

В нулевом приближении данного метода учитывается однородное магнитное поле, направленное вдоль оси соленоида, которое описывается формулой

$$B_0(t) = \frac{4\pi}{c} nI = \frac{4\pi}{c} nI_0 \cos(\omega t),$$

где  $n$  – число витков на единицу длины соленоида,  $c$  – скорость света в вакууме ( $3 \cdot 10^8$  м/с).

Так как, при применении метода последовательных приближений для начального (нулевого) приближения достаточно грубой начальной оценки, то для примера с соленоидом таким приближением будет являться условие, что электрического поля в нулевом приближении нет, т.е.  $E_0(t) = 0$ .

Исходя из того, что переменное магнитное поле нулевого приближения является источником вихревого электрического поля, то в первом приближении для нахождения  $\vec{E}_1$  и  $\vec{B}_1$  используем сумму данных нулевого приближения и первых поправок  $\delta$  к ним

$$\begin{aligned} \vec{E}_1 &= \vec{E}_0 + \delta \vec{E}_1 \\ \vec{B}_1 &= \vec{B}_0 + \delta \vec{B}_1 \end{aligned}$$

Таким образом, значения  $E_1$  для первого приближения можно вычислить с помощью уравнения Максвелла. Запишем его в интегральной форме [3]

$$\oint_l E_l dl = -\frac{1}{c} \cdot \frac{\partial}{\partial t} \int_S (\vec{B}, d\vec{S}) \Rightarrow E_1(t) \cdot 2\pi r = -\frac{1}{c} \cdot \frac{\partial}{\partial t} (B_0(t) \cdot S)$$

Подставив в данное уравнение  $B_0(t)$ , выразим  $E_1(t)$

$$E_1(t) = \frac{2\pi r n I_0 \omega \sin(\omega t)}{c^2}, \text{ где } r < R \text{ (внутри соленоида),}$$

$$E_1(t) = \frac{2\pi R^2 n I_0 \omega \sin(\omega t)}{c^2 r}, \text{ где } r \geq R \text{ (снаружи соленоида).}$$

Во втором приближении первая поправка электрического поля  $\delta \vec{E}_1 = \vec{E}_1$  является источником второй поправки  $\delta \vec{B}_2$ , которую и прибавляем к магнитному полю нулевого приближения  $\vec{B}_0$  (рисунок 2б).

$$\oint_l (\delta B_2)_l dl = -\frac{1}{c} \cdot \frac{\partial}{\partial t} \int_S (\vec{E}_1, d\vec{S})$$

Так как, в данной задаче не выполняется условие квазистационарности ( $r \ll \lambda$ , где  $r$  – размер области с полем,  $\lambda$  – длина волны поля,  $\lambda = \frac{c}{\nu}$ , где  $\nu$  – частота изменения электромагнитного поля), то вторая поправка расходится при  $r \rightarrow \infty$ .

Аналогичным образом можно найти поправки для любого приближения.

**Заключение.** Применение метода численных приближений при расчете электронно-оптических систем способно ускорить процесс создания новых электронно-лучевых технологий.

1. Ильин, В.П. Численные методы решения задач электрофизики / В.П. Ильин. – М.: Наука, 1985. – 336 с.
2. Молоковский, С.И. Интенсивные электронные и ионные пучки / С.И. Молоковский, А.Д. Сушков. – Л.: Энергия. 1972.
3. Ландау Л.Д., Лившиц Е.М. Теоретическая физика: Учеб. пособие. В 10т. Т. II. Теория поля. – 7-е изд., испр. – М.: Наука. Гл. ред. физ.-мат. лит., 1988. – 512 с.

## РАСКРЫТИЕ ВОЗМОЖНОСТЕЙ ПАТТЕРНА ИНВЕРСИИ УПРАВЛЕНИЯ И ВНЕДРЕНИЯ ЗАВИСИМОСТЕЙ

**Стук А.В.,**

*студентка 3 курса ВГУ имени П.М. Машерова, г. Витебск, Республика Беларусь  
Научный руководитель – Волкова Е.Д., канд. физ.-мат. наук*

Ключевые слова. Архитектурные паттерны, паттерны проектирования, инверсия управления, внедрение зависимостей.

Keywords. Architectural patterns, design patterns, inversion of control, 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.

**Материал и методы.** В качестве материала в исследовании были использованы статьи IT-специалистов, официальные интернет-ресурсы, документации по технологиям, были применены такие методы исследования, как изучение и сравнительный анализ.

**Результаты и их обсуждение.** Сначала обозначим, что такое паттерны проектирования и для чего они нужны. Паттерны проектирования – это типичный способ решения какой-либо часто встречающейся проблемы, возникающей при проектировании программ [1]. 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. 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 [2].

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 избежать использования зависимостей посредством создания экземпляра вручную, что приводит к уменьшению связи и предоставляет более обширный контроль над временем жизни объектов [2].

Рассмотрим функционал приложения, которое продемонстрирует работу паттерна внедрения зависимостей:

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

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.

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

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 внедрения зависимостей и инверсии управления позволяет облегчить и оптимизировать разработку приложения.

1. Гамма, Э. Паттерны объектно-ориентированного проектирования / Э. Гамма, Р. Хелм, Р. Джонсон, Дж. Влассидес. – СПб. : Питер, 2021. – 448 с.

2. Фримен, Э. Head First. Паттерны проектирования / Э. Фримен, Э. Робсон, К. Сьерра, Б. Бейтс. – Издательский дом «Питер», 2021. – 640 с.