

# РОЗРОБКА КОНСОЛЬНОГО ЗАСТОСУНКУ ДЛЯ ПЛАНУВАННЯ РОЗКЛАДУ ІЗ ЗАСТОСУВАННЯМ ПРИНЦИПІВ ОБ'ЄКТНО-ОРІЄНТОВАНОГО ПРОГРАМУВАННЯ НА PYTHON

Вінницький національний технічний університет

## *Анотація*

*У роботі реалізовано консольний додаток «Планувальник розкладу» на базі принципів об'єктно-орієнтованого програмування на Python. Програма дозволяє автоматизувати процес формування розкладу занять, підтримує рольову модель доступу (User/Admin) та автоматично перевіряє доступність аудиторного фонду. Використання контекстних менеджерів дозволяє надійно блокувати часові колізії (накладки занять в одному кабінеті) під час створення нових подій, а кастомні ітератори забезпечують швидкий покроковий відбір пар за номерами аудиторій без перевантаження оперативної пам'яті.*

**Ключові слова:** об'єктно-орієнтоване програмування, Python, консольний додаток, планування розкладу, інкапсуляція, поліморфізм, кастомний ітератор, контекстний менеджер.

## **Abstract**

*The paper describes the development of a console application "Schedule Planner" based on object-oriented programming principles in Python. The application automates the university scheduling process, supports a role-based access model (User/Admin), and checks classroom availability. Implementing context managers ensures that time collisions are blocked automatically when adding new events, while custom iterators enable fast and memory-efficient step-by-step filtering of classes by room numbers.*

**Keywords:** object-oriented programming, Python, console application, schedule planning, encapsulation, polymorphism, custom iterator, context manager.

## **Вступ**

Формування навчального розкладу в університетах є складним процесом, який вимагає постійного контролю великої кількості інформації. Автоматизація цього процесу спрощує роботу диспетчерів та студентів, оскільки зменшує ризик появи часових або просторових накладок занять (колізій) та дозволяє ефективно використовувати аудиторний фонд.

З погляду системного аналізу, проектування подібних інструментів вимагає побудови стійкої об'єктно-орієнтованої архітектури. Це є критичною умовою для швидкої фільтрації даних, надійного збереження бази розкладу між запусками програми та оптимізації оперативної пам'яті через «лінійні» обчислення.

Метою роботи є розробка консольного застосунку «Планувальник розкладу (Schedule CLI)» на базі принципів ООП у середовищі Python. Програма реалізує гнучке відстеження завантаженості викладачів, посторінкове виведення занять та автоматичне блокування накладок за допомогою контекстних менеджерів, кастомних ітераторів, генераторів та вбудованих модулів серіалізації даних..

## **Результати дослідження**

У процесі розроблення консольного застосунку «Планувальник розкладу» було проведено комплексне дослідження, спрямоване на оптимізацію процесів організації навчального часу та розробку гнучкого інструменту для управління розкладом занять. Отримані результати стали основою для застосування об'єктно-орієнтованого підходу, створення модульного коду та реалізації ефективної бізнес-логіки системи для уникнення часових конфліктів [1].

Було проаналізовано ряд популярних систем управління розкладом, зокрема My Study Life, Prime Timetable та Smart Timetable. Під час аналізу було виявлено ряд недоліків: відсутність суворих механізмів автоматизованого контролю за використанням фізичних ресурсів (аудиторного фонду),

складність візуальної валідації при великих масивах даних, а також слабка бізнес-логіка, яка часто допускає створення візуальних «накладок» занять в одній аудиторії на один і той самий час.

Отже, виникла потреба у створенні власного спрощеного консольного застосунку, який є простішим у використанні, гарантує відсутність логічних конфліктів та дозволяє реалізувати основні функції системи планування. Розроблений додаток поєднує в собі високу швидкодію, мінімальне споживання системних ресурсів завдяки кастомним ітераторам та лінивому завантаженню даних, а також 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 базуються на рольовій моделі доступу, де базовий користувач (User) взаємодіє із системою для перегляду розкладу, пошуку занять викладача та підбору вільних аудиторій. Адміністратор (Admin) успадковує ці права та має розширені можливості: керування розкладом (яке включає скасування пари та додавання події), експорт даних у формат CSV, а також збереження і завантаження стану системи за допомогою модуля Pickle. Також передбачено автоматичну фонову функцію системи — обов'язкову перевірку накладок аудиторії, яка ініціюється щоразу при створенні нової події [2,3].



Рисунок 1. Use case діаграма додатку

Експлуатація застосунку розпочинається з динамічного завантаження конфігураційних налаштувань та ініціалізації робочого простору, після чого підсистема десеріалізації відновлює попередньо збережену базу розкладу із бінарного файлу. Далі користувач, пройшовши авторизацію через систему рольового доступу, потрапляє до інтерактивного меню, де отримує змогу додавати нові навчальні події, вказуючи дисципліну, викладача, аудиторію та часові межі. При відображенні розкладу клас-менеджер автоматично сортує об'єкти у хронологічному порядку, а для вибірки занять за конкретним приміщенням застосовується кастомний ітератор. Процес створення пар суворо контролюється спеціальним контекстним менеджером, який на льоту перевіряє доступність аудиторій та надійно блокує будь-які часові колізії (накладки). Окрім цього, система підраховує сумарне навантаження викладачів за допомогою функтора, підтримує посторінкове виведення даних (пагінацію) та дозволяє експортувати готові звіти у формат CSV.

На рис. 2 зображена UML-діаграма класів застосунку «Планувальник розкладу», яка є детальним статичним представленням об'єктно-орієнтованої архітектури розробленої моделі.

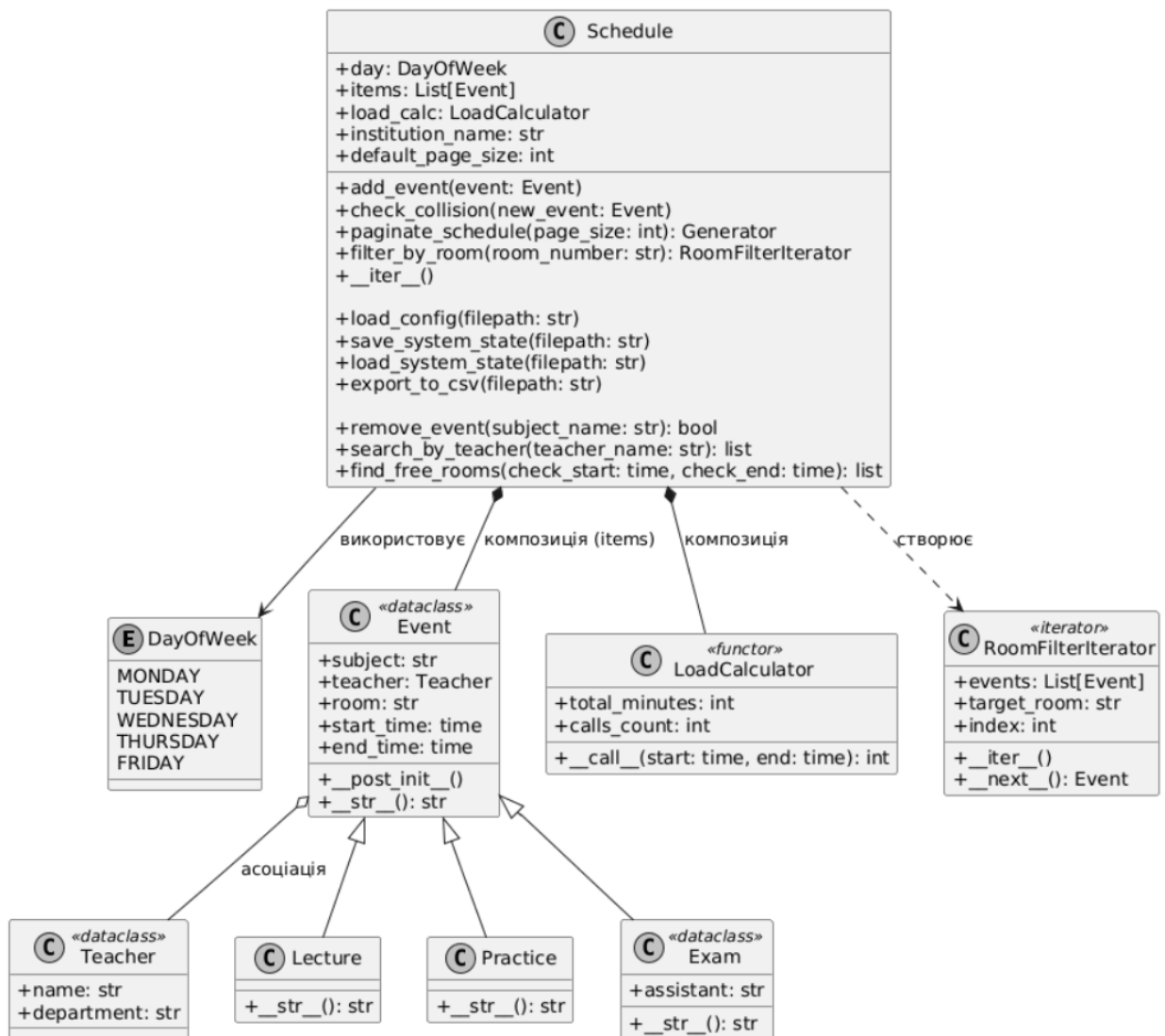


Рисунок 2. Діаграма класів

Структура додатку побудована на глибокому використанні парадигми об'єктно-орієнтованого програмування. Базовим елементом виступає клас Event (Подія), який інкапсулює фундаментальні атрибути заняття, використовуючи можливості модуля dataclasses та магічного методу `__post_init__` для строгої валідації часових меж. Окремо захист критичних даних реалізовано через властивості (properties) у класі Teacher. Специфічні типи навчальних подій реалізовані через класи-нащадки Lecture, Practice та Exam за допомогою механізмів наслідування та поліморфізму. Централізоване управління життєвим циклом розкладу покладено на клас-менеджер Schedule, який пов'язаний із колекцією подій жорстким відношенням композиції. Для строгої типізації робочих днів застосовано перелічуваний тип Enum (DayOfWeek). Також реалізовано власні ітератори (RoomFilterIterator) для економного обходу колекцій при фільтрації та контекстні менеджери (CollisionChecker) для безпечного додавання пар із блокуванням часових накладок [2].

Для розробки застосунку було використано мову програмування Python у середовищі Visual Studio Code. Для довготривалого локального збереження об'єктної моделі (персистентності) застосовано стандартний модуль pickle, який забезпечує повноцінну бінарну серіалізацію графа об'єктів. Конфігураційні параметри системи (назва закладу, ліміти пагінації) гнучко зберігаються у форматі JSON за допомогою однойменного модуля. Генерація аналітичних табличних звітів із підтримкою кирилиці здійснюється з використанням стандартного модуля csv, а підрахунок загального навчального навантаження реалізовано за допомогою спеціалізованого патерна «Функтор» (LoadCalculator) [3].

## Висновки

У результаті виконання роботи було розроблено консольний застосунок «Планувальник розкладу (Schedule CLI)». Завдяки проведеному дослідженню предметної області та проектуванню UML-діаграм, вдалося чітко розділити логіку системи. На основі розробленої архітектури отримано такі результати:

1. Реалізовано базові принципи ООП: наслідування та поліморфізм у підкласах Lecture, Practice, Exam дозволили уникнути дублювання коду та реалізувати специфічні формати виведення рядків для різних типів занять. Інкапсуляція та магічний метод `__post_init__` захистили об'єкти Event від введення некоректного часу початку та завершення пар.

2. Створення класу-менеджера Schedule на основі композиції дозволило централізовано керувати колекцією подій, а використання контекстного менеджера CollisionChecker вирішило проблему часових накладок, автоматично блокуючи додавання нової події, якщо аудиторія в цей час уже зайнята.

3. Написання кастомного ітератора RoomFilterIterator та методу-генератора пагінації забезпечило «лінійний» покроковий перебір занять, що мінімізує навантаження на оперативну пам'ять.

4. Інтеграція модулів pickle, json та csv забезпечила надійне збереження бази розкладу між запусками терміналу, читання зовнішніх конфігурацій та можливість експорту фінансових чи аналітичних звітів у табличний формат.

Створена модульна архітектура є гнучкою, що дає змогу в майбутньому легко підключити графічний інтерфейс (GUI) або інтегрувати додаток із хмарними базами даних університету, взагалі не змінюючи логічне ядро програми.

## 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.

**Воронова Ірина Олександрівна** – студентка групи СА-25б, факультет інтелектуальних інформаційних технологій та автоматизації, Вінницький національний технічний університет, м. Вінниця, e-mail: irav78805@gmail.com

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

**Voronova Iryna O.** – student of group SA-25b, Faculty of Intellectual Information Technologies and Automation, Vinnytsia National Technical University, Vinnytsia, e-mail: irav78805@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