

ПРИНЦИПЫ SOLID

Принципы SOLID [1]:

- SRP: Single Responsibility Principle – принцип единственной ответственности;
- OCP: Open-Closed Principle – принцип открытости/закрытости;
- LSP: Liskov Substitution Principle – принцип подстановки Барбары Лисков.
- ISP: Interface Segregation Principle – принцип разделения интерфейса).
- DIP: Dependency Inversion Principle – принцип инверсии зависимости.

Цель принципов – создать программные структуры среднего уровня, которые:

- терпимы к изменениям;
- просты и понятны;
- образуют основу для компонентов, которые могут использоваться во многих программных системах.

Принцип единственной ответственности. Традиционно принцип единственной ответственности описывается так: модуль должен иметь одну и только одну причину для изменения.

Принцип открытости / закрытости. Программные сущности должны быть открыты для расширения и закрыты для изменения.

Принцип подстановки Барбары Лисков. Принцип в формулировке Роберта Мартина декларирует, что функции, которые используют базовый тип, должны иметь возможность использовать подтипы базового типа, не зная об этом. Простое нарушение совместимости может вызвать загрязнение архитектуры системы значительным количеством дополнительных механизмов.

Принцип разделения интерфейсов. Происхождение названия принципа разделения интерфейсов наглядно иллюстрирует схема на рисунке 1. В данной ситуации имеется несколько классов, пользующихся операциями в классе OPS. Допустим, что User1 использует только операцию op1, User2 – только op2 и User3 – только op3.

Представим, что OPS – это класс, написанный на таком языке, как Java. Очевидно, что в такой ситуации исходный код User1 непременно будет зависеть от op2 и op3, даже при том, что он не поль-

зуются ими. Эта зависимость означает, что изменения в исходном коде метода `op2` в классе `OPS` потребуют повторной компиляции и развертывания класса `User1`, несмотря на то, что для него ничего не изменилось.

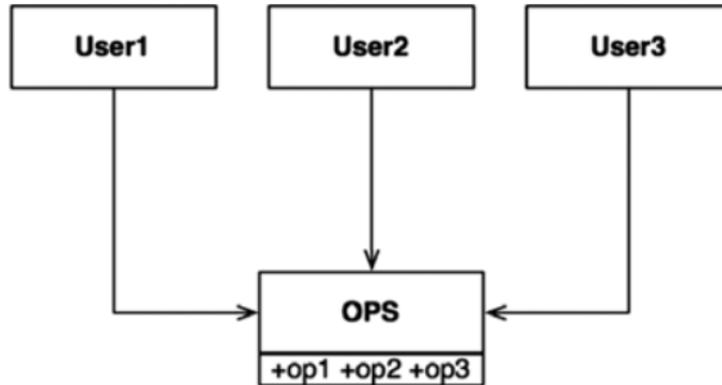


Рисунок 1 – Принцип разделения интерфейса

Эту проблему можно решить разделением операций по интерфейсам, как показано на рисунке 2 [2].

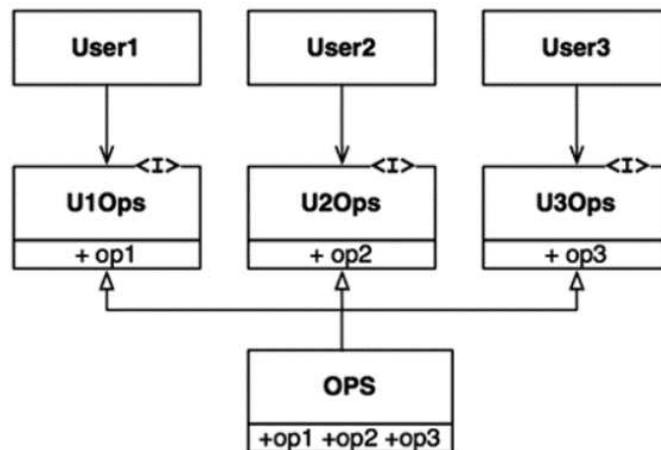


Рис. 10.2. Разделение операций

Рисунок 2 – Принцип операций

Принцип инверсии зависимости. Принцип инверсии зависимости утверждает, что наиболее гибкими получаются системы, в которых зависимости в исходном коде направлены на абстракции, а не на конкретные реализации. Мы должны избегать зависимости от неустойчивых конкретных элементов системы. То есть от модулей, которые продолжают активно разрабатываться и претерпевают частые изменения. Рассмотренные принципы позволяют строить на базе объектно-ориентированного программирования масштабируемые и сопровождаемые программные продукты с понятной бизнес-логикой.

ЛИТЕРАТУРА

1. Роберт, М. Чистая архитектура. Искусство разработки про-

граммного обеспечения. – СПб.: Питер, 2018.

2. Принципы SOLID [Электронный ресурс] / Medium – Режим доступа: <https://medium.com/webbdev/solid-4ffc018077da>. – Дата доступа: 15.04.2021.

УДК 004.9

Студ. Е.А. Устименко
Науч. рук. доц. Р.О. Короленя
(кафедра информатики и веб-дизайна, БГТУ)

ОСОБЕННОСТИ САЙТОВ В НИШЕ ЛАНДШАФТНОГО ДИЗАЙНА

Человек активно преобразует окружающий мир, изменяя его внешний облик в зависимости от своих представлений о гармонии. Ландшафтный дизайн – это комплекс специальных мероприятий и решений по благоустройству территории, направленный на изменение внешнего вида территории путем активного использования природных компонентов.

Для определения особенностей сайтов данной тематики было выбрано несколько сайтов, как фирм постсоветского пространства, так и зарубежные веб-сайты: www.fitonia.by; www.agrostis.by; www.landshaft-design.kiev.ua; www.april-studio.by; www.hargreaves.com.

Анализ сайтов проводился по следующим критериям:

- структура;
- функционал;
- контент;
- дизайн;
- цветовое решение.

Структура сайта – это схема размещения его основных разделов и страниц относительно друг друга. В зависимости от типа сайта выделяются несколько видов структур. При анализе структуры сайтов в нише ландшафтного дизайна была выделена четкая иерархия страниц, что позволяет сделать вывод о применении древовидной структуры.

Основными разделами в структуре сайтов рассматриваемой тематики являются «*О нас*», «*Портфолио/Работы/Проекты*» и «*Контакты*». В структуре русскоязычных сайтов также встречаются разделы «*Услуги*» и «*Цены*».

Функционал – это то, с чем пользователь может взаимодействовать на сайте. Перечень функциональных возможностей достаточно велик.