УДК 004.415.25

Кучерявий І.В.[1]
Ліщинська Л.Б.[1]

ДОСЛІДЖЕННЯ АРХІТЕКТУРНОГО ПАТЕРНУ MVC НА ПРИКЛАДІ МОВИ JAVA

[1]Вінницький національний технічний університет

Анотація

Досліджено архітектуру MVC, розподілення компонент програмного додатку, забезпечення гнучкої та масштабованої структури. Наведено приклад використання патерну MVC за допомогою фреймворку Spring MVC.

Ключові слова: шаблон проектування MVC, Spring, гнучкість та масштабованість додатків, архітектура ПЗ.

Abstract

The MVC architecture, the distribution of components of the software application, the provision of a flexible and scalable structure were studied. An example of using the MVC pattern using the Spring MVC framework is provided.

Keywords: MVC design pattern, Spring, application flexibility and scalability, software architecture.

Вступ

На сьогоднішній день архітектурний патерн Model-View-Controller (MVC) широко використовується в розробці програмного забезпечення для створення додатків з чітким розподілом обов'язків. Для мови Java шаблон MVC є популярним вибором для створення користувацьких інтерфейсів, веб-додатків та корпоративного програмного забезпечення [1]. Однак, незважаючи на те, що патерн MVC добре відомий і зрозумілий, існує багато варіацій і підходів до його реалізації в Java.

В зв'язку з актуальністю, необхідно дослідити ключові концепції патерну, включаючи ролі та обов'язки компонентів Model, View та Controller, а також механізми зв'язку між ними, а також варто розглянути практики проектування та реалізації Java-додатків на основі MVC, для досягнення вільного зв'язку між компонентами.

Основна частина

MVC (Model-View-Controller) – це архітектурний патерн, що використовується в програмній інженерії, який розділяє дані програми (Model), користувальницький інтерфейс (View) та елемент керування (Controller) на три окремі компоненти [2]. Головна ціль полягає в тому, щоб розділити компоненти, роблячи код простішим у розробці, підтримці та тестуванні.

Для реалізації архітектури MVC в Java створюються три основні компоненти:

1. Модель (Model): Цей компонент представляє дані та бізнес-логіку додатка. Інкапсулює дані та надає інтерфейс для доступу до них та їх модифікації. Він може взаємодіяти з базою даних та зовнішніми сервісами для отримання й зберігання даних. Для відображення моделі часто використовують POJO (Plain Old Java Objects) об'єкти.

2. Вигляд (View): Компонент використовується для представлення користувацького інтерфейсу програми. Він відповідає за представлення даних користувачеві у вигляді, зручному для розуміння і взаємодії з ними. У Java для Desktop додатків, View зазвичай реалізується за допомогою Swing або JavaFX, а для веб-додатків вигляд реалізується за допомогою шаблонізаторів таких як Thymeleaf або JSP (JavaServerPages).

3. Контролер (Controller): Цей компонент діє як посередник між моделлю та представленням. Він отримує користувацьке введення від Представлення, відповідно оновлює Модель, а потім оновлює Представлення, щоб відобразити зміни. Він містить потік управління додатком і реалізує будь-яку необхідну логіку для координації взаємодії між моделлю і виглядом.

Створення MVC-додатку на Java зазвичай починається зі створення класів моделі, які визначають дані та бізнес-логіку додатка. Потім створюються класи View, які визначають користувацький інтерфейс. Нарешті, потрібно створювати класи контролери, які будуть обробляти дані, введені користувачем, оновлювати модель і відповідно оновлювати представлення.

Для Java є популярний фреймворк SpringMVC. Щоб використовувати фреймворк MVC Spring, потрібно налаштувати кілька речей [3]:

1. Спершу потрібно налаштувати файл web.xml, щоб налаштувати DispatcherServlet, який відповідає за маршрутизацію запитів до відповідного контролера.

2. Далі потрібно створити контролери як класи Java з анотацією @Controller. Кожен контролер повинен мати методи з анотацією @RequestMapping, які обробляють конкретні запити від користувача.

3. Визначити шаблони подання за допомогою шаблонізатора, наприклад, Thymeleaf або JSP.

4. Налаштувати контекст додатку, використовуючи контейнер Spring's IoC (Інверсія управління) для управління об'єктами Моделі та вбудовування їх у Контролери.

Після того, як додаток налаштовано, фреймворк MVC від Spring буде керувати маршрутизацією запитів до відповідного контролера, а також автоматично керуватиме рендерингом представлень (Views). IoC-контейнер Spring керуватиме створенням та ін'єкцією об'єктів моделі у контролери, дозволяючи зосередитися на бізнес-логіці додатка.

В цілому, MVC фреймворк Spring забезпечує надійну та гнучку реалізацію патерну MVC для створення веб-додатків на Java. Використовуючи Spring, варто скористатися такими потужними функціями, як ін'єкція залежностей, AOP (Аспектно-орієнтоване програмування) та управління транзакціями, що в поєднані з дотриманням головних принципів SOLID дозволяє створювати складні, масштабовані та підтримувані веб-додатки.

Важливо пам'ятати, що при роботі з архітектурою MVC в Java компоненти повинні бути слабко пов'язані між собою. Це означає, що кожен компонент повинен бути незалежним від інших, і зміни в одному компоненті не повинні вимагати змін в інших. Щоб досягти цього, потрібно використовувати інтерфейси та слухачі подій, щоб дозволити компонентам взаємодіяти один з одним, не маючи прямих залежностей.

Висновки

Для ефективного використання патерну MVC в Java важливо розуміти ключові концепції патерну, в тому числі ролі та обов'язки компонентів Model, View і Controller, а також механізми зв'язку між ними. Також важливо використовувати найкращі практики для проектування та реалізації Java-додатків на основі MVC, такі як використання інтерфейсів та слухачів подій для досягнення вільного зв'язку між компонентами.

Загалом, патерн MVC є потужним та гнучким архітектурним патерном, який можна адаптувати до широкого спектру випадків використання Java. Використовуючи MVC, розробники можуть створювати модульні, масштабовані та підтримувані додатки, що дозволяє їм зосередитися на створенні високоякісного програмного забезпечення, яке відповідає потребам користувачів.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Advantages of Using MVC in Application Development - fireup.pro. IT engineers. We develop custom software to boost your business - fireup.pro. URL: https://fireup.pro/blog/advantages-of-using-mvc-in-application-development.
2. MVC Framework - Introduction. Online Courses and eBooks Library. URL: https://www.tutorialspoint.com/mvc_framework/mvc_framework_introduction.htm
3. Spring MVC Tutorial - javatpoint. www.javatpoint.com. URL: https://www.javatpoint.com/spring-mvc-tutorial.

Кучерявий Ігор Володимирович студент групи 3ПІ-22м, факультет інформаційних технологій та комп'ютерної інженерії, Вінницький національний технічний університет, м. Вінниця, e-mail: kucherjavyj228@gmail.com

Ліщинська Людмила Броніславівна – д-р техн. наук, професор, професор кафедри програмного забезпечення, Вінницький національний технічний університет, м. Вінниця, e-mail: llb@vntu.edu.ua

Kucheriavyi Ihor, student of group 3PI-22m, Faculty for Information Technologies and Computer Engineering, Vinnytsia National Technical University, Vinnytsia, e-mail: kucherjavyj228@gmail.com

Lishchynska Lyudmyla Bronislavivna – Dr. Sc. (Eng.), Full Professor, Professor of Program Engineering, Vinnytsia National Technical University, Vinnytsia, e-mail: llb@vntu.edu.ua