

ОБ'ЄКТНО-ОРІЄНТОВАНЕ МОДЕЛЮВАННЯ ТА АРХІТЕКТУРНЕ ПРОЄКТУВАННЯ КОНСОЛЬНИХ СИСТЕМ УПРАВЛІННЯ ЗАВДАННЯМИ

Вінницький національний технічний університет

Анотація

У роботі представлено розробку програмного консольного додатку «Система управління завданнями (Kanban CLI)», створеного із застосуванням принципів об'єктно-орієнтованого програмування. Додаток надає користувачеві функціонал для ефективного керування робочим процесом, включаючи створення завдань різних типів, розподіл їх за статусами, призначення пріоритетів, а також фільтрацію та сортування справ за критичністю. Використання мови Python та таких інструментів, як функтори, кастомні ітератори і генератори, забезпечує оптимізацію використання оперативної пам'яті, гнучкість системи та зручну організацію робочого часу. Особливу увагу приділено структурно-функціональному аналізу предметної галузі та декомпозиції логіки керування станами об'єктів, що дозволяє використовувати систему як надійну базу для автоматизації персонального менеджменту.

Ключові слова: консольний застосунок, управління завданнями, Kanban, об'єктно-орієнтоване програмування, Python, ітератор, серіалізація даних, структурна декомпозиція, об'єктне моделювання.

Abstract

The paper presents the development of a console software application, "Task Management System (Kanban CLI)", created using the principles of object-oriented programming. The application provides the user with functionality for effective workflow management, including the creation of tasks of various types, their distribution by status, priority assignment, as well as filtering and sorting tasks by criticality. The use of the Python language and tools such as functors, custom iterators, and generators ensures the optimization of RAM usage, system flexibility, and convenient organization of working time. Particular attention is paid to the structural and functional analysis of the domain and the decomposition of object state management logic, enabling the system's application as a reliable foundation for personal management automation.

Ключові слова: console application, task management, Kanban, object-oriented programming, Python, iterator, data serialization, structural decomposition, object modeling.

Вступ

У сучасному світі постійне зростання обсягів інформації та високий темп роботи вимагають ефективних рішень для організації часу, через що програмні системи для управління завданнями стають невід'ємною частиною повсякдення. Спрощення процесу організації робочого часу на основі принципів методології Kanban дозволяє користувачам ефективно розподіляти навантаження, зменшує ризик пропуску дедлайнів, допомагає фокусуватися лише на актуальних задачах та цілісно оцінювати особисту продуктивність.

З погляду системного аналізу, проєктування подібних інструментів персонального менеджменту вимагає чіткої формалізації інформаційних потоків та побудови стійкої об'єктної архітектури. Це є необхідною умовою для мінімізації ризиків логічних помилок при переході завдань між станами, забезпечення високої обчислювальної ефективності процесу, оптимізації використання оперативної пам'яті та надійного збереження прогресу між сесіями без надмірного навантаження на локальні ресурси пристрою.

Метою роботи є дослідження принципів об'єктно-орієнтованого програмування та розробка консольної системи управління завданнями (Kanban CLI), яка забезпечує створення, класифікацію, гнучке відстеження за статусами і пріоритетами, а також зручну фільтрацію та сортування справ під поточні потреби користувача за допомогою інструментів мови Python (зокрема кастомних ітераторів, генераторів, функторів та систем серіалізації даних).

Результати дослідження

У процесі розроблення консольного застосунку «Kanban CLI» було проведено комплексне дослідження, спрямоване на оптимізацію процесів організації робочого часу та розробку гнучкого інструменту для управління завданнями. Отримані результати стали основою для застосування об'єктно-орієнтованого підходу, створення модульного коду та реалізації ефективної бізнес-логіки системи.

Було проаналізовано ряд популярних систем управління завданнями, зокрема Trello , Jira та Asana. Під час аналізу було виявлено ряд недоліків: складність інтерфейсів та надлишковий функціонал для невеликих проєктів (як у Jira) , а також обмеженість частини важливих функцій, які доступні лише у платних версіях чи розширених тарифах (у Trello та Asana).

Отже, виникла потреба у створенні власного спрощеного консольного застосунку, який є простішим у використанні, не потребує складного налаштування та дозволяє реалізувати основні функції системи управління завданнями. Розроблений додаток поєднує в собі високу швидкодію, мінімальне споживання системних ресурсів завдяки лінивому завантаженню даних та легко масштабується.

На основі цих потреб здійснено проектування програмного забезпечення. Основні можливості користувача та системи описані у вигляді Use Case діаграми на рис. 1. Головні сценарії передбачають, що користувач взаємодіє із системою для створення завдань типу BugReport та FeatureRequest з автоматичною валідацією параметрів. Інші ключові випадки використання включають посторінковий перегляд завдань (пагінацію) , зміну статусу з обов'язковим транзакційним аудитом , пошук, фільтрацію та вибірку задач за виконавцем. Також передбачено автоматичні фонові функції системи, такі як автоблокування прострочених задач та збереження або відновлення бази даних.

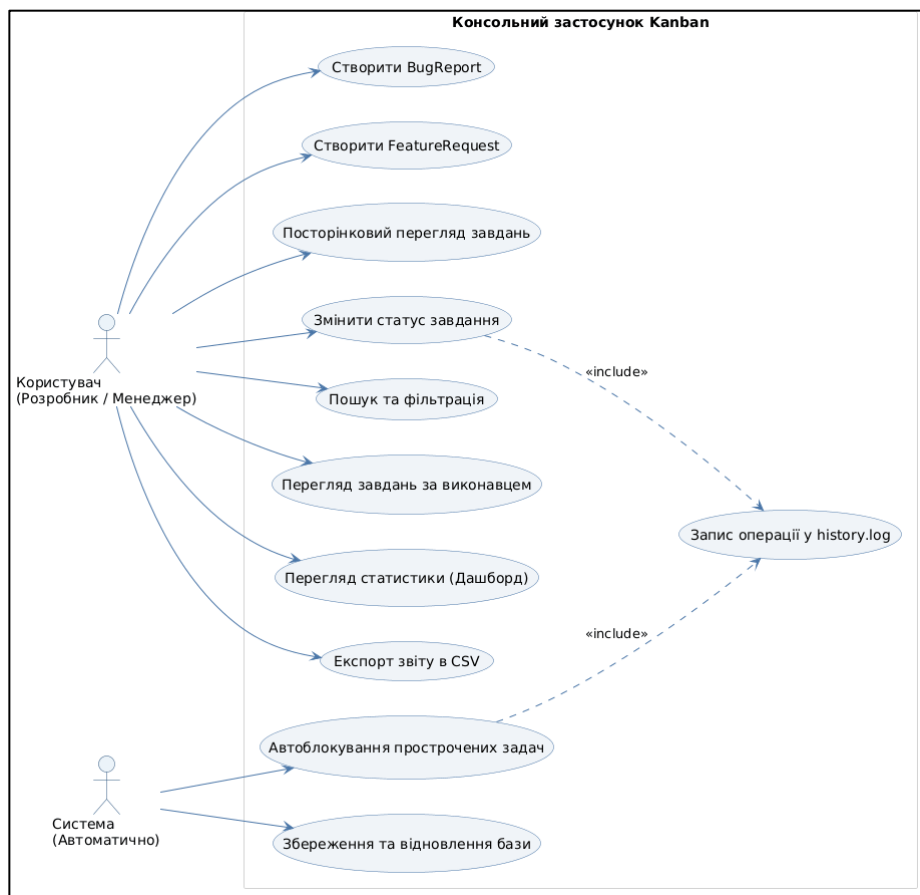


Рисунок 1. Use case діаграма додатку

Робота застосунку розпочинається з безпечного зчитування конфігураційних параметрів та відновлення збереженої структури об'єктів із бінарного файлу за допомогою підсистеми десеріалізації. Після цього користувач через інтерактивний цикл головного меню може створювати нові завдання,

вказуючи назву, пріоритет, виконавця та дедлайн. При перегляді стану Kanban-дошки керуючий менеджер використовує кастомний ітератор, який динамічно сортує об'єкти від найвищого пріоритету до найнижчого. Зміна статусу завдання повністю підпорядкована логіці контекстного менеджера, який оновлює стан та автоматично фіксує транзакцію у системному журналі. Додаток також формує аналітичний дашборд, обчислюючи складність завдань, блокує прострочені дедлайни і дозволяє експортувати звіт у формат CSV.

На рис. 2 зображена діаграма класів застосунку «Kanban CLI», яка є статичним представленням структури програмної моделі.

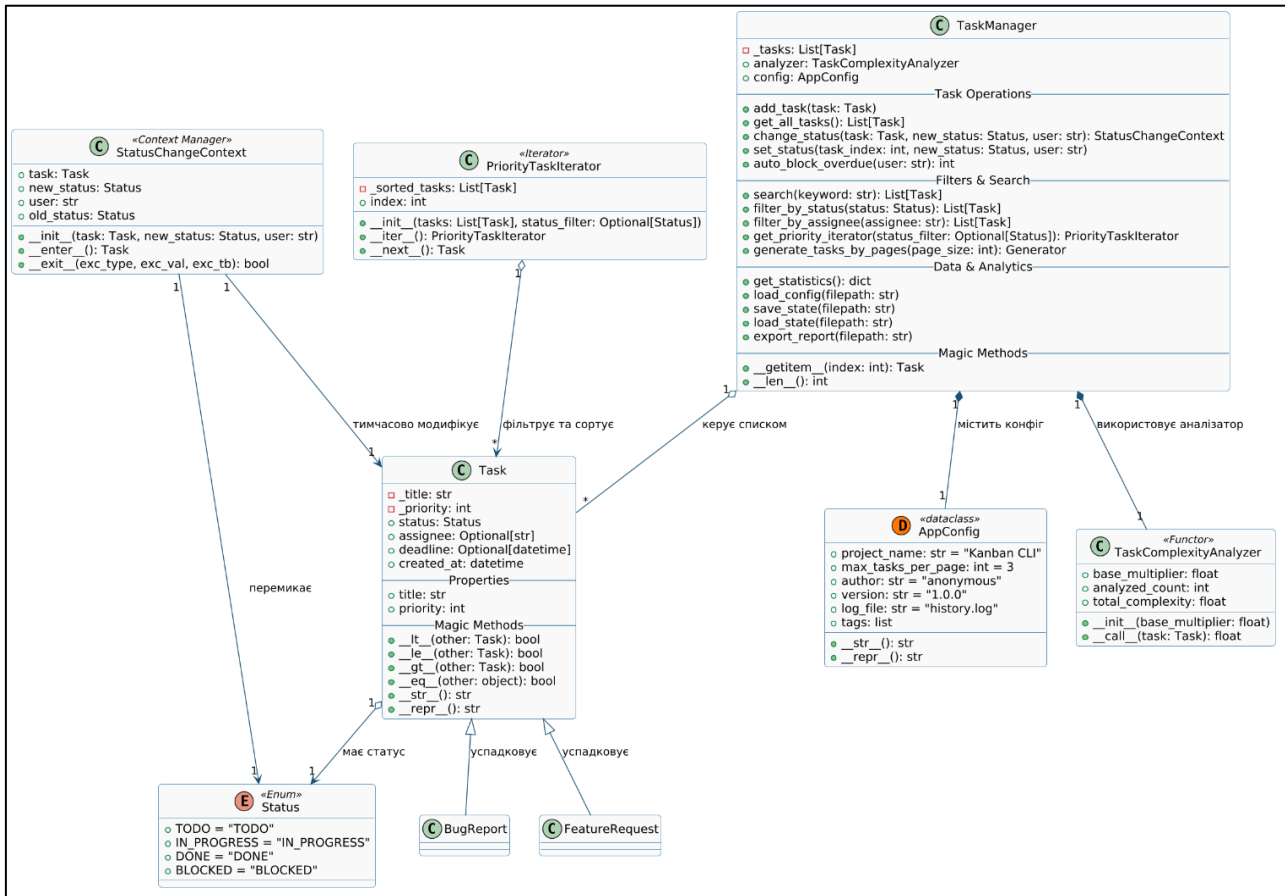


Рисунок 2. Діаграма класів

Структура додатку побудована на глибокому використанні парадигми об'єктно-орієнтованого програмування. Базовим елементом виступає клас Task, який інкапсулює фундаментальні атрибути та використовує властивості (properties) для валідації даних. Специфічні типи завдань реалізовані через класи-нащадки BugReport та FeatureRequest за допомогою наслідування та поліморфізму. Централізоване управління життєвим циклом завдань покладено на клас-менеджер TaskManager, який пов'язаний із завданнями відношенням композиції. Для зберігання налаштувань використовується оптимізований системний модуль dataclasses (клас AppConfig). Також реалізовано власні ітератори (PriorityTaskIterator) для гнучкого обходу колекцій та контекстні менеджери (StatusChangeContext) для безпечної зміни станів.

Для задоволення потреб у розробці було використано мову програмування Python у середовищі Visual Studio Code. Для довготривалого локального збереження об'єктної моделі (персистентності) застосовано стандартний модуль pickle, який забезпечує бінарну серіалізацію. Налаштування середовища та параметрів системи зберігаються у форматі JSON за допомогою однойменного модуля. Генерація аналітичних звітів із підтримкою кирилиці здійснюється з використанням стандартного модуля csv, а безпечне логування бізнес-операцій реалізовано за допомогою вбудованого інструменту logging.

Висновки

У результаті виконання роботи було успішно спроектовано та розроблено програмний консольний додаток «Система управління завданнями (Kanban CLI)». Практично доведено, що використання принципів об'єктно-орієнтованого програмування на мові Python дозволяє створити гнучку, масштабовану та зручну систему для організації робочого часу.

Реалізований застосунок ефективно вирішує завдання керування поточними процесами, надаючи користувачеві інструменти для створення, пріоритетизації та відстеження завдань без перевантаження складним інтерфейсом чи зайвим функціоналом, характерним для великих комерційних систем. Застосування таких технологічних рішень, як кастомні ітератори, контекстні менеджери та механізми бінарної серіалізації, забезпечило високу швидкодію, надійне збереження даних між сесіями та оптимізацію використання системних ресурсів.

Розроблений додаток є повністю автономним і готовим до використання, а його модульна архітектура створює міцне підґрунтя для подальшого масштабування. У перспективі функціонал системи може бути розширений шляхом додавання повноцінного графічного інтерфейсу, інтеграції з хмарними базами даних або впровадження системи сповіщень через популярні месенджери.

Проведене дослідження підтверджує, що чіткий архітектурний розподіл бізнес-логіки та використання вбудованих інструментів оптимізації мови Python забезпечують необхідний рівень гнучкості системи. Це створює надійне підґрунтя для її подальшої інтеграції в більш масштабні інформаційно-аналітичні комплекси підтримки прийняття рішень.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Бичков, О. С. Основи об'єктно-орієнтованого програмування [Текст] : підручник / О. С. Бичков, І. Ю. Жук, Г. В. Порев ; 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.

Бакус Анна Дмитрівна – студентка групи СА-256, факультет інтелектуальних інформаційних технологій та автоматизації, Вінницький національний технічний університет, м. Вінниця, e-mail: bakusanna69@gmail.com

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

Bakus Anna D. – student of group SA-25b, Faculty of Intellectual Information Technologies and Automation, Vinnytsia National Technical University, Vinnytsia, e-mail: bakusanna69@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