

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

Факультет прикладної математики

Кафедра програмного забезпечення комп'ютерних систем

Курсова робота

**з дисципліни «Компоненти програмної інженерії. Якість та тестування програмного
забезпечення»**

на тему: «Файлова система»

Виконав:

студент II курсу, групи КП-03

Хоменко Максим Вячеславович

Керівник роботи:

Погорелов Володимир Володимирович

Оцінка _____

(дата, підпис)

ЗМІСТ

СПИСОК ВИЗНАЧЕНЬ, СКОРОЧЕНЬ ТА ПОЗНАЧЕНЬ.....	3
ВСТУП.....	4
1. СТРУКТУРНО-АЛГОРИТМІЧНА ОРГАНІЗАЦІЯ ПРОГРАМНОЇ СИСТЕМИ МАГАЗИНУ ДИТЯЧИХ ІГРАШОК.....	6
1.1.Модульна організація програми.....	6
1.2.Функціональні характеристики.....	8
2. ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ ЗА ДОПОМОГОЮ ШАБЛОНІВ ПРОЄКТУВАННЯ.....	9
2.1.Обґрунтування вибору та опис шаблонів проєктування для програмної реалізації фалової системи.....	9
ВИСНОВКИ.....	28
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ.....	29

СПИСОК ВИЗНАЧЕНЬ, СКОРОЧЕНЬ ТА ПОЗНАЧЕНЬ

ПЗ – програмне забезпечення.

Фреймворк – програмне забезпечення, яке полегшує процес розроблення

на об'єднання різних модулів програмного проекту.

API – додаток, що, по-перше, складається з набору функцій, об'єднаних іменованим простором, по-друге, виконує функції за запитами від користувача.

HTTP – протокол прикладного рівня, що використовується для обміну інформації між відкритими системами.

ВСТУП

Файлові системи існують для збереження, поповнення та перегляду інформації, наявної у директоріях/файлах.

Велика кількість файлових систем за своїм принципом роботи передбачають необхідність керувати даними: додавати директорії/файли(

файли певного типу), заповнювати файли деякою інформацією, змінювати місцезнаходження директорії/файлу, видаляти директорію/файл, а також переглядати структуру системи.

Рішення, презентоване в цій роботі, є кросплатформеним аналогом для персонального комп'ютера.

Об'єктом дослідження даного проєкту є процес обліку даних у форматі текстових рядків, збережених у текстових файлах різних типів.

Метою роботи є розроблення програмного забезпечення аналогу файлової системи.

Для досягнення визначеної мети необхідно виконати такі завдання:

- абстрагувати об'єкти предметної галузі;
- розробити структурну організацію ПЗ за допомогою застосування основних принципів ООП та шаблонів проектування;
- визначити та описати функціональні характеристики програми;
- створити прототипи;
- створити тести;
- виконати реалізацію програмного забезпечення відповідно до вимог технічного завдання та прототипів;
- контейнеризувати API за допомогою ПЗ Docker;
- виконати тестування розробленої програми;
- оформити документацію з курсової роботи.

Розроблене ПЗ складається з таких модулів:

- модуль бізнес-логіки, який містить класи, що містять логіку системи, взаємодіючи, реалізують вимоги до програмної системи;
- модуль тестування класів бізнес-логіки;
- модуль API, що містить ресурси(кінцеві точки), через які відбувається управління файловою системою(класами бізнес-логіки);
- модуль тестування API;

- консольного інтерфейсу користувача;
- модуль тестування консольного інтерфейсу користувача.

До функціональних можливостей програми належать: додавання, видалення, зміна місця знаходження файлу/директорії, читання структури файлової системи, вмісту файлу. Для функціонування розробленої програми необхідно забезпечити наявність на комп'ютері 4 ГБ оперативної пам'яті, від 800 МБ до 210 ГБ вільного місця на жорсткому диску, встановлений Docker.

Розроблене програмне забезпечення може бути використане для ознайомлення.

1. СТРУКТУРНО-АЛГОРИТМІЧНА ОРГАНІЗАЦІЯ ФАЙЛОВОЇ СИСТЕМИ

1.1. Модульна організація програми

Програмну систему умовно можна розділити на рівні: рівень бізнес-логіки, рівень API, консольний додаток.

На рівні бізнес-логіки знаходяться класи предметної області з певною поведінкою.

Рівень API характеризується набором методів, що викликаються клієнтом через HTTP.

Рівень клієнта у вигляді консольного додатка, що виконує HTTP-запити до сервера, змінюючи стан системи та отримуючи інформацію поточний.

1.2. Функціональні характеристики

Система надає такі можливості як:

- вмістом кореневої папки: додавання, видалення, релокація директорій/файлів;
- читання вмісту файлу;
- читання повної структури кореневої папки.

2. ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ ЗА ДОПОМОГОЮ ШАБЛОНІВ ПРОЄКТУВАННЯ

2.1. Обґрунтування вибору та опис шаблонів проєктування для програмної реалізації файлової системи

1) Композит

Визначення: Структурний шаблон проєктування, що надає абстрактний інтерфейс (набір методів) для управління серією вкладених об'єктів, пропагування операцій, винуваних над коренем, на компоненти-листи..

Проблема: Можна було б спробувати прямий підхід: розгорнути всі директорії, перебрати всі файли, а потім обчислити загальний вивід. Але в програмі це не так просто, як запуск циклу. Ви повинні заздалегідь знати класи директорій і файлів, через які ви проходите, рівень вкладеності директорій та інші неприємні деталі. Все це робить прямий підхід або занадто незручним, або навіть неможливим.

Рішення: Компонувальник пропонує розглядати файли і директорії через єдиний інтерфейс зі спільним методом отримання виводу.

Файли просто поверне своє ім'я, а Директорія запитає про вивід кожного предмета всередині себе і поверне загальний вивід зі своїм іменем. Якщо одним із внутрішніх елементів виявиться інша директорія, вона теж буде перебирати власний вміст, і так далі, допоки не порахується вивід усіх складових частин.

ВИСНОВКИ

Метою роботи було розроблення програмного забезпечення та тестування фалової системи.

Для досягнення визначеної мети було виконано в повному обсязі такі

завдання:

- абстрагування об'єктів предметної галузі;
- розробка структурної організації ПЗ за допомогою застосування основних принципів ООП та шаблонів проектування;
- визначити та описати функціональні характеристики програми;
- створити прототипи;
- створити тести;
- виконати реалізацію програмного забезпечення відповідно до вимог технічного завдання та прототипів;
- контейнеризувати API за допомогою ПЗ Docker;
- виконати тестування розробленої програми;
- оформити документацію з курсової роботи.

Розроблений застосунок надає можливість додавати/зберігати інформацію у форматі текстових рядків у файлах різного типу. Створені файли можна розміщувати в користувацьких директоріях. Структура файлової системи можна переглядати, так само як і вміст доданих файлів.

Для функціонування розробленої програми необхідна наявність на комп'ютері 4 ГБ оперативної пам'яті, від 800 МБ до 210 ГБ вільного місця на жорсткому диску, встановлений Docker.

Розроблене програмне забезпечення відтестоване на всіх рівнях: бізнес-логіка, API, клієнтська частина у вигляді консольного додатка.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Flask [Електронний ресурс]: [Веб-сайт]. - Режим доступу:

<https://flask.palletsprojects.com/en/2.2.x/>(дата звернення 29.12.2022) - Назва з екрана.

2. Дизайн-патерни - просто як двері /Андрій будай. 2012. — 90с.
3. Digital Ocean [Електронний ресурс]: [Веб-сайт]. - Режим доступу: <https://www.digitalocean.com/community/tutorials/how-to-install-and-use-docker-on-ubuntu-20-04> (дата звернення 29.12.2022) - Назва з екрана.
4. Pytest [Електронний ресурс]: [Веб-сайт]. - Режим доступу: <https://docs.pytest.org/en/7.1.x/contents.html> (дата звернення 29.12.2022) - Назва з екрана.
5. Robot Framework [Електронний ресурс]: [Веб-сайт]. - Режим доступу: <https://robotframework.org/> (дата звернення 29.12.2022) - Назва з екрана.