Государственный университет Республики Молдова

Факультет Математики и Информатики

Курс «JAVA для интернета»

Лабораторная работа №1

«Паттерн MVC»

Студент: Негура Денис  
Группа: I2002  
Профессор: G. Latul

Кишинёв 2022

Паттерн MVC:

«Модель — Представление — Контроллер» (MVC) представляет собой паттерн проектирования J2EE.

Основной подход данного паттерна заключается в продвижении в среде разработчиков ПО принципа разделения ответственности. Паттерн MVC разделяет систему на три вида компонентов, причем у каждого компонента системы есть свои конкретные обязанности. Вот эти три вида компонентов:

* Модель. Модель в паттерне MVC отвечает за данные для представления,

с целью дальнейшей их визуализации в одном из шаблонов представления.

Если вкратце, можно сказать, что модель представляет собой объект данных.

* Представление. Представление в паттерне MVC отвечает за визуализацию

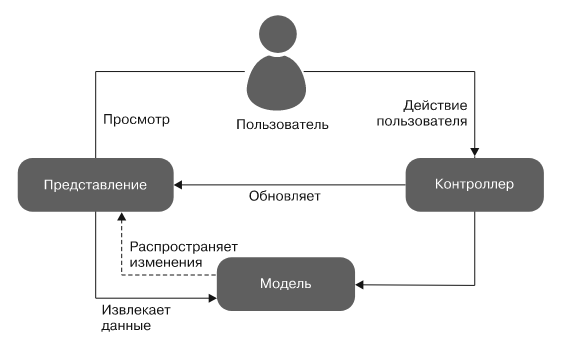
модели в веб-приложении в виде веб-страницы. Оно представляет данные мо-

дели в удобочитаемом для пользователя формате. Для этой цели существует

несколько технологий, например JSP, страницы JSF, PDF, XML и др.

* Контроллер. Это тот компонент паттерна MVC, который фактически производит все действия. Код контроллера управляет взаимодействием между представлением и моделью. Такие взаимодействия, как отправка формы или щелчок на ссылке, являются в корпоративном приложении частью контроллера. Кроме того, контроллер отвечает за создание и обновление модели, а также перенаправление модели представлению для визуализации.

В приложении есть три компонента, у каждого из которых — свои обязанности. Как уже упоминалось, паттерн MVC нацелен на разделение ответственности. В программной системе разделение ответственности играет очень важную роль, обеспечивая гибкость и легкость тестирования компонентов, а также чистую структуру кода. В паттерне MVC пользователь взаимодействует с компонентом «контроллер» посредством компонента «представление», а компонент «контроллер» запускает фактические действия по подготовке компонента «модель». Компонент «модель» распространяет изменения на «представление», и, наконец, компонент «представление» визуализирует модель на экране пользователя.



Принцип работы паттерна MVC

Разделим приложение на компоненты паттерна:

1. **Model:**

Группа классов для работы с данными, включает класс Student – сущность содержащая данные о студентах. StudentRepo – интерфейс позволяющий обращаться к базе данных для чтения, удаления, сохранения и обновления данных. Саму базу данных на СУБД H2 хранящуюся в оперативной памяти.

**2) View:**

Файлы html и css для структурирования и визуализации данных, для предоставления просмотра конечному пользователю.

**3) Controller:**

Класс StudentController отвечает за извлечение данных о студентах и передачу данных представлению через метод addStudents, вызываемый запросом при посещении страницы. За чтение и сохранение данных в методе setStudent, вызываемом при подтверждении в форме представления.