

3. Структури даних є важливою частиною ефективної роботи блокчейн-технологій, оскільки правильна організація даних впливає на швидкість і надійність системи. Однією з основних структур є дерево Меркля (Merkle Tree), яке використовується для ефективної перевірки транзакцій у блоці. Завдяки цій структурі зменшується обсяг інформації, яку необхідно зберігати, що забезпечує високу швидкість перевірки. Дерево Меркля дає змогу перевіряти транзакції з мінімальними витратами ресурсів завдяки зменшенню кількості необхідних для перевірки даних. Також важливими є зв'язаний список (Linked List), який можна використовувати для представлення блоків у блокчейн-системі, де кожен блок містить хеш попереднього блоку, що утворює ланцюг [3].

4. Алгоритми пошуку та сортування сприяють ефективному управлінню транзакціями і доступу до даних. Наприклад, сортування транзакцій за розміром комісії дає змогу майнерам обробляти транзакції з найвищим пріоритетом, що оптимізує використання мережевих ресурсів. Для пошуку та доступу до даних використовуються структури, як-от хеш-таблиці, які забезпечують швидкий доступ до таких даних, наприклад, транзакції або баланси рахунків [4].

Алгоритми є основою ефективного функціонування блокчейн-технологій, забезпечуючи безпеку, децентралізацію та прозорість мереж. Основними типами алгоритмів є алгоритми консенсусу (PoW, PoS, PBFT тощо), хешування, а також структури даних, як-от дерево Меркля, що дають змогу ефективно перевіряти транзакції та зберігати цілісність даних. Правильне розуміння і застосування цих алгоритмів критичне для забезпечення стабільності та безпеки блокчейн-систем.

Список використаних джерел

1. Exbase. Алгоритм консенсусу. URL: <https://exbase.io/uk/wiki/algorithm-konsensusu> (дата звернення 02.12.2024).
2. Plisio. Як працює хешування в блокчейні? 2024. URL: <https://plisio.net/uk/blog/how-does-hashing-in-blockchain-work> (дата звернення 02.12.2024).
3. Merkle Tree: A Fundamental Component of Blockchains / H. Liu, X. Luo, H. Liu, X. Xia. 2021. URL: <https://ieeexplore.ieee.org/document/9588047> (дата звернення 02.12.2024).

УДК 004.414.23:641.5

*Кузьміна М. О., здобувачка вищої освіти,
Якубич К. О., асистент кафедри
інформаційних технологій*

РОЗРОБКА ПРОГРАМИ ДЛЯ КУХАРІВ НА PYTHON

Донецький національний університет імені Василя Стуса, м. Вінниця

Вступ. Сучасна кухня все частіше стає місцем впровадження новітніх технологій. Програми для кухарів дають змогу зручно управляти рецептами, формувати списки продуктів, планувати раціон та оптимізувати час. Python, як одна з найпопулярніших мов програмування, пропонує широкий спектр інструментів для створення інтуїтивних та ефективних рішень у цій сфері.

Завдяки простоті синтаксису та багатим бібліотекам ця мова стає вибором номер один для розробників, які прагнуть створювати корисні додатки [1].

Метою цієї роботи є створення програми для кухарів, яка не лише автоматизує рутинні завдання, але й допомагає професіоналам та аматорам готувати з легкістю та задоволенням.

Актуальність цієї теми зумовлена зростанням попиту на програми, які допомагають вирішувати щоденні завдання, як-от розрахунок харчової цінності страв, підбір рецептів відповідно до наявних продуктів та генерація списків покупок. До того ж цифрові інструменти для кухарів важливі для впровадження принципів здорового харчування, оскільки вони дають змогу враховувати калорійність та інші параметри.

Функції програми. Програма для кухарів на Python розроблена так, щоб виконувати широкий спектр завдань.

Програма дає можливість зручно управляти базою рецептів. Користувачі можуть додавати нові рецепти або редагувати вже наявні, зберігаючи детальну інформацію про інгредієнти, етапи приготування та час, необхідний для готування.

Особливістю програми є вбудований калькулятор калорійності та харчової цінності. Він автоматично розраховує калорії, а також кількість білків, жирів і вуглеводів у кожній страві на основі її інгредієнтів. Користувач може масштабувати рецепти залежно від бажаної кількості порцій, і програма автоматично адаптує всі обчислення.

Для полегшення планування покупок програма може створювати списки продуктів на основі вибраних рецептів. Ці списки можна експортувати у текстовий формат або в додатки для покупок, що значно спрощує організацію приготування страв.

Це універсальне рішення для професійних кухарів і кулінарних ентузіастів, яке значно полегшує процес приготування їжі, планування раціону та управління запасами продуктів.

У лістингу 1 наведений приклад коду, який розраховує калорійність страви.

Лістинг 1

```
import pandas as pd
# Дані про інгредієнти
data = {
    'ingredient': ['Молоко', 'Цукор', 'Борошно'],
    'calories': [42, 387, 364],
    'weight': [100, 100, 100]
}
# Функція для розрахунку калорійності
def calculate_calories(recipe):
    total_calories = 0
    for item in recipe:
        for index, row in data.iterrows():
            if row['ingredient'] == item['name']:
                calories_per_unit = row['calories'] / row['weight']
                total_calories += calories_per_unit * item['quantity']
    return total_calories
# Приклад страви
```

```
recipe = [{'name': 'Молоко', 'quantity': 200}, {'name': 'Цукор', 'quantity': 50}]
print(f"Калорійність страви: {calculate_calories(recipe)} ккал")
```

Результат програми: *Калорійність страви: 277.5 ккал*

Висновки. Розробка програми для кухарів на Python демонструє значний потенціал для автоматизації кулінарних процесів. Використання бібліотек *pandas*, *matplotlib* та інших дає змогу створити функціональний інструмент, що задовольнить потреби як професійних кухарів, так і любителів. Програма зменшує час, витрачений на планування страв, сприяє дотриманню здорового харчування і полегшує управління ресурсами на кухні [4].

Список використаних джерел

1. Lutz M. Learning Python. O'Reilly Media. 2013. 1643 p.
2. VanderPlas J. Python Data Science Handbook. O'Reilly Media, 2016. 548 p.
3. Taibi D., Lenarduzzi V., Pahl C. Processes, Motivations, and Issues for Migrating to Microservices Architectures. *IEEE Software*. 2017. P. 22–32.
4. Документація Python. URL: <https://docs.python.org>. (дата звернення: 01.12.2024).

УДК 004.415:004.432.4

*Лик В. В., здобувач вищої освіти,
Сеник І. О., асистент кафедри
інформаційних технологій*

АРХІТЕКТУРА МІКРОСЕРВІСІВ ДЛЯ МАСШТАБНИХ КЛІЄНТ-СЕРВЕРНИХ СИСТЕМ

Донецький національний університет імені Василя Стуса, м. Вінниця

В епоху цифрової трансформації компанії потребують архітектурних рішень, які забезпечують швидке реагування на зміни, гнучкість у розвитку та високу масштабованість. Мікросервісна архітектура стає все популярнішою через можливість адаптації до змінного навантаження та здатність задовольняти вимоги високонавантажених систем.

Останнім часом мікросервіси почали широко застосовуватися у великих компаніях (Netflix, Amazon, Uber) для розробки високонадійних, продуктивних систем.

Порівняння з традиційним монолітним підходом, де додаток розробляється як єдине ціле, ускладнює гнучкість та масштабування.

Мета роботи – проаналізувати основні переваги та виклики впровадження мікросервісної архітектури у клієнт-серверні системи, виявити умови, за яких вона є найефективнішою;

- 1) дослідити ключові принципи мікро сервісної архітектури;
- 2) визначити, як мікро сервіси можуть забезпечити масштабованість та продуктивність;
- 3) розглянути виклики, з якими стикаються компанії, що впроваджують мікросервіси [1].