

Университет ИТМО
Факультет ФПИ и КТ

АПС

Лабораторная работа 2

Студент:

Ляо Ихун

Гр.Р33131

Преподаватель:

Перл Иван Андреевич

Задача

Из списка шаблонов проектирования GoF и GRASP выбрать 3-4 шаблона и для каждого из них придумать 2-3 сценария, для решения которых могут применены выбранные шаблоны.

Сделать предположение о возможных ограничениях, к которым можем привести использование шаблона в каждом описанном случае. Обязательно выбрать шаблоны из обоих списков.

Выполнение

GoF – Factory Pattern

1. Когда создаем какую-то систему, которая взаимодействует с базой данных, чтобы сделать систему более гибкой, все DAO определяются как интерфейсы и реализуются Hibernate. И мы создаем фабрику, которая создает экземпляры объектов DAO на основе выбора типа базы данных. Различные экземпляры объектов DAO могут быть сгенерированы путем изменения ввода на фабрику DAO, который может быть типом базы данных, url к базе данных и т.д. Таким образом реализуем настраиваемость конкретного метода реализации DAO и повышая гибкость системы.
2. Как лабораторные работы второго семестра, когда мы создаем команды, при помощи которых пользователи взаимодействуют с программой, мы можем создать фабрику, которая автоматически создает разные экземпляры команд на основе ввода пользователя.

Ограничение:

Если количество команд достаточно большое, то класс Factory тоже будет 'большой'.

GoF – Singleton Pattern

1. Когда мы пишем программу, которая будет выполнять многопоточные операции с файлом, мы можем использовать этот шаблон проектирования и synchronized, чтобы обеспечить безопасность операций с файлами.

Ограничение:

Если количество файл достаточно большое, то количество классов тоже будет большим из-за того, что каждый класс только представляет только один файл.

2. Компьютер имеет несколько принтеров, но только один PrinterSpooler, чтобы избежать одновременного вывода на принтер двух заданий печати.

Ограничение:

Если то printer spooler не работает, то нет возможности печати.

GoF – Strategy Pattern

1. При разработке платежной функции будут разные методы реализации для разных способов оплаты. В том можно использовать strategy.

Ограничение:

Один способ оплаты соответствует одному классу. Тем больше способ оплаты, тем больше класс.

2. Когда создают программное обеспечение для заказа такси, такое как yandexgo, по-разному обрабатывают разные типы заказов. Например, изменится отображение на карте, изменится метод расчета цены и так далее. В этом случае также можно использовать паттерн strategy.

Ограничение:

Как и в первом случае, стратегий будет слишком много. Нам необходимо использовать смешанные шаблоны проектирования для решения этой проблемы.

Grasp – Polymorphism

1. Предположим, у нас есть бизнес-система, поддерживающая бизнес более чем в 100 странах. Тогда у нас есть более 100 налогов для обработки в системе. Мы можем создать интерфейс TaxDealer. Затем создаем один класс для каждой страны, чтобы наследует интерфейс.

Ограничение:

Если мы хотим получить метод расчета налогов для каждой страны, то нам нужно читать все классы.

2. Когда мы пишем программу, которая может рисовать разные фигуры, мы сначала определяем интерфейс с помощью метода рисования. Затем всякий раз, когда мы хотим, чтобы программа рисовала еще одну новую фигуру, нам просто нужно создать еще один класс, который наследует тот интерфейс.

Ограничение:

Если количество фигур достаточно большое, то класс Factory тоже будет 'большой'.

Вывод

Шаблоны проектирования могут помочь нам решить проблемы фиксированного типа и написать более качественный и читаемый код. GRASP является предпосылкой шаблона проектирования GoF, а шаблон проектирования GoF — это объектно-ориентированный шаблон проектирования, который соответствует требованиям принципов шаблона GRASP.