

Е.Д. Волкова, А.В. Стук

*Витебский государственный университет им. П.М. Машерова,
г. Витебск, Беларусь*

РАЗРАБОТКА КОНСОЛЬНОГО ПРИЛОЖЕНИЯ ДЛЯ ДЕМОНСТРАЦИИ РАБОТЫ ПАТТЕРНОВ DEPENDENCY INJECTION

На сегодняшний день существует разнообразие фреймворков, которые облегчают работу разработчиков: позволяют создавать более модульный, тестируемый и обслуживаемый код. Примерами фреймворков могут стать Spring Framework, Google Guice, Vue, Ember и другие. Однако многие основываются на паттерне внедрения зависимости, который предоставляет требуемые зависимости объекту вместо создания их внутри самого себя.

Целью работы является исследование паттерна инверсии управления и внедрения зависимостей, а также разработка консольного приложения для демонстрации работы данных паттернов.

В качестве исходных данных использовался открытый код фреймворка Spring Framework. Основным методом исследования является разработка консольного приложения, который будет находить применения для обучения студентов паттерну внедрения зависимостей и основам работы с Spring Framework. Применение IoC и DI способствует снижению степени связанности между модулями системы, что упрощает сопровождение и расширение приложений [1].

Сначала обозначим, что такое паттерны проектирования и для чего они нужны:

1. Для решения типичной проблемы, которая часто встречается при проектировании программ.
2. Паттерны не являются готовым решением, которое можно сразу скопировать в свой код. Это общее описание решения проблемы, которое после некоторой разработки можно реализовать в разных ситуациях.

3. Существуют различные группы паттернов, такие как порождающие, структурные и поведенческие.

Рассмотрим паттерн инверсии управления и внедрения зависимостей. Инверсия управления является общей характеристикой фреймворком, потому что они используют инверсию контроля за объектами и состояниями. Это означает, что компоненты должны зависеть от абстракций других компонентов и не несут ответственности за обработку и создание зависимых объектов. Вместо этого экземпляры объектов предоставляются во время выполнения контейнеров IoC посредством внедрения зависимостей.

Внедрение зависимостей – это метод передачи зависимостей в конструктор объекта. Если необходимый объект был загружен из контейнера IoC, то его зависимости самостоятельно будут предоставлены контейнером. Это позволяет избегать использования зависимостей посредством создания экземпляра вручную, что приводит к уменьшению связи и предоставляет более обширный контроль над временем жизни объектов.

Рассмотрим функционал приложения, которое демонстрирует работу паттерна внедрения зависимостей:

1. Инициализация контекста. Данная команда инициализирует контекст приложения, что включает в себя создание контекста приложения, а также настройку всех компонентов, объявленных в модуле внедрения зависимостей.

2. Получение бина и его зависимостей. После инициализации контекста пользователь может запросить конкретный компонент (бин) и вывести информацию о нем и его зависимостях. Если контекст не инициализирован, будет выведено сообщение о необходимости выполнения первой команды.

3. Вывод всех бинов. Пользоваться может посмотреть список доступных компонентов в контексте. Это позволяет ознакомиться со всеми зарегистрированными бинами и их текущим состоянием.

4. Вывод результата аннотации Value, который служит для определения компонента. Это демонстрирует управление конфигурацией через внешние файлы.

Весь этот функционал может быть подробно исследован с помощью инструмента “Debug”, что позволяет понять, как начинает свою работу Spring приложение.

Заключение: использование паттерна внедрение зависимостей и инверсии управления, который написан самостоятельно или реализован уже в таких фреймворках, таких как Spring и Google Guice, позволяя создавать слабо связанные и легко тестируемые компоненты. Существует несколько способов внедрения зависимостей, включая внедрение через конструктор, методы и свойства, что обеспечивает гибкость при настройке зависимостей объектов [2].

Литература

1. Применение инверсии управления для снижения степени связанности модулей в подсистеме учёта сельскохозяйственных животных за счёт внедрения зависимостей. URL: <https://cyberleninka.ru/article/n/primenenie-inversii-upravleniya-dlya-snizheniya-stepeni-svyazannosti-moduley-v-podsisteme-uchyota-selskohozyaystvennyh-zhivotnyh-za> (дата обращения: 09.01.2025).
2. Инверсия управления и внедрения зависимостей. URL: <https://cyberleninka.ru/article/n/inversiya-upravleniya-i-vnedrenie-zavisimostey> (дата обращения: 10.01.2025).