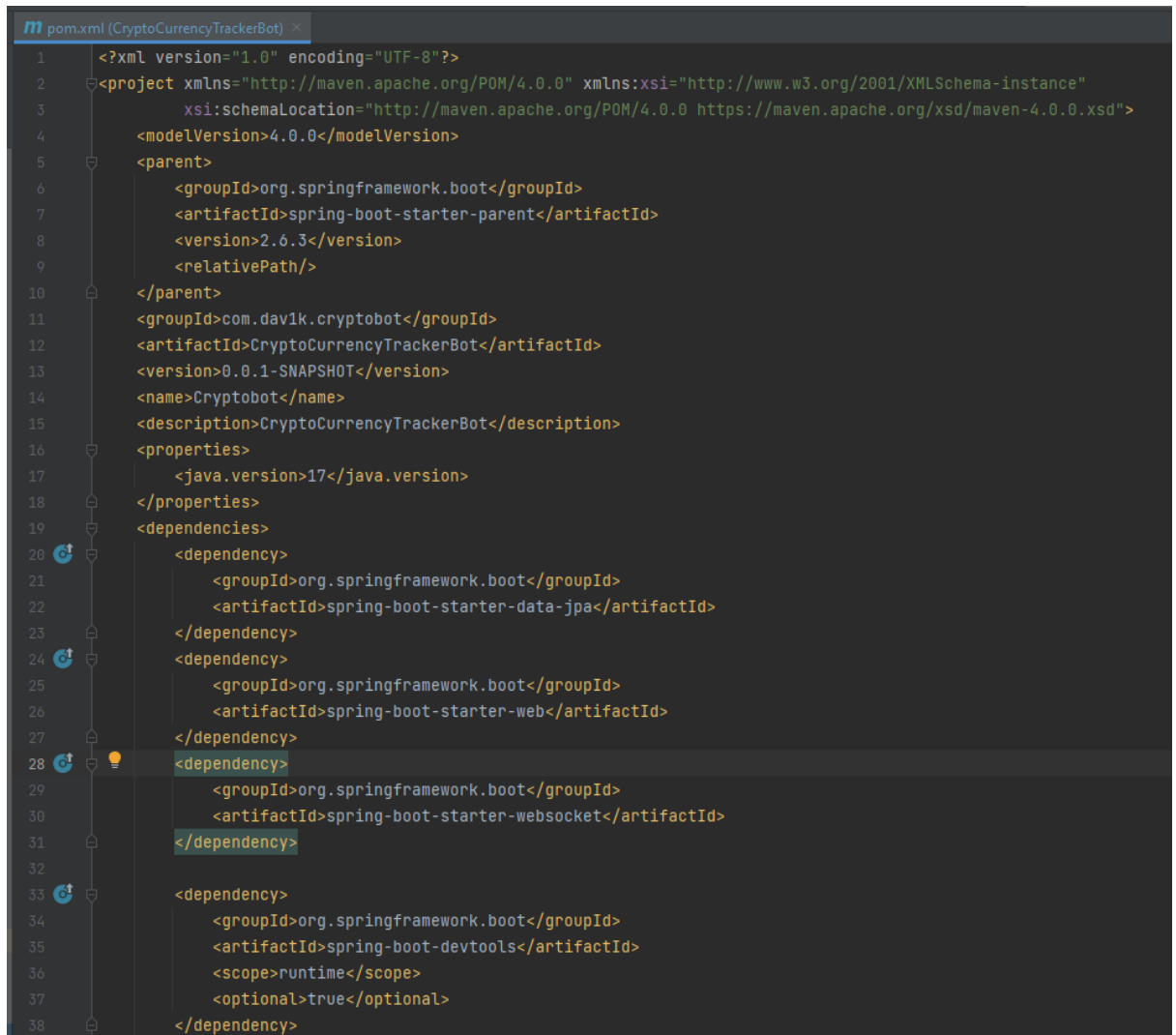


Рис. 4.1 - Створення Spring проекту

Наступним кроком потрібно підключити всі зовнішні бібліотеки, з якими нам доведеться працювати. Так як для збирання

проекту я обрав Maven, то всі залежності будем прописувати в файлі pom.xml (рис. 4.2)



```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
3     xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd">
4     <modelVersion>4.0.0</modelVersion>
5     <parent>
6         <groupId>org.springframework.boot</groupId>
7         <artifactId>spring-boot-starter-parent</artifactId>
8         <version>2.6.3</version>
9         <relativePath/>
10    </parent>
11    <groupId>com.dav1k.cryptobot</groupId>
12    <artifactId>CryptoCurrencyTrackerBot</artifactId>
13    <version>0.0.1-SNAPSHOT</version>
14    <name>Cryptobot</name>
15    <description>CryptoCurrencyTrackerBot</description>
16    <properties>
17        <java.version>17</java.version>
18    </properties>
19    <dependencies>
20        <dependency>
21            <groupId>org.springframework.boot</groupId>
22            <artifactId>spring-boot-starter-data-jpa</artifactId>
23        </dependency>
24        <dependency>
25            <groupId>org.springframework.boot</groupId>
26            <artifactId>spring-boot-starter-web</artifactId>
27        </dependency>
28        <dependency>
29            <groupId>org.springframework.boot</groupId>
30            <artifactId>spring-boot-starter-websocket</artifactId>
31        </dependency>
32
33        <dependency>
34            <groupId>org.springframework.boot</groupId>
35            <artifactId>spring-boot-devtools</artifactId>
36            <scope>runtime</scope>
37            <optional>true</optional>
38        </dependency>

```

Рис. 4.2 - Частковий вигляд файлу pom.xml

Наступним кроком буде створення Telegram бота як UI інтерфейсу для мого торгового бота. Токен я буду діставати з файлу application.properties. Це спеціальний конфігураційний файл, з якого Spring дістає певну інформацію, яку не слід містити в програмному коді. Далі потрібно створити самого бота та встановити update listener (рис. 4.3).

```
@Service
@Slf4j
@RequiredArgsConstructor
public class CryptoBot {

    @Value("${telegram.bot.token}")
    private String TELEGRAM_BOT_TOKEN;

    private static TelegramBot bot;

    private final UpdateHandler updateHandler;

    @PostConstruct
    public void init() {
        bot = new TelegramBot(TELEGRAM_BOT_TOKEN);
        bot.setUpdatesListener(updates -> {
            updates.forEach(updateHandler::handleUpdate);
            return UpdatesListener.CONFIRMED_UPDATES_ALL;
        });
        log.info("Bot started...");
    }
}
```

Рис. 4.3 - Опис об'єкта бот

Після цього потрібно обрати, що робити з кожним update. Update - це свого роду запит, який приходить до бота і його потрібно буде обробити, після чого дати якусь відповідь. Зверніть увагу що сам метод *init()* ніде зовні не викликається. Це все завдяки анотації *@PostConstruct*. Під час ініціалізації програми фреймворк Spring перед створенням біна (bean) запускає його *init()* метод. Bean — це об'єкт, який створюється, збирається та іншим чином керується контейнером Spring IoC (Inversion of Control). В іншому випадку bean є просто одним із багатьох об'єктів у вашій програмі.

Для обробки update створимо допоміжний клас UpdateHandler (рис. 4.4)

```

@Slf4j
@Service
@RequiredArgsConstructor
public class UpdateHandler {

    private final MessageHandler messageHandler;
    private final CallbackQueryHandler callbackQueryHandler;

    public void handleUpdate(Update update) {
        if (isMessage(update)) {
            messageHandler.handleMessage(update.message());
        } else if (isCallbackQuery(update)) {
            callbackQueryHandler.handleCallback(update.callbackQuery());
        } else {
            log.error("Unknown update type");
        }
    }
}

```

Рис. 4.4 - Update handler

Цей клас буде визначати сутність updat'a і перенаправляти його до наступного обробника. На даний час я обробляю лише два типи:

1. message - звичайне повідомлення від користувача.
2. callBackQuery - цей об'єкт представляє вхідний запит callBack який іде від кнопки на вбудованій клавіатурі. (продемонструю пізніше).

Реалізація цих класів вже значно більша тому, кому цікаво, пропоную звернутися до мого github репозиторію. (Посилання було на початку розділу).

Власне, як я і описував у вступі, у бота буде три основних функції. Реалізація першої – моніторинг ринку – доволі проста.

Я буду використовувати Binance API для отримання актуальних цін на криптовалюту. В першому варіанті я робив звичайні GET запити, але там був значний мінус. Через ліміти не можна було робити запити надто часто, а це означало що інформація, з якою працюватиме бот, буде не зовсім актуальною. Тому прийшлося шукати рішення як за допомогою Java зробити WebSocket клієнт. Це виявилось доволі складним завданням, тому

що на Java в пишуть сервери і в інтернеті майже не було інформації. Але все ж мені вдалось знайти певну бібліотеку, яка вирішувала мою проблему. Для отримання актуальних цін потрібно підписатись на сокет *all@aggTrade*.

Ось код класу, який відповідає за підключення до сокету:

```
@Slf4j
@Service
public class BinanceWebsocketClient extends WebSocketClient {
    private static final String SUCCESS_SOCKET_SUBSCRIPTION = "{\"result\":null,\"id\":1}";
    private static final Map<String, String> CURRENCIES = new HashMap<>();
    private final JsonToClassConverter jsonToClassConverter;
    private final BinanceService binanceService;
    public BinanceWebsocketClient(Uri binanceWebsocketUri,
        JsonToClassConverter jsonToClassConverter,
        BinanceService binanceService) {
        super(binanceWebsocketUri);
        this.jsonToClassConverter = jsonToClassConverter;
        this.binanceService = binanceService;
    }
    @PostConstruct
    public void init() {
        connect();
    }
    @Override
    public void onOpen(ServerHandshake serverHandshake) {
        send(String.format("{"
            {
                "method": "SUBSCRIBE",
                "params": %s,
                "id": 1
            }", preparePairsForSubscription(binanceService.getUsdtPairs())));
        log.info("WS opened. Subscribed to all@aggTrade");
    }
    @SneakyThrows
    @Override
    public void onMessage(String message) {
        if (!SUCCESS_SOCKET_SUBSCRIPTION.equals(message)) {
            if (CURRENCIES.isEmpty()) log.info("Tracking started");
            BinanceWebsocketResponse binanceWebsocketResponse =
                jsonToClassConverter.convertFromJson(message, BinanceWebsocketResponse.class);
            CURRENCIES.put(binanceWebsocketResponse.symbol.replace("USDT", ""),
                binanceWebsocketResponse.price);
        }
    }
    @Override
    public void onClose(int code, String reason, boolean remote) {
        log.error("Connection closed by " + (remote ? "remote peer" : "us") + " Code: " + code + "
Reason: " + reason);
    }
}
```


І після конвертації назва валюти та її ціна поміщаються до спеціальної мапи CURRENCIES. Саме вона і буде містити актуальну інформацію про кожну з валют всередині моєї програми.

Наступним на черзі розробки буде сервіс (рис. 4.5), який безпосередньо буде займатися моніторингом.

```
@Slf4j
@Service
@RequiredArgsConstructor
public class TrackingService {

    private final UserTrackingCurrencyService userTrackingCurrencyService;
    private final BinanceWebsocketClient binanceWebsocketClient;

    @Scheduled(fixedRate = 5000)
    public void checkCurrencies() {
        List<UserTrackingCurrency> userTrackingCurrencies = userTrackingCurrencyService.getAllTrackingCurrencies();
        Map<String, String> currencies = binanceWebsocketClient.getCurrenciesWithPrice();
        userTrackingCurrencies.forEach(trackingCurrency -> {
            if (currencies.get(trackingCurrency.getName()) != null) {
                Float livePrice = parseFloat(currencies.get(trackingCurrency.getName()));
                if (trackingCurrency.wasTracked()) {
                    if (isMarginCrossed(trackingCurrency, livePrice)) {
                        notifyAboutMarginCrossing(trackingCurrency, livePrice);
                    }
                    if (isTargetReached(trackingCurrency, livePrice) && !trackingCurrency.isTargetReached()) {
                        notifyAboutTargetReaching(trackingCurrency);
                    }
                } else {
                    setDefaultCurrencyParams(trackingCurrency, livePrice);
                }
            }
        });
    }
}
```

Рис. 4.5 - Tracking service

Тут ви можете помітити клас, про який я раніше не згадував - UserTrackingCurrency(рис. 4.6). Він відповідає за збереження інформацію про відстежувану валюту кожного з користувачів.

```

@Entity
@Data
@AllArgsConstructor
@NoArgsConstructor
public class UserTrackingCurrency {

    @Id
    @GeneratedValue(strategy = IDENTITY)
    private Long id;

    private String name;

    private Float lastPrice;

    private boolean tracking;

    private Float margin;

    private Float marginPercentage = 0.1f;

    private Float maxPrice;

    private Float targetPrice;
    private boolean isTargetReached = false;

    private Long chatId;
    private Long userId;

    public UserTrackingCurrency(String name) { this.name = name; }

    public boolean wasTracked() { return margin != null && lastPrice != null; }

    public boolean hasTargetPrice() { return targetPrice != null; }
}

```

Рис. 4.6 - UserTrackingCurrency

Особливу увагу тут слід звернути на анотацію `@Entity`. Вона означає, що цей об'єкт буде зберігатись в базі даних.

Повернемось до сервісу моніторингу. Отже, він звертається до бази за допомогою ще одного проміжного сервісу. (Це такий патерн розробки, коли для кожного об'єкту, який зберігається в бд, створюється клас репозиторій, який безпосередньо пов'язується з бд, та клас сервіс, який буде керувати всім, що

пов'язано з об'єктом). Після того, як ми отримали абсолютно всі валюти, які відстежують користувачі, сервіс дістає актуальні ціни і порівнює їх. Сама логіка для кожного з користувачів може відрізнятися, тому що є можливість виставити числове або відсоткове значення. Наприклад, повідомляти коли ціна піросла на 10\$ від останнього повідомлення або ж на 5%.

Тепер приступимо до більш складної реалізації - моніторинг та аналіз ринку.

Перш за все підключаємо бібліотеку ta4j. Після цього потрібно реалізувати підписку на інший сокет, який повертає більше інформації: ціну закриття, ціну відкриття, максимальну та мінімальну ціну, об'єм торгової валюти. Ці дані залежать від часової шкали.



Рис. 4.6 - Хвилинний графік BTC/USDT

Ось для прикладу рис. 4.6. Це свічковий графік пари BTC/USDT. Кожна свічка (зелена чи червона) якраз таки містить в собі всю необхідну нам інформацію. Для отримання такої інформації потрібно підписатись на сокет `@kline_1m`, де значення після `_` це часова шкала, за якою ми хочемо отримувати інформацію. Реалізація аналогічна як з отриманням актуальної ціни, тому її описувати не буду. Тепер розглянемо як аналізувати отримані дані.

На початку створюємо об'єкт часового ряду:

```
BarSeries series = new BaseBarSeriesBuilder().withName("BTCUSDT").build();
```

Після створення часового ряду ми можемо додати дані та обсяг до ряду:

```
series.addBar(dateTime, openPrice, hightPrice, lowPrice, closePrice, volume)
```

де `dateTime` - дата з часом, за яку беруться дані; `openPrice` - ціна, з якою відкрилася свічка; `hightPrice` - максимальна ціна, якої досягав графік за обраний нами період часу, аналогічно `lowPrice`; `closePrice` - ціна, з якою свічка закрилася, тобто проміжок часу вийшов і почала малюватися наступна; `volume` - обсяг валюти яка була в обороті за період часу (наприклад за хвилину було здійснено операцій на загальну суму 3 btc).

Далі потрібно використовувати певні індикатори з технічного аналізу. Ми можемо розрахувати індикатор по цьому ряду стовпчиків, щоб спрогнозувати напрямок цін через вивчення минулих ринкових даних.

Отримання ціни закриття:

```
Num firstClosePrice = series.getBar(0).getClosePrice();
```

Або в межах показника:

```
ClosePriceIndicator closePrice = new ClosePriceIndicator(series);
```

```
closePrice.getValue(0) дорівнює firstClosePrice
```

Отримання простої ковзаючої середньої (SMA) ціни закриття за останні 5 тиків:

```
SMAIndicator shortSma = new SMAIndicator(closePrice, 5);
```

Отримання більш тривалого SMA (наприклад, понад 30 останніх тиків):

```
SMAIndicator longSma = new SMAIndicator(closePrice, 30);
```

Далі ми повинні побудувати нашу торгову стратегію. Вона складається з двох правил торгівлі: одне для входу, інше для виходу.

Правило купівлі. Ми хочемо купити, коли 5 SMA перетне зверху 30 SMA або ціна опуститься нижче вказаної:

```
Rule buyingRule = new CrossedUpIndicatorRule(shortSma, longSma)  
    .or(new CrossedDownIndicatorRule(closePrice, 20000d));
```

Правило продажу. Ми хочемо продати, коли 5 SMA перетне знизу 30 SMA або якщо ціна втрачає більше ніж на 3% чи перевищує більше чим на 2%

```
Rule sellingRule = new CrossedDownIndicatorRule(shortSma, longSma)  
    .or(new StopLossRule(closePrice, 3.0))  
    .or(new StopGainRule(closePrice, 2.0));
```

Далі користуючись цими правилами, якщо вони будуть здійснюватись, бот буде повідомляти користувача про те, що зараз є підходящий момент для того, щоб відкрити угоду на покупку чи продажу певної валюти. Вся детальна реалізація знаходиться в моєму github репозиторії (посилання на початку розділу). Реалізація надсилання повідомлень ботом є надто проста, тому не бачу сенсу її описувати. Аналогічно з відкриттям торгових угод, потрібно лише зареєструватись на біржі, отримати api token і з його допомогою робити POST запити згідно документації. Власне кажучи я використовував самі примітивні правила, через брак певних економічних знань, але я впевнений, що справжній економічний спеціаліст, може створити максимально безпрограшну торгову стратегію.

4.2 Демонстрація роботи бота

Отже, спочатку кожен користувач повинен підключитись до бота через телеграм за посиланням @crypto_trkr_bot.

Далі за допомогою команди /menu потрапити до меню налаштувань (рис 4.7). Наразі для користувачів доступний лише функціонал моніторингу. Це пов'язано з тим, що для торгівлі потрібно зберігати арі ключі кожного клієнта в незашифрованому вигляді, а також нести відповідальність за чужі кошти, на що я не можу наважитись. Тому кожен хто хоче безпосередньо, щоб бот створював за нього торгові угоди, може скористатись кодом, на свій страх і ризик.

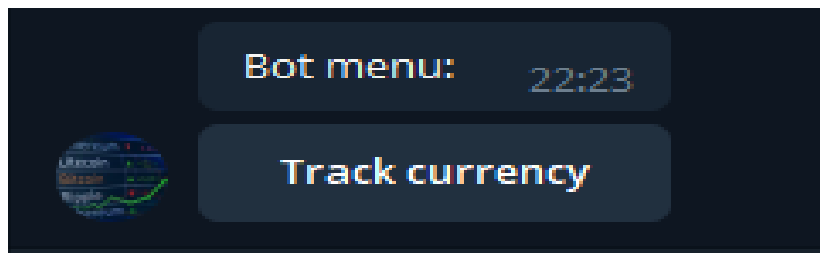


Рис. 4.7 - Меню

Натиснувши на клавiшу Track currency користувач попадає на список всіх доступних на біржі Binance криптовалют в парі до USDT (рис. 4.8)

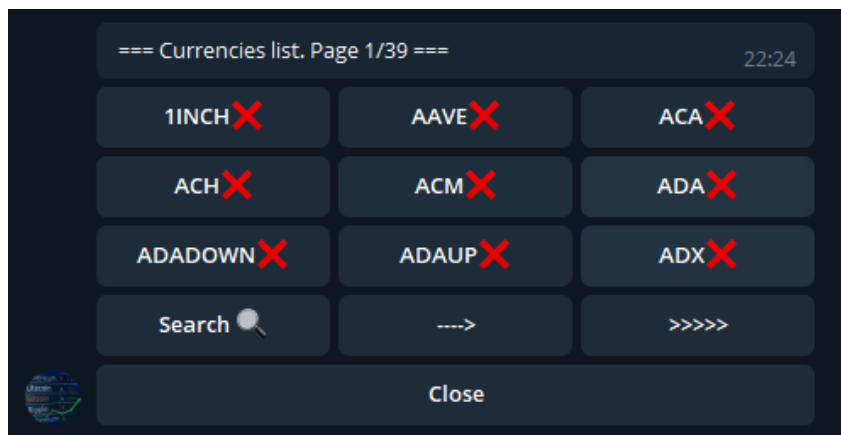


Рис. 4.8 - Список доступних валют

Далі користувач може скористатись навігацією по списку або пошуком, для того щоб знайти бажану криптовалюту. Після цього потрібно натиснути на неї і вона буде відмічена зеленим на початку списку, що буде означати, що моніторинг для даної валюти увімкнений (рис. 4.9)

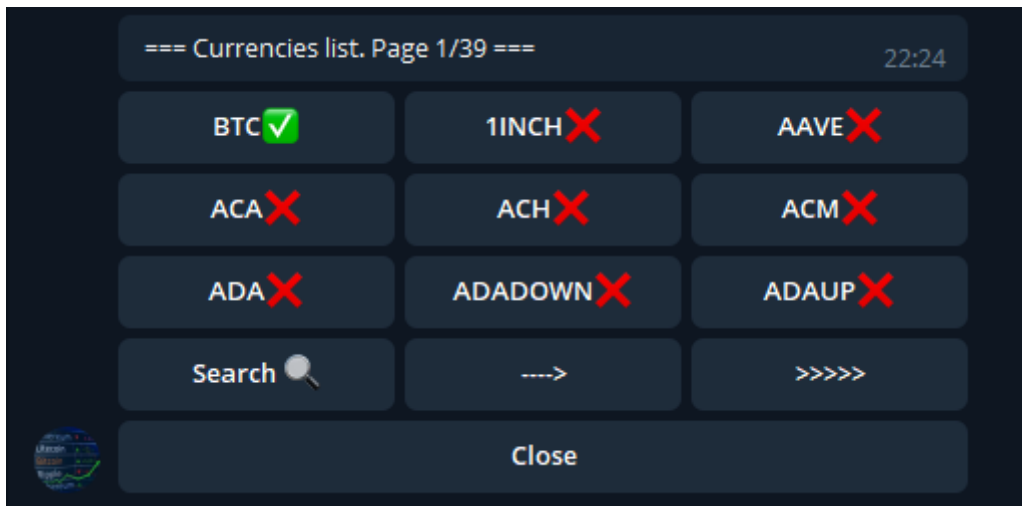


Рис. 4.9 - Список доступних валют включаючи обрані для відстеження

На цьому основні дії користувача закінчуються і все, що залишилось робити - просто очікувати. Згодом бот почне сповіщати про ті чи інші події на ринку, пов'язані з валютами які кожен обирає для себе особисто. На (рис 4.10) видно що за 27 та 30 травня були певні повідомлення від бота. Перше повідомлення говорить, що ціна на NEAR різко впала. А друге, вже значно цікавіше. Воно повідомляє що зараз чудова можливість купити SOL за ціною 41.3. І вже 30 травня ціна на SOL підростає до 45.88, про що бот нас звісно ж повідомляє.

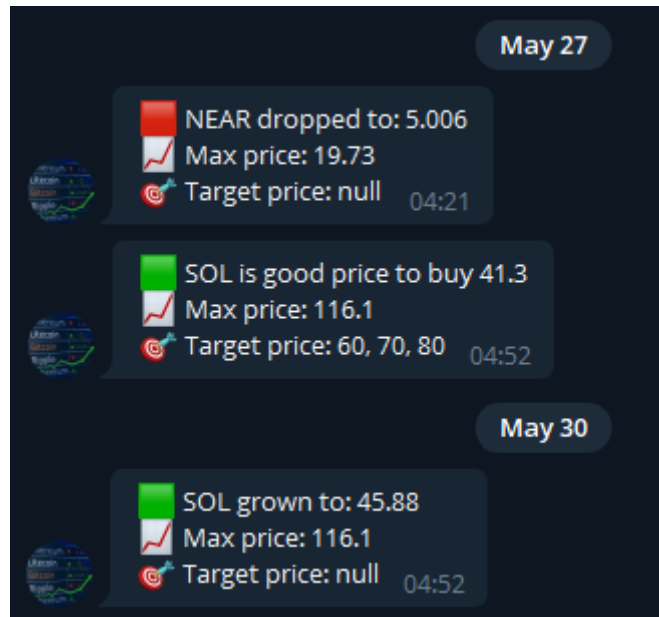


Рис. 4.10 - Приклад повідомлень

Станом на 22:50 30.05 ціна SOL становить 45.83\$, а це більше ніж на 10% перевищує ту ціну – 41.3\$, за яку бот радив купити SOL три дні тому.



Рис. 4.11 - Графік SOL/USDT 26.05-30.05

ВИСНОВКИ

Під час виконання роботи було досліджено історію та роботу технології blockchain. Оглянуто сфери застосування та порівняння з вже існуючими технологіями зберігання інформації. Було розглянуто сутність та характеристики криптовалют, які побудовані на технології blockchain. Оглянуто основні проблеми використання криптовалют. Було проведено порівняння фіатних та крипто валют. Розглянуто перспективи розвитку криптовалютного ринку. Був розроблений та описаний торговий бот, наділений трьома основними функціями: моніторингу, аналізу та торгівлі на криптовалютному ринку за допомогою біржі Binance. Отже, використання технології blockchain є актуальним та популярним. Блокчейн з кожним днем продовжує розвиватися, тому я вірю що технологія здобуде ще більшу кількість варіантів застосування в повсякденному житті. І як я вже продемонстрував в кінці попереднього розділу, бот працює так як і задумувалось. Не завжди ідеально, але далі все залежить від економічних знань і правильно підібраної торгової стратегії.

Розроблений бот дає можливість трейдерам економити надзвичайно велику кількість часу. А як ми всі знаємо час – найцінніший ресурс, яким володіє кожна людина. Тому краще витратити його не на сварки, маячню чи бізнес, а на сім'ю, близьких, друзів та здоров'я.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Blockchain [Електронний ресурс] – Режим доступу до ресурсу:
<https://en.wikipedia.org/wiki/Blockchain>
2. Дерево Меркла [Електронний ресурс] – Режим доступу до ресурсу:
https://en.wikipedia.org/wiki/Merkle_tree
3. Blockchain definition [Електронний ресурс] – Режим доступу до ресурсу: <https://www.investopedia.com/terms/b/blockchain.asp>
4. Криптовалюта [Електронний ресурс] – Режим доступу до ресурсу:
<https://en.wikipedia.org/wiki/Cryptocurrency>
5. Легалізація криптовалюти в Україні [Електронний ресурс] – Режим доступу до ресурсу: <https://minfin.com.ua/ua/2022/03/16/82176211/>
6. Will be crypto replace fiat? [Електронний ресурс] – Режим доступу до ресурсу: <https://builtin.com/blockchain/fiat-crypto>
7. How is cryptocurrency similar to fiat [Електронний ресурс] – Режим доступу до ресурсу:
<https://www.ndtv.com/business/cryptocurrency-vs-fiat-money-similarities-and-differences-2528740>
8. What's the difference between crypto and fiat money [Електронний ресурс] – Режим доступу до ресурсу:
<https://www.bitpanda.com/academy/en/lessons/whats-the-difference-between-a-cryptocurrency-like-bitcoin-and-fiat-money/>
9. The future of cryptocurrency [Електронний ресурс] – Режим доступу до ресурсу:
<https://time.com/nextadvisor/investing/cryptocurrency/future-of-cryptocurrency/>
10. Cryptocurrency market size [Електронний ресурс] – Режим доступу до ресурсу:

<https://www.fortunebusinessinsights.com/industry-reports/cryptocurrency-market-100149>

11. Java [Электронный ресурс] – Режим доступа до ресурсу:
<https://www.java.com/>
12. ORM [Электронный ресурс] – Режим доступа до ресурсу:
https://en.wikipedia.org/wiki/Object%E2%80%93relational_database
13. Spring framework [Электронный ресурс] – Режим доступа до ресурсу: <https://spring.io/>
14. Telegram Bot API [Электронный ресурс] – Режим доступа до ресурсу: <https://core.telegram.org/bots/api>
15. Pengrad telegram bot for java [Электронный ресурс] – Режим доступа до ресурсу: <https://github.com/pengrad/java-telegram-bot-api>
16. Та4j [Электронный ресурс] – Режим доступа до ресурсу:
<https://github.com/ta4j/ta4j>
17. Technical analysis [Электронный ресурс] – Режим доступа до ресурсу: https://en.wikipedia.org/wiki/Technical_analysis
18. Spring boot DevTools [Электронный ресурс] – Режим доступа до ресурсу: <https://www.javatpoint.com/spring-boot-devtools>
19. Project Lombok [Электронный ресурс] – Режим доступа до ресурсу:
<https://projectlombok.org/>
20. Binance [Электронный ресурс] – Режим доступа до ресурсу:
<https://www.binance.com/en>
21. Binance API [Электронный ресурс] – Режим доступа до ресурсу:
<https://binance-docs.github.io/apidocs/#change-log>
22. Heroku [Электронный ресурс] – Режим доступа до ресурсу:
<https://dashboard.heroku.com/>