УДК 004.4

ПОНИМАЯ ВНЕДРЕНИЕ ЗАВИСИМОСТЕЙ. ПРИНЦИПЫ, ПРАКТИКИ И ПАТТЕРНЫ DI И IOC.

Калибров И. (Университет ИТМО, Санкт-Петербург)

Данная работа посвящена внедрению зависимостей и принципу инверсии контроля. Представлены и проанализированы методики рефакторинга сильного связанного кода с помощью DI, техники работы с зависимостями для статически-типизированных объектно-ориентированных языков, паттерны и анти-паттерны принципа инверсии контроля и внедрения зависимостей, в частности.

Введение.

“Внедрение зависимостей” – это краеугольный камень объектно-ориентированного программирования. Впервые понятие “внедрение зависимостей” было использовано в статье Мартина Фаулера “Inversion of Control Containers and the Dependency Injection Pattern”. Позже оно было включено Робертом Мартином в 5 основных принципов объектно-ориентированного проектирования, которые мы сегодня знаем как SOLID. Однако, внедрение зависимостей – это один из тех концептов ООП, которые чаще других бывают неправильно или неполностью поняты. Какие преимущества дает внедрение зависимостей? В чем отличие DI от Service Location? Необходимо ли использование контейнеров для использования DI?

Основная часть.

Чтобы раскрыть тему в данном докладе будет проанализированы практическое применение и суть таких понятий как “зависимость”, “внедрение зависимостей”, “контейнер для внедрения зависимостей”, паттерны и анти-паттерны создания объектов, а также недостатки некоторых реализаций DI-контейнеров.

Начав с основных терминов внедрения зависимостей в приложениях и обозначив основные проблемы, которые решает DI, а также показав отличие между сильно и слабо связным кодом, углубимся в тему рассказав про основные паттерны и анти-паттерны внедрения зависимостей, такие как: сервис локатор, внедрение в конструктор и метод, окружающий контекст.

Закончим же примером конкретной реализации DI-контейнера на примере Autofac. Разберем работу со множеством компонентов и сложными API.

Выводы.

В итоге результаты исследования на тему “внедрения зависимостей” можно будет использовать в написании кода на любом объектно-ориентированном языке. Результаты дадут основу для написания слабо связанного кода, что даст ряд неоспоримых преимуществ. Код будет проще читать, переиспользовать и тестировать, а значит и поддерживать.

Калибров И. (автор) Подпись