

Для торговой марки Iasconflex используются следующие функционал и возможности:

- каталог товаров: позволяет создавать каталоги на различные категории товаров, имеет гибкие настройки и товары можно упорядочить по цене алфавиту и новизне. Также для каждого товара автоматически создается уникальная страница, которая индексируется поисковыми системами;

- блочных механизм редактирования: механизм позволяет без вреда для других блоков редактировать и не беспокоиться, что редактирование затронет остальную часть страницы. Есть возможность удобно создавать каждый блок под различные устройства пользователей;

- редактор собственных блоков и HTML-кодов: предлагает Zero-блок в можно создавать собственный дизайн страницы и вносить добавлять свой код на страницу. Таким образом, расширяя возможности создания страниц.

Tilda имеет широкий инструмент возможности, позволяет выполнять различные задачи, обеспечивает комфортную работу и простоту пользования. Конструктор не требует от пользователя специальных знаний.

Tilda представляет собой инновационную платформу для создания веб-сайтов, которая объединяет в себе простоту использования и гибкость настройки. С помощью Tilda вы можете создавать не только статические страницы, но и интерактивные лендинги, блоги, интернет-магазины и даже многостраничные сайты.

Одной из ключевых особенностей Tilda является её блочная система. Вы можете создавать свой сайт, выбирая из библиотеки предварительно разработанных блоков, каждый из которых можно настроить под свои нужды. Благодаря этому, даже без знаний программирования, вы можете создать профессионально выглядящий сайт.

Кроме того, предлагает интеграцию с множеством платежных систем и сервисов, что позволяет вам создавать функциональные интернет-магазины и собирать платежи прямо на вашем сайте.

Tilda – это мощный и гибкий инструмент для создания веб-сайтов, который подойдет как новичкам, так и опытным веб-разработчикам.

Выше описанный функционал был использован для создания веб-сайта поскольку он обеспечивают высокую производительность, удобство в обращении и гибкость в производстве.

А. А. Марочкин

(ГГУ имени Ф. Скорины, Гомель)

Науч. рук. **О. М. Дерюжкова**, канд. физ.-мат. наук, доцент

ИСПОЛЬЗОВАНИЕ ПАТТЕРНОВ BUILDER И SINGLETON

Строитель (Builder) – шаблон проектирования, который инкапсулирует создание объекта и позволяет разделить его на различные этапы [1]. Шаблон Builder можно разбить на следующие важные компоненты [2]:

1. **Product (продукт)** – класс, который определяет сложный объект, необходимый шаг за шагом сконструировать, используя простые объекты.

2. **Builder (строитель)** – абстрактный класс/интерфейс, который определяет все этапы, необходимые для производства сложного объекта-продукта. Как правило, здесь объявляются (абстрактно) все этапы (buildPart), а их реализация относится к классам конкретных строителей (ConcreteBuilder).

3. **ConcreteBuilder (конкретный строитель)** – класс-строитель, который предоставляет фактический код для создания объекта-продукта. У нас может быть несколько разных ConcreteBuilder-классов, каждый из которых реализует различную разновидность или способ создания объекта-продукта.

4. **Director (распорядитель)** – супервизионный класс, под контролем которого строитель выполняет скоординированные этапы для создания объекта-продукта. Распорядитель обычно получает на вход строителя с этапами на выполнение в четком порядке для построения объекта-продукта (рисунок 1).

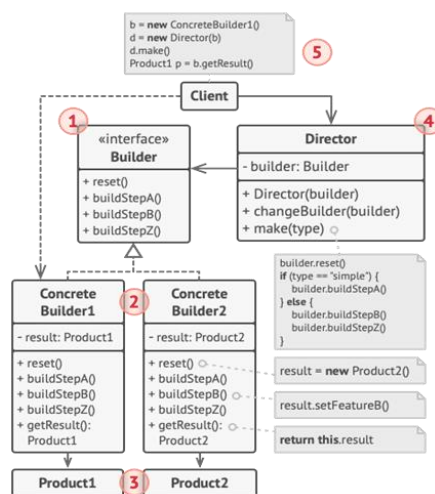


Рисунок 1 – Визуальное изображение паттерна Builder

Одиночка (Singleton) – это порождающий паттерн проектирования, который гарантирует, что у класса есть только один экземпляр, и предоставляет к нему глобальную точку доступа [1].

Одиночка решает сразу две проблемы, нарушая принцип единственной ответственности класса:

1. Наличие единственного экземпляра класса. Чаще всего это полезно для доступа к какому-то общему ресурсу, например, базе данных. Если через некоторое время после создания объекта требуется получить точку доступа к классу, то в этом случае лучше использовать старый объект, вместо создания нового. Такое поведение невозможно реализовать с помощью обычного конструктора, так как конструктор класса всегда возвращает новый объект [3].

2. Предоставляет глобальную точку доступа. Это не просто глобальная переменная, через которую можно получить доступ к определённому объекту, а глобальная переменная, защищенная от записи. Следовательно, внешний код не может подменять их значения напрямую.

Основным достоинством паттерна Одиночка является простой и удобный интерфейс, а также гарантированное хранение одного экземпляра нужного класса (рисунок 2).

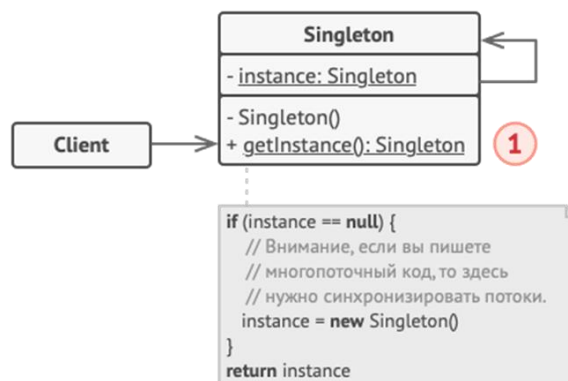


Рисунок 2 – Визуальное изображение паттерна Singleton

Все реализации шаблона Одиночка сводятся к тому, чтобы скрыть конструктор по умолчанию и создать публичный статический метод, который и будет контролировать жизненный цикл объекта-одиночки. Если есть доступ к классу одиночке, следовательно, будет доступ и к этому статическому методу. При вызове из любой точки, он всегда будет возвращать один и тот же объект [3].

Таким образом, паттерны Builder и Singleton упрощают программы и делают код более понятным и читаемым, разбивая ответственность и права на различные объекты, которые содержат свои поля и методы.

Литература

1. Гамма, Э. Паттерны объектно-ориентированного проектирования. Серия «Библиотека программиста» / Э. Гамма, Р. Хелм, Р. Джонсон, Дж. Влссидес. – СПб. : Питер, 2020 – 448 с. – ISBN 978-5-4461-1595-2.

2. Builder pattern [электронный ресурс]. – 2023. – URL: <https://refactoring.guru/ru/design-patterns/builder> . – Дата доступа: 28.01.2024.

3. Singleton pattern [электронный ресурс]. – 2023. – URL: <https://refactoring.guru/ru/design-patterns/builder>. – Дата доступа: 12.02.2024.

А. В. Маршалов

(ГГУ имени Ф. Скорины, Гомель)

Науч. рук. **Н. А. Аксенова**, ст. преподаватель

РАЗРАБОТКА ДОПОЛНЕННОЙ РЕАЛЬНОСТИ

Дополненная реальность (AR) – это результат введения в зрительное поле любых сенсорных данных с целью дополнения сведений об окружении и изменения восприятия окружающей среды. Дополненная реальность представляет собой воспринимаемую смешанную реальность, где реальные объекты дополняются наложенной на них информацией.

Примеры дополнения воспринимаемой реальности включают:

- параллельные цветные линии на дороге, показывающие направления движения;
- стрелки, указывающие расстояние от места поворота до конечного пункта;
- оживление любой картинки;
- смешение реальных и вымышленных объектов в кинофильмах и компьютерных играх.

Применение и преимущества:

– улучшение пользовательского опыта: AR предоставляет дополнительную информацию и интерактивный контент, улучшая восприятие пользователем окружающей среды;

– интеграция с реальным миром: в отличие от виртуальной реальности (VR), которая создает полностью симулированную среду, AR интегрирует виртуальные элементы в реальное окружение;

– применение в разных областях: AR используется в учебном моделировании, архитектурной визуализации, медицине, игровой индустрии и других областях.

Дополненная реальность – это увлекательная технология, которая меняет способ, которым мы взаимодействуем с окружающим миром. Она позволяет нам видеть реальный мир с новой перспективой и обогащает наш опыт.