ШАБЛОН ПРОЕКТИРОВАНИЯ ПРОГРАММНЫХ ПРОДУКТОВ: MVC

Курьян Илья Сергеевич

студент, Сибирский государственный индустриальный университет, РФ, г. Новокузнецк

Негина Дарья Вячеславовна

студент, Сибирский государственный индустриальный университет, РФ, г. Новокузнецк

Костырева Софья Андреевна

студент, Сибирский государственный индустриальный университет, РФ, г. Новокузнецк

Качалкова Катерина Игоревна

научный руководитель, преподаватель, Сибирский государственный индустриальный университет, РФ, г. Новокузнецк

В данной статье приведена информация об основных шаблонах проектирования программных продуктов типа MV*. Среди них можно выделить наиболее часто используемые шаблоны такие как MVC, MVVM, MVP. Каждый шаблон проектирования обладает своими отличительными особенностями, которые применяются при разработке программных продуктов.

Шаблон MVC представляет собой схему разделения данных приложения и управляющей логики на три отдельных компонента: модель, представление и контроллер – таким образом, что модификация каждого компонента может осуществляться независимо [1].

Рассмотрим пример использования шаблона проектирования MVC, который будет реализован в примерах программ, предоставляемых в данной статье. В этом примере модель может быть представлена объектом, выполняющим функцию переключателя. Простейший вариант этой модели определяется двумя состояниями: выключено или включено, и также предоставляет возможность их изменения. Объект модели обладает методами для изменения состояния, такими как "выключить" и "включить". Представление отображает текущее состояние переключателя на экране с использованием текстового или графического представления. Например, текстовая метка может отображать "выключено" или "включено" в зависимости от состояния модели. Помимо отображения, представление также позволяет пользователю изменять состояние переключателя через графические элементы, такие как две кнопки с надписями "Включить" и "Выключить". Представление отвечает только за отображение состояния модели, а для изменения состояния оно обращается к контроллеру. Контроллер, в свою очередь, представляет собой объект, способный изменять состояние модели переключателя. Схема шаблона MVC представлена на рисунке 1.

Курьян И.С., Негина Д.В., Костырева С.А. ШАБЛОН ПРОЕКТИРОВАНИЯ ПРОГРАММНЫХ ПРОДУКТОВ: MVC // Студенческий форум: электрон. научн. журн. 2024. № 3(270). URL: https://nauchforum.ru/journal/stud/270/144147 (дата обращения: 07.09.2025).

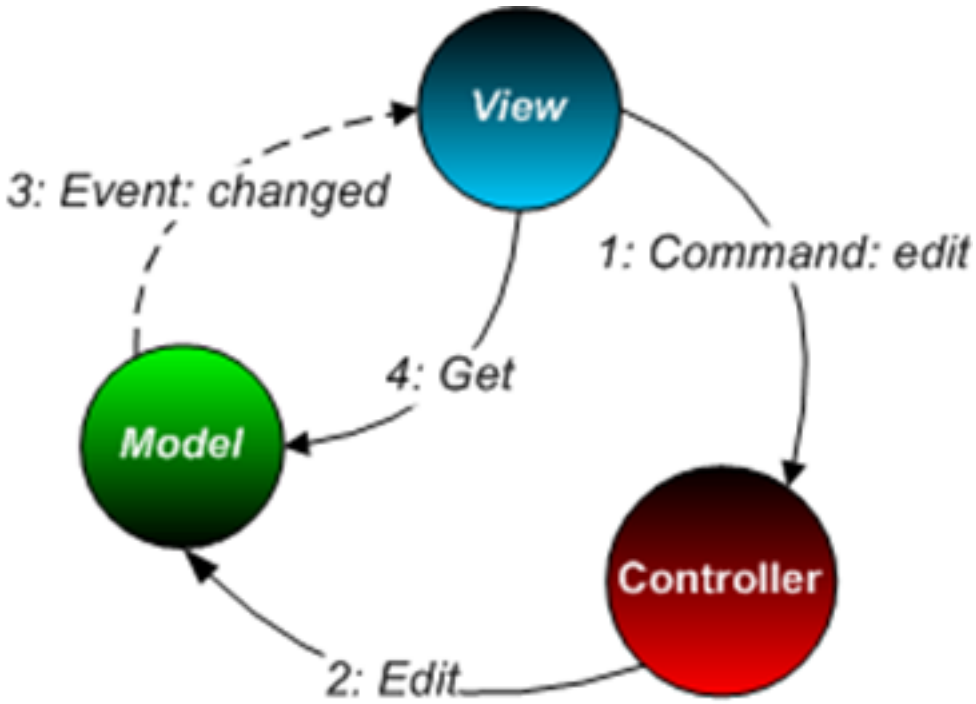

Рисунок 1. Схема шаблона MVC

Весь цикл функционирования данной MVC-триады, включающей модель, представление и контроллер переключателя, может быть описан следующим образом. При инициализации пользователем представление обращается к модели и устанавливает текстовую метку в соответствии с текущим состоянием переключателя. Пользователь инициирует изменение состояния переключателя, нажимая на определенную кнопку. В ответ представление передает соответствующую команду контроллеру: включить или выключить. Контроллер интерпретирует команду и вносит изменения в модель. После этого представление регистрирует изменение в модели, и по этому событию оно обновляет текстовую метку, чтобы отразить новое состояние переключателя.

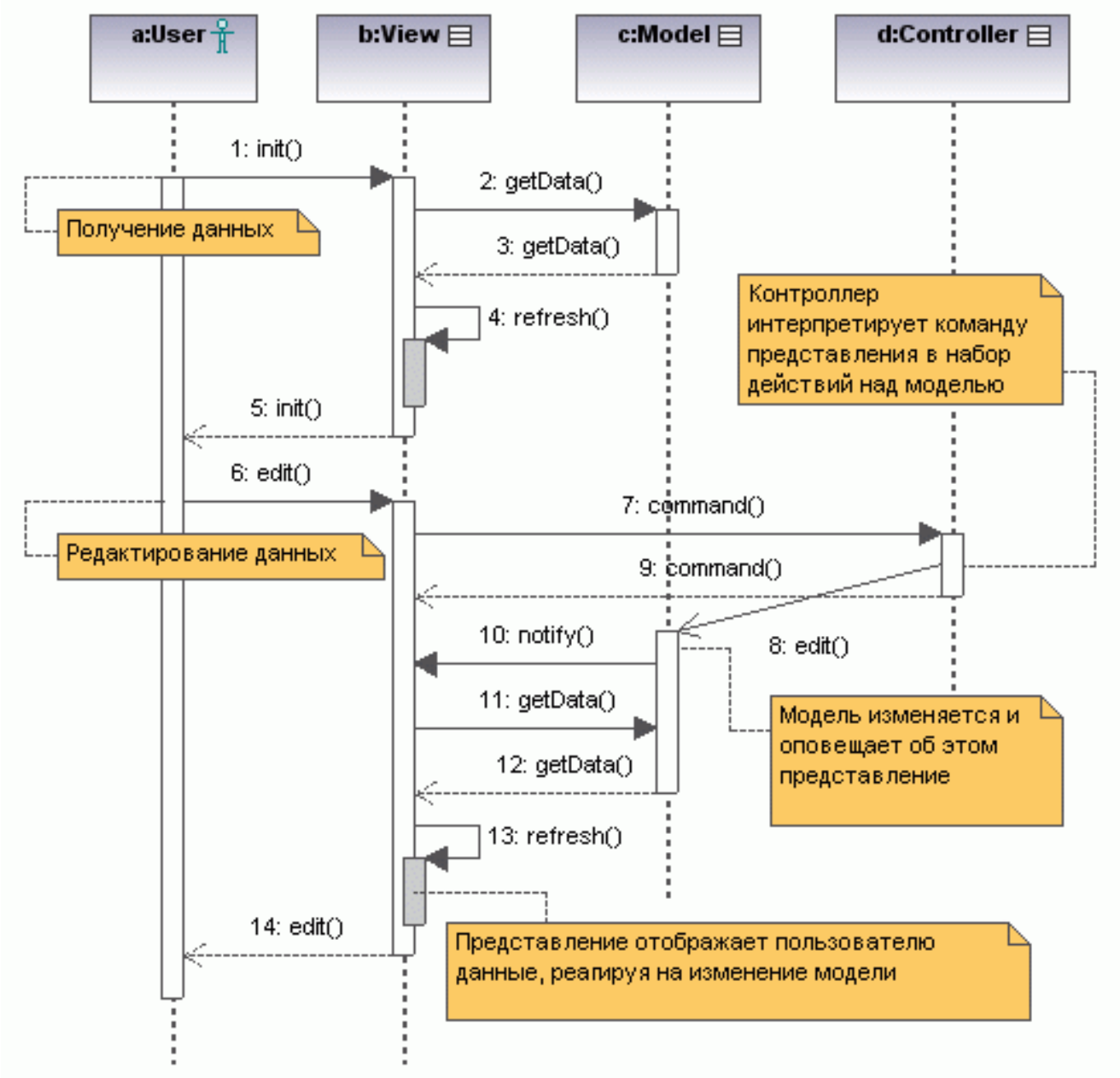

Рисунок 2. Диаграмма последовательности MVC

Назначение шаблона MVC

Основной целью использования данной концепции является разделение бизнес-логики (модели) от ее визуализации (представления, вида) с целью повышения возможности повторного использования кода. Эта концепция особенно полезна в ситуациях, когда необходимо представлять одни и те же данные одновременно в различных контекстах или с разных точек зрения. В частности, она решает следующие задачи:

Модель может быть связана с несколькими видами, не затрагивая реализацию самой модели. Например, данные могут быть представлены одновременно в виде электронной таблицы, гистограммы и круговой диаграммы.

Изменения в реакции на действия пользователя (например, нажатие кнопки мыши или ввод данных) могут быть внесены без изменения реализации видов, достаточно использовать другой контроллер.

Разработчики могут специализироваться только в одной области, либо в графическом интерфейсе, либо в разработке бизнес-логики. Это позволяет программистам, занимающимся разработкой бизнес-логики (модели), оставаться неосведомленными о том, какое конкретное представление будет использоваться.

Концепция паттерна MVC

Модель: Модель предоставляет данные и методы для их обработки, включая запросы в базу данных и проверку на корректность. Она остается независимой от представления (без знания о том, как данные визуализировать) и контроллера (без точек взаимодействия с пользователем), предоставляя лишь доступ к данным и управление ими.

Модель разрабатывается так, чтобы реагировать на запросы, изменяя свое состояние, и может включать уведомления для своих "наблюдателей".

Благодаря своей независимости от визуального представления модель может иметь несколько различных представлений для одной и той же "модели".

Представление: Представление отвечает за получение необходимых данных из модели и их передачу пользователю. Представление не обрабатывает введенные пользователем данные [2].

Контроллер: Контроллер обеспечивает "связь" между пользователем и системой, управляя и направляя данными от пользователя к системе и обратно. Он использует модель и представление для реализации необходимых действий.

Функциональные возможности и расхождения: поскольку у MVC нет строгой реализации, различные варианты могут быть реализованы по-разному. Отсутствует универсальное определение того, где следует размещать бизнес-логику, которая может находиться как в контроллере, так и в модели. В последнем случае модель будет включать все бизнес-объекты с соответствующими данными и функциями.

Некоторые фреймворки строго определяют, где должна располагаться бизнес-логика, в то время как другие не предъявляют таких ограничений.

Также нет четких указаний о том, где должна осуществляться проверка введенных пользователем данных. Простая валидация может происходить даже в представлении, но чаще всего она находится в контроллере или модели.

Интернационализация и форматирование данных также не имеют четких указаний относительно своего расположения.

Для реализации схемы «Model-View-Controller» используется достаточно большое число шаблонов проектирования (в зависимости от сложности архитектурного решения), основные из которых – «наблюдатель», «стратегия», «компоновщик»[3].

Список литературы:

1. Рогачев, С. Обобщенный Model-View-Controller / С. Рогачев // RSDN Magazine. — 2008. — № 4. — 2007 г.

2. Model-View-Controller [Электронный ресурс] // Wikipedia. — Режим доступа: https://ru.wikipedia.org/wiki/Model-View-Controller.

3. Гамма, Э. Приемы объектно-ориентированного проектирования. Пат-терны проектирования / Э. Гамма, Р. Хелм, Р. Джонсон, Дж. Влиссидес. — СПб : Питер, 2001. — 368 с. — (Серия "Библиотека программиста"). — ISBN 5-272-00355-1.