

РОЗРОБКА КОНСОЛЬНОГО ЗАСТОСУНКУ СИСТЕМИ СКЛАДСЬКОГО ОБЛІКУ ІЗ ЗАСТОСУВАННЯМ ПРИНЦИПІВ ОБ'ЄКТНО-ОРІЄНТОВАНОГО ПРОГРАМУВАННЯ

Вінницький національний технічний університет

Анотація

У роботі представлено процес розробки консольного застосунку для автоматизації складського обліку (Inventory Manager). Програмне забезпечення побудоване на основі принципів об'єктно-орієнтованого програмування (ООП) з використанням мови Python. Застосунок забезпечує повний цикл маніпуляції даними (пошук, лінійне сортування, безпечне редагування через контекстний менеджер with, фільтрацію), відмовостійкість, автоматичний розрахунок логістичних витрат та реалізує трирівневу систему персистентності (Pickle, JSON, CSV). Використання консольного інтерфейсу дозволило мінімізувати споживання системних ресурсів та забезпечити високу швидкодію.

Ключові слова: об'єктно-орієнтоване програмування, Python, система складського обліку, консольний застосунок, персистентність даних, відмовостійкість.

Abstract

This paper presents the development process of a console application for warehouse inventory automation (Inventory Manager). The software is built based on the principles of object-oriented programming (OOP) using the Python language. The application provides a full cycle of data manipulation (search, linear sorting, safe editing via the with context manager; filtering), fault tolerance, automatic calculation of logistics costs, and implements a three-level persistence system (Pickle, JSON, CSV). The use of a console interface allowed to minimize the consumption of system resources and ensure high performance.

Keywords: object-oriented programming, Python, inventory management system, console application, data persistence, fault tolerance.

Вступ

У сучасних умовах глобалізації економіки, стрімкого розвитку електронної комерції та інтенсифікації логістичних процесів ефективно управління складськими запасами стає одним із визначальних факторів конкурентоспроможності будь-якого підприємства. Склад є складним динамічним вузлом ланцюга постачання, де щоденно здійснюються сотні операцій із приймання, маркування, сортування, обліку та відвантаження матеріальних цінностей. Традиційні паперові чи напівавтоматизовані методи ведення обліку вже не здатні забезпечити необхідну точність і швидкість обробки інформації, що призводить до виникнення фінансових втрат, пересортиці товарів та порушення термінів логістичних операцій.

На сучасному ринку програмного забезпечення існує велика кількість промислових систем управління складом (Warehouse Management Systems, WMS). Проте більшість із цих рішень орієнтовані на транснаціональні корпорації, мають надлишковий функціонал, вимагають високих витрат на ліцензування, складного адміністрування та потужних апаратних ресурсів. Крім того, важкі графічні інтерфейси (GUI) таких програм часто уповільнюють рутинну роботу комерційних та операторів в умовах реального складу, де ключове значення має швидкість безпомилкового введення даних «з клавіатури». Для потреб малих, середніх або локальних автономних складських комплексів актуальною науково-практичною задачею є розробка легких, швидкодійних консольних застосунків (CLI), які здатні працювати повністю офлайн, споживають мінімум системних ресурсів і мають високий рівень відмовостійкості.

Найбільш ефективним інженерним підходом до проектування архітектури таких інформаційних систем є об'єктно-орієнтоване програмування (ООП). Парадигма ООП надає можливість виконати точне декомпозиційне моделювання предметної області, представивши кожен реальну складську сутність (базовий інвентар, швидкопсувні продукти харчування, електроніку чи безпосередньо зони

зберігання) у вигляді незалежних програмних об'єктів. Завдяки фундаментальним принципам інкапсуляції, наслідування та поліморфізму стає можливим створення чистого, модульного коду з ізольованою логікою валідації даних та можливістю динамічного розрахунку специфічних параметрів (наприклад, вартості утримання різних категорій товарів) без дублювання алгоритмів.

Для реалізації відмовостійкого складського ядра ідеальним інструментом виступає мова програмування Python. Її розвинена екосистема, вбудовані модулі для обробки структурованих даних (JSON, CSV), механізми глибокої бінарної серіалізації (Pickle) та лаконічний синтаксис дозволяють швидко розробляти надійні кросплатформні системи. Створення консольного застосунку «Inventory Manager» на базі ООП мовою Python дозволяє поєднати високу обчислювальну якість логістичних розрахунків із гнучкими механізмами довготривалого збереження стану системи, забезпечуючи надійний захист від помилок ручного введення даних (Fault Tolerance).

Результати дослідження

У процесі розробки системи складського обліку «Inventory Manager» було проведено проектування об'єктної моделі предметної області з використанням уніфікованої мови моделювання (UML). Спроектвана архітектура чітко розмежовує бізнес-логіку системи та користувацький інтерфейс, що відповідає принципу єдиної відповідальності (SRP).

Основні можливості користувача (оператора складу) описані у вигляді сценаріїв взаємодії на діаграмі варіантів використання (рис. 1). Головний сценарій передбачає перегляд поточного інвентарю та додавання нових товарів із захистом від некоректного введення даних. Важливим елементом архітектури є реалізація функцій пошуку, лінійного сортування товарів за ціною, а також безпечного редагування об'єктів із використанням магічних методів та контекстного менеджера with. Окрім цього, впроваджено алгоритм фільтрації інвентарю специфічних зон зберігання. Окрему увагу приділено функціям збереження стану бази даних та експорту фінансових звітів.

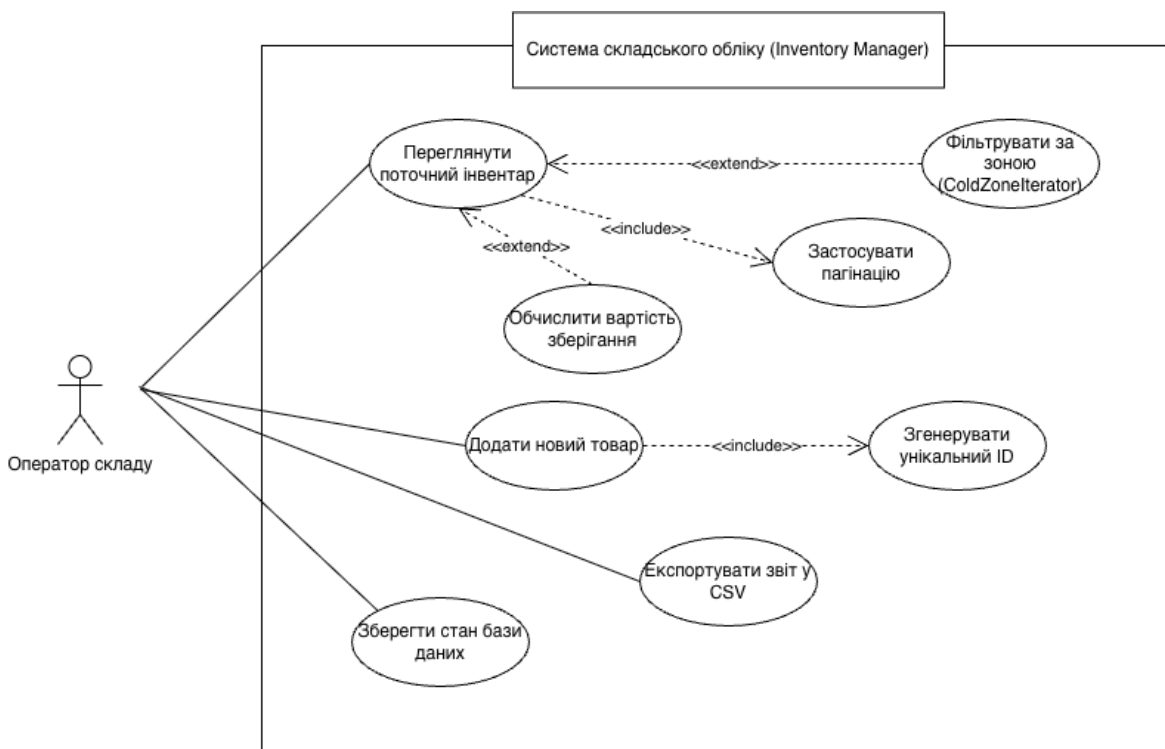


Рисунок 1 - Діаграма варіантів використання системи складського обліку

Програмна реалізація бізнес-логіки виконана мовою Python. В основу архітектури покладено базовий клас Item, який інкапсулює ключові характеристики товару. Для забезпечення суворої типізації зон зберігання використано перелічуваний тип Enum. Механізм наслідування дозволив утворити спеціалізовані класи PerishableItem (для швидкопсувних продуктів, реалізовано через dataclasses) та Electronics (для техніки). Завдяки поліморфізму кожен нащадок перевизначає базовий метод розрахунку

вартості зберігання з урахуванням власних особливостей (націнка за температурний режим або страхування). Структуру класів системи наведено на рисунку 2.

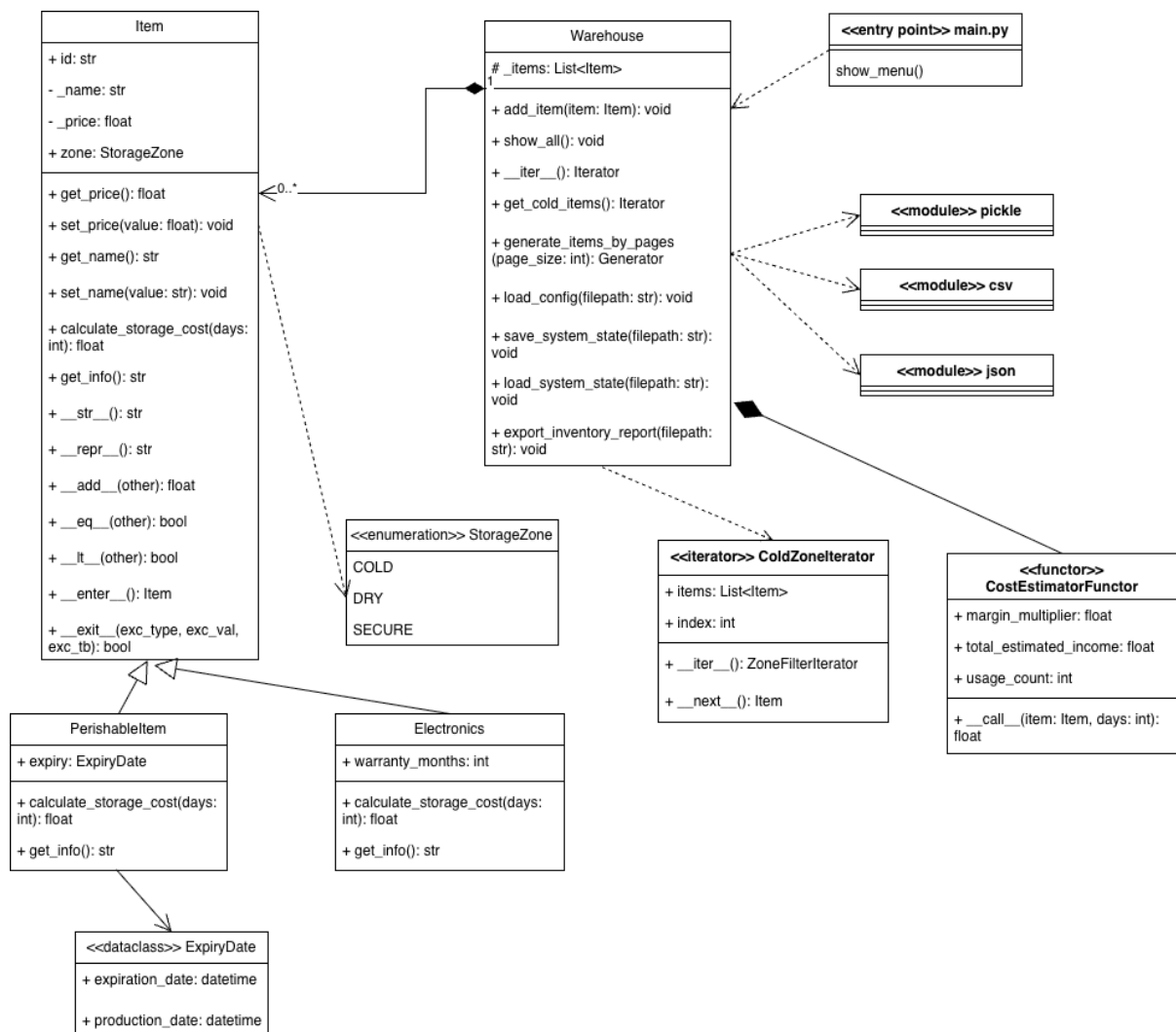


Рисунок 2 - UML-діаграма класів розробленого застосунку

Для оптимізації роботи з оперативною пам'яттю реалізовано кастомний ітератор, який забезпечує «лінивий» обхід колекції товарів під час фільтрації. Фінансова логіка відокремлена від моделей даних за допомогою патерну «Функтор» (CostEstimatorFuncnor).

Особливістю розробленого застосунку є трирівнева підсистема персистентності:

1. Бінарна серіалізація (модуль pickle) - для повного збереження (snapshot) та відновлення ієрархії об'єктів між сесіями (файл system.pkl).
2. Управління конфігураціями (JSON) - для гнучкого налаштування параметрів складу (файл config.json).
3. Експорт аналітики (CSV) - для генерації плоских табличних звітів, сумісних із зовнішнім ПЗ (наприклад, MS Excel).

Надійність консольного застосунку забезпечується багаторівневою системою перехоплення виняткових ситуацій (try-ехсепт). Система успішно блокує спроби введення некоректних типів даних, від'ємних значень чи неіснуючих складських зон без аварійного завершення роботи, гарантуючи відмовостійкість (Fault Tolerance).

Висновки

У результаті виконання роботи було успішно розв'язано актуальну науково-практичну задачу — спроектовано та програмно реалізовано відмовостійку консольну інформаційну систему складського обліку (Inventory Manager). Завдяки застосуванню парадигми об'єктно-орієнтованого програмування

вдалося побудувати гнучку архітектуру, яка точно відображає реальні логістичні сутності. Використання поліморфізму та наслідування суттєво спростило алгоритми обчислення вартості зберігання для різних категорій товарів без надлишкового дублювання коду, а розділення бізнес-логіки та інтерфейсу (за принципом SRP) зробило систему легкою для підтримки.

Важливим здобутком проекту є високий рівень надійності розробленого застосунку. Багаторівневий механізм перехоплення виняткових ситуацій (Fault Tolerance) гарантує повний захист внутрішньої бази даних від непередбачуваних помилок користувача, таких як введення некоректних типів даних чи від'ємних фінансових значень. Це зводить до нуля ризик аварійного завершення програми та забезпечує безперервність робочого процесу комерсанта.

Окрім того, інтегрована трирівнева підсистема персистентності повністю закриває потреби локального складу в зберіганні інформації: бінарна серіалізація об'єктів (Pickle) надійно консервує поточний стан сесії, використання JSON дозволяє гнучко керувати конфігураціями, а експорт у CSV забезпечує сумісність із зовнішнім корпоративним ПЗ для подальшого фінансового аудиту.

Таким чином, розроблений консольний застосунок мовою Python є обчислювально легким, ефективним та готовим до впровадження інструментом для оптимізації рутинних операцій локальної складської логістики. Його модульна структура, мінімальні вимоги до системних ресурсів та висока швидкість обробки даних створюють надійне інженерне підґрунтя для подальшого масштабування функціоналу.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Соммервілл І. Програмна інженерія. Об'єктно-орієнтований підхід. 10-те видання. К.: Вільямс, 2016. 816 с.
2. Офіційна документація мови програмування Python 3. [Електронний ресурс] – URL: <https://docs.python.org/3/>
3. Робота з бінарною серіалізацією (модуль pickle) у Python. [Електронний ресурс] – URL: <https://docs.python.org/3/library/pickle.html>
4. Робота з конфігураційними даними (модуль json) у Python. [Електронний ресурс] – URL: <https://docs.python.org/3/library/json.html>
5. Специфікація Unified Modeling Language (UML). [Електронний ресурс] – URL: <https://www.omg.org/uml/>

Горейко Максим Вікторович — студент групи СА-25б, факультет інтелектуальних інформаційних технологій та автоматизації, Вінницький національний технічний університет, м. Вінниця, e-mail: mxmgrko@gmail.com

Жуков Сергій Олександрович — к.т.н., доцент кафедри системного аналізу та інформаційних технологій, Вінницький національний технічний університет, e-mail: sazhukov@gmail.com

Норейко Максим V. — student of group SA-25b, Faculty of Intelligent Information Technologies and Automation, Vinnytsia National Technical University, Vinnytsia, e-mail: mxmgrko@gmail.com

Zhukov Serhii O. — Ph.D., Assistant Professor of the Department of Systems Analysis and Information Technologies, Vinnytsia National Technical University, Vinnytsia, e-mail: sazhukov@gmail.com