

ОБ'ЄКТНО-ОРІЄНТОВАНЕ ПРОЄКТУВАННЯ ТА РОЗРОБКА КОНСОЛЬНИХ СИСТЕМ УПРАВЛІННЯ ТРАНСПОРТНОЮ ЛОГІСТИКОЮ

Вінницький національний технічний університет

Анотація

У роботі досліджено принципи об'єктно-орієнтованого аналізу та проєктування транспортно-логістичних систем, а також розроблено консольний застосунок «Global Logistics UA» для автоматизованого керування автопарком підприємства. Обґрунтовано доцільність створення локального програмного рішення для диспетчерського обліку на основі аналізу існуючих промислових аналогів. Побудовано гнучку архітектуру на базі фундаментальних принципів ООП, де реалізовано поліморфні алгоритми калькуляції рейсів, строгу інкапсульовану валідацію вантажопідйомності за допомогою сетерів. Використання мови Python та таких інструментів, як бінарна серіалізація pickle, кастомні менеджери контексту та ітераційні протоколи, забезпечує оптимізацію використання оперативної пам'яті, гнучкість системи та надійний захист даних від перевантажень.

Ключові слова: Python, ООП, поліморфізм, наслідування, інкапсуляція, композиція, транспортна логістика, управління автопарком, валідація, менеджер контексту, кастомний ітератор, серіалізація.

Abstract

The paper investigates the principles of object-oriented analysis and design of transport and logistics systems, and develops a console application "Global Logistics UA" for automated enterprise fleet management. The expediency of creating a local software solution for dispatch accounting is substantiated based on the analysis of existing industrial analogues. A flexible architecture has been built on the basis of fundamental OOP principles, implementing polymorphic algorithms for route cost calculation and strict encapsulated validation of payload capacity using setters. The use of the Python language and tools such as pickle binary serialization, custom context managers, and iteration protocols ensures optimization of RAM usage, system flexibility, and reliable data protection against overloading.

Keywords: Python, OOP, polymorphism, inheritance, encapsulation, composition, transport logistics, fleet management, validation, context manager, custom iterator, serialization.

Вступ

Об'єктно-орієнтоване програмування (ООП) є базовою парадигмою для проєктування сучасного програмного забезпечення, зокрема у сфері транспортної логістики. Моделювання сутностей автопарку у вигляді взаємодіючих класів дозволяє ефективно перенести реальні логістичні процеси у гнучкі цифрові структури.

У межах розробки інформаційної системи «Global Logistics UA» ключовими завданнями є суворі типізація рухомого складу, інкапсульована валідація лімітів вантажопідйомності за допомогою сетерів та поліморфний розрахунок вартості доставки. Звичайні лінійні підходи до обліку не забезпечують цілісності даних при зміні параметрів автопарку. Для підвищення відмовостійкості програми було інтегровано розширені інструменти мови Python: кастомні менеджери контексту для безпечного контролю завантаження машин (VehicleLoadTransaction) та ітераційні протоколи (ActiveVehicleIterator), що забезпечують оптимізацію оперативної пам'яті та захист від логічних збоїв.

Результати дослідження

Під час створення консольної програми «Global Logistics UA» було досліджено, як краще організувати роботу транспортної компанії та зручно керувати автопарком. Завдяки використанню об'єктно-орієнтованого програмування (ООП) код розділено на окремі логічні модулі, а сама програма працює чітко та надійно.

Було розглянуто готові популярні системи для логістики, такі як Wialon, Odoo Fleet та Samsara. Виявилося, що всі вони мають спільні недоліки: надто складні інтерфейси, довге налаштування та

багато зайвих функцій для маленьких фірм , а також вимога купувати дороге GPS-обладнання чи платити за підписку.

Тому виникла потреба створити власний спрощений текстовий застосунок, який є легким у використанні, не вимагає складних конфігурацій і допомагає диспетчеру вести базовий облік. Програма працює швидко, є стійкою до помилок користувача , не навантажує комп'ютер і її легко розширювати в майбутньому.

На основі цих вимог розроблено структуру програми та складено діаграму використання (Use Case). Головні функції дозволяють користувачеві додавати нові вантажівки (Truck) та фургони (Van) , причому система сама перевіряє ліміти їхньої вантажності. Також у програмі можна переглянути весь список машин , безпечно додавати вантаж завдяки менеджеру контексту (який автоматично скасовує зміни, якщо є перевантаження) , бачити через кастомний ітератор лише ті авто, що зараз у рейсі , та зберігати відсортовані звіти у форматі CSV. Автоматично у фоні програма зберігає всю базу даних через модуль pickle та зчитує налаштування з файлу JSON (рис. 1).

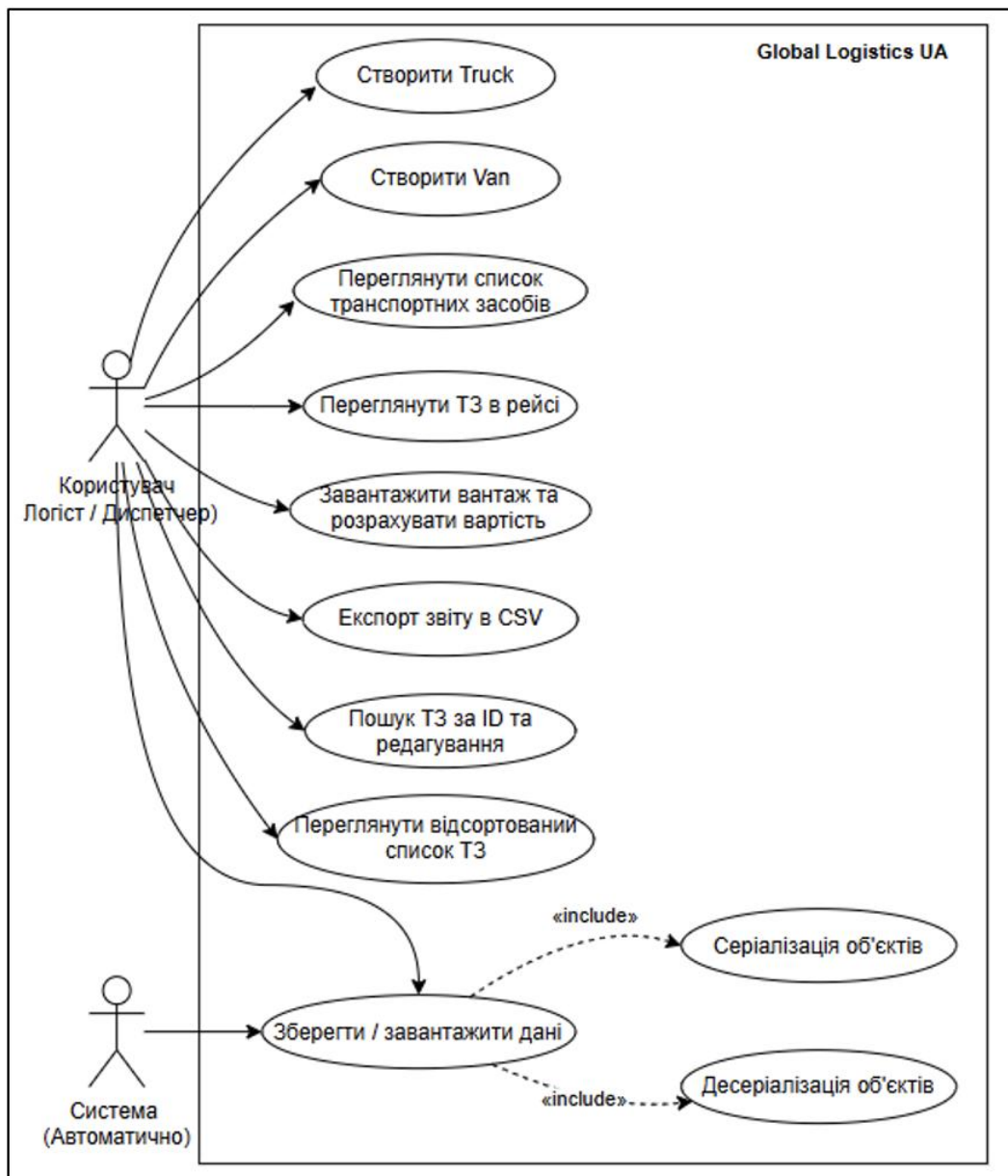


Рисунок 1. Use case діаграма додатку

Робота застосунку починається із безпечного зчитування конфігураційних параметрів із файлу JSON та автоматичного відновлення збереженої структури об'єктів із бінарного файлу за допомогою

модуля pickle. Після цього користувач через інтерактивний цикл головного меню може реєструвати нові транспортні засоби, вказуючи їхній унікальний ID, марку, модель, тип пального та максимальну вантажопідйомність.

При перегляді поточного стану автопарку керуючий модуль використовує кастомний ітератор ActiveVehicleIterator, який динамічно відбирає та виводить лише ті автомобілі, що зараз завантажені та перебувають у рейсі. Будь-яка зміна поточного навантаження машини повністю підпорядкована логіці контекстного менеджера VehicleLoadTransaction, який оновлює вагу та автоматично скасовує зміни (робить відкат до безпечного стану), якщо виявлено спробу перевантаження. Додаток також формує аналітичні звіти, обчислюючи поліморфну вартість доставки, сортує транспорт за його спроможністю та дозволяє експортувати ці дані у формат CSV (рис. 2).

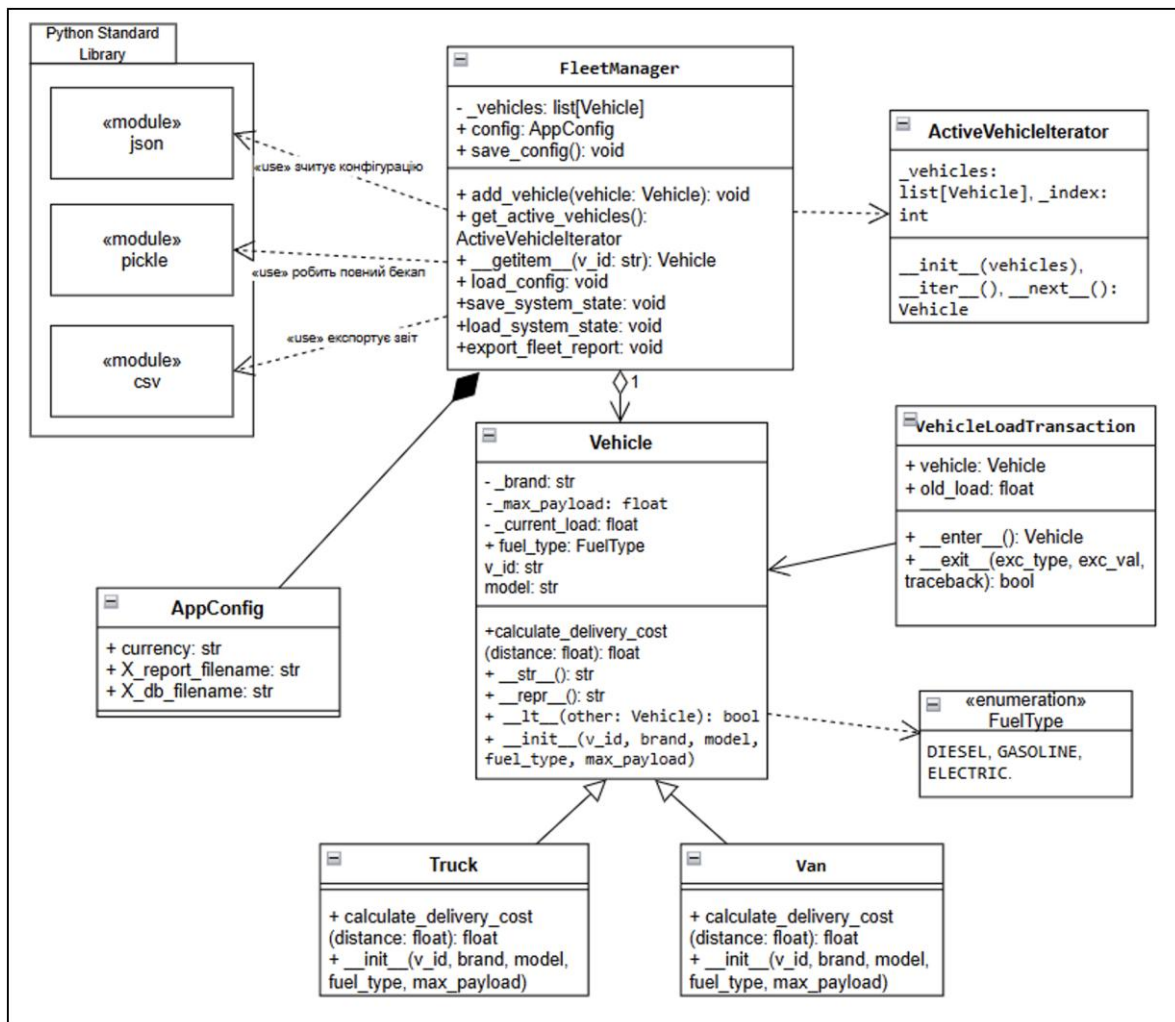


Рисунок 2. Діаграма класів

Структура додатку побудована на глибокому використанні парадигми об'єктно-орієнтованого програмування. Базовим елементом виступає клас Vehicle, який інкапсулює фундаментальні атрибути (ID, марка, модель, тип пального, ліміти ваги) та використовує властивості у вигляді геттерів і сетерів для обов'язкової валідації даних. Специфічні категорії транспорту реалізовані через класи-нащадки Truck (Вантажівка) та Van (Фургон) за допомогою механізмів наслідування та поліморфізму, де для кожного типу авто перевизначено унікальний алгоритм розрахунку вартості рейсу.

Централізоване управління всією колекцією машин покладено на клас-менеджер FleetManager. Він пов'язаний із об'єктом конфігурації AppConfig відношенням жорсткої композиції, а з об'єктами транспорту — через агрегацію. Для зберігання глобальних налаштувань середовища використовується зручний системний модуль dataclasses. Також у системі реалізовано власні ітератори

(ActiveVehicleIterator) для гнучкого обходу лише завантаженого транспорту та контекстні менеджери (VehicleLoadTransaction) для безпечної та транзакційної зміни ваги на борту.

Для технічної реалізації проєкту було використано мову програмування Python у середовищі Visual Studio Code. Для довготривалого локального збереження об'єктної моделі (персистентності) застосовано стандартний модуль pickle для бінарної серіалізації. Налаштування системи зберігаються у форматі JSON за допомогою використання однойменного модуля. Генерація підсумкових звітів із повною підтримкою кирилиці здійснюється за допомогою стандартного модуля csv, а безпечно й надійне логування помилок чи системних збоїв реалізовано через вбудований інструмент logging.

Висновки

У результаті виконання роботи було успішно спроектовано та розроблено програмний консольний додаток «Global Logistics UA». Практично доведено, що використання об'єктно-орієнтованого підходу дозволяє створити легку та надійну систему для диспетчерського обліку без використання громіздких комерційних аналогів. На основі розробленої архітектури отримано такі результати:

1. Реалізовано ієрархію наслідування та поліморфізм у підкласах Truck (Вантажівка) та Van (Фургон), що дозволило уникнути дублювання коду та забезпечило автоматичний розрахунок унікальної вартості доставки для кожного типу авто без конструкцій if/else.

2. Впроваджено інкапсуляцію полів суперкласу Vehicle за допомогою геттерів та сетерів властивостей, що забезпечило автоматичну превентивну валідацію даних і заблокувало можливість введення некоректних технічних відомостей користувачем.

3. Написано кастомний контекстний менеджер VehicleLoadTransaction для операцій із завантаженням машин, що дозволило захистити систему від збоїв за рахунок механізму автоматичного відкату (rollback) ваги до початкового стану у разі виявлення перевантаження .

4. Розроблено ітераційний протокол ActiveVehicleIterator для обходу колекції класу-менеджера FleetManager, що дозволило реалізувати «лінивий» покроковий вивід лише завантажених машин, економлячи ресурси оперативної пам'яті комп'ютера .

5. Інтегровано модулі pickle, json та csv, завдяки чому було повністю автоматизовано процеси збереження об'єктного графа між запусками терміналу, гнучкого зчитування конфігурацій та генерації табличних звітів для бізнесу .

Створена модульна архітектура є повністю автономною, стійкою до помилок користувача і придатною для подальшого розширення та масштабування.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Бичков, О. С. Основи об'єктно-орієнтованого програмування [Текст] : підручник / О. С. Бичков, І. Ю. Жук, Г. В. Порєв ; КНУ ім. Т. Шевченка. – Київ : Каравела, 2023. – 204 с. 2021. – 514 с.
2. Висоцька, В. А. Python: алгоритмізація та програмування [Текст] : навчальний посібник / В. А. Висоцька, О. В. Оборська ; НУ "Львівська політехніка". – Львів : Новий Світ-2000, 2023. – 516 с.
3. Лосєв, М. Ю. Програмування мовою Python [Текст] : навчальний посібник / М. Ю. Лосєв, В. М. Федорченко ; ХНЕУ ім С. Кузнеця. – Харків : ХНЕУ ; Львів : Новий Світ-2000, 2023. – 178 с.

Яковенко Каріна Олександрівна – студентка групи СА-25б, факультет інтелектуальних інформаційних технологій та автоматизації, Вінницький національний технічний університет, м. Вінниця, e-mail: 26karina07@gmail.com

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

Yakovenko Karina O. – student of group SA-25b, Faculty of Intellectual Information Technologies and Automation, Vinnytsia National Technical University, Vinnytsia, e-mail: 26karina07@gmail.com

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