|  |
| --- |
| **МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ** |
| **федеральное государственное бюджетное образовательное учреждение высшего образования «Российский экономический университет имени Г.В. Плеханова»**  **Московский приборостроительный техникум** |

ОТЧЕТ

по учебной практике

УП.04.01 Внедрение и поддержка программного обеспечения\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.

Профессионального модуля ПМ.04 Сопровождение и обслуживание программного обеспечения компьютерных систем .

Специальность 09.02.07 Информационные системы и программирование \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.

Студент Сорокин Дмитрий Максимвич.

(фамилия, имя, отчество)

Группа П50-6-20

Руководитель по практической подготовке от техникума

Серяк Даниил Владимирович.

(фамилия, имя, отчество)

«\_\_» \_\_\_\_\_\_ 2023 года

Оглавление

[Практическая работа №1 3](#_Toc153271504)

Практическая работа №1

Калькулятор и Конвертер валют

Цель: Необходимо создать приложение которое включает в себя страницы:

1) Главная страница , на которой находятся кнопки для перехода на страницу "Калькулятор" и "Конвертер валют"

2) Калькулятор, данная страница выполняет функционал базового калькулятора, но после получения результата, пользователя должно перекинуть на страницу с ответом. Необходимо использовать @PostMapping

3)Конвертер валют, страница на которой находится 2 выпадающих списка: в первом находится валюта из которой надо перевести деньги, а во втором в какую валюту надо перевести.

Основные требования по коду:

1)Использовать @GetMapping и @PostMapping

2)Использовать @RequestParam

Ход работы:

1. Создание фала c html разметкой для страниц

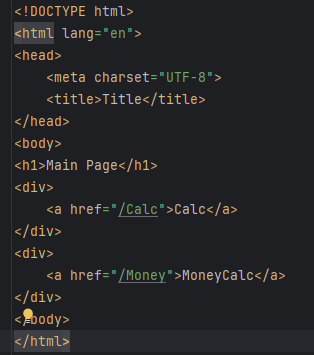


Рисунок 1 Главная страница

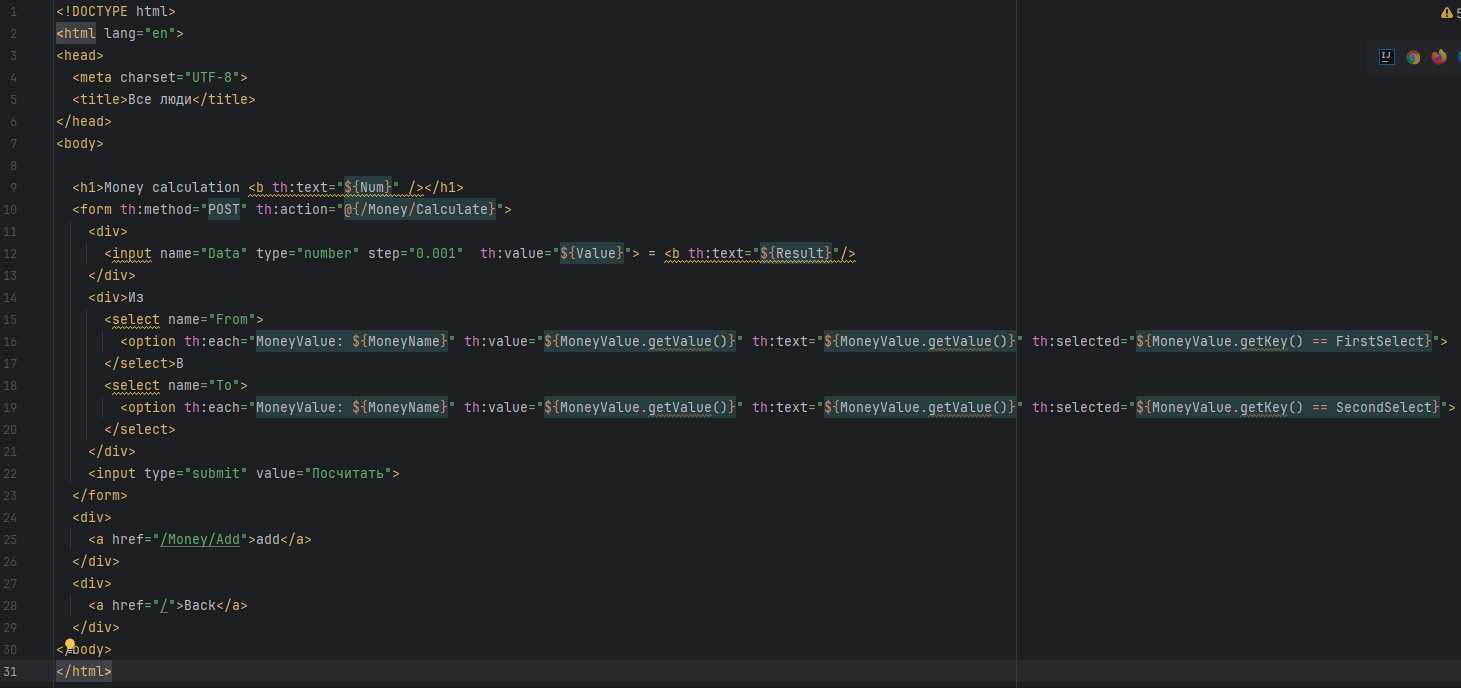


Рисунок 2 Страница перевода валют

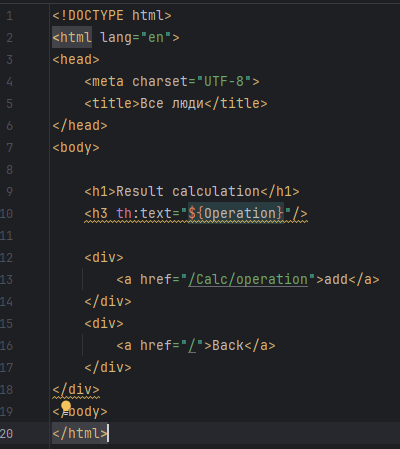


Рисунок 3 Страница с результатов вычислений

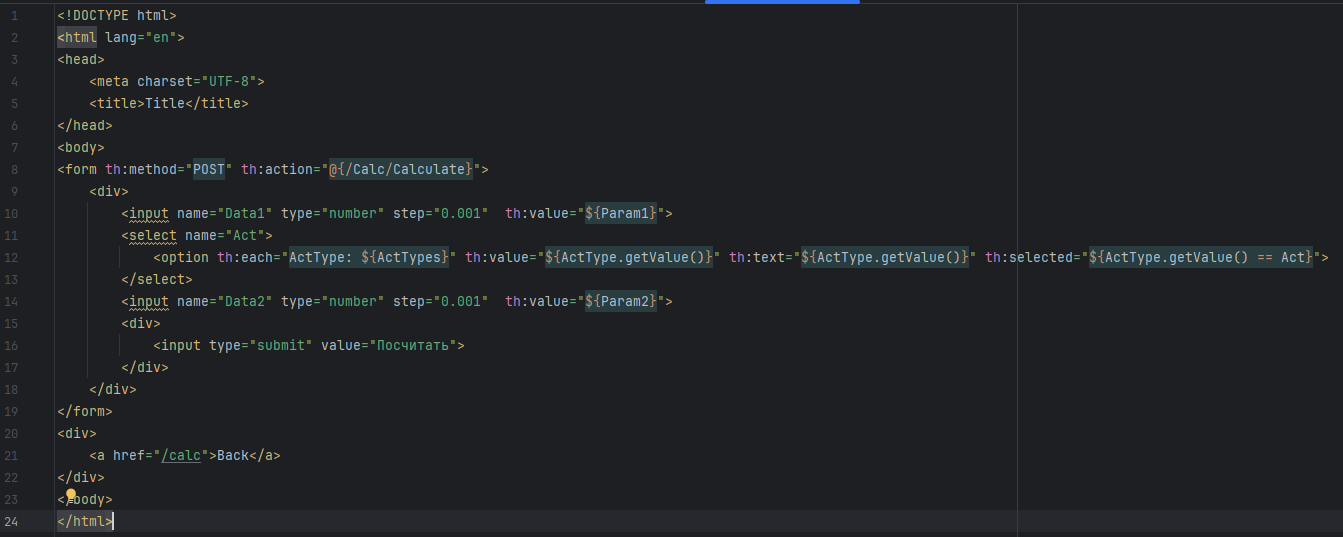


Рисунок 4 Страница указания параметров вычисления

1. Создание контроллеров управляющие переходами между страницами

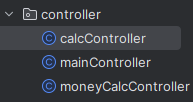


Рисунок 5 Контроллеры



Рисунок 6 Контроллер калькулятора

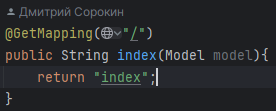


Рисунок 7 Контроллер начального окна

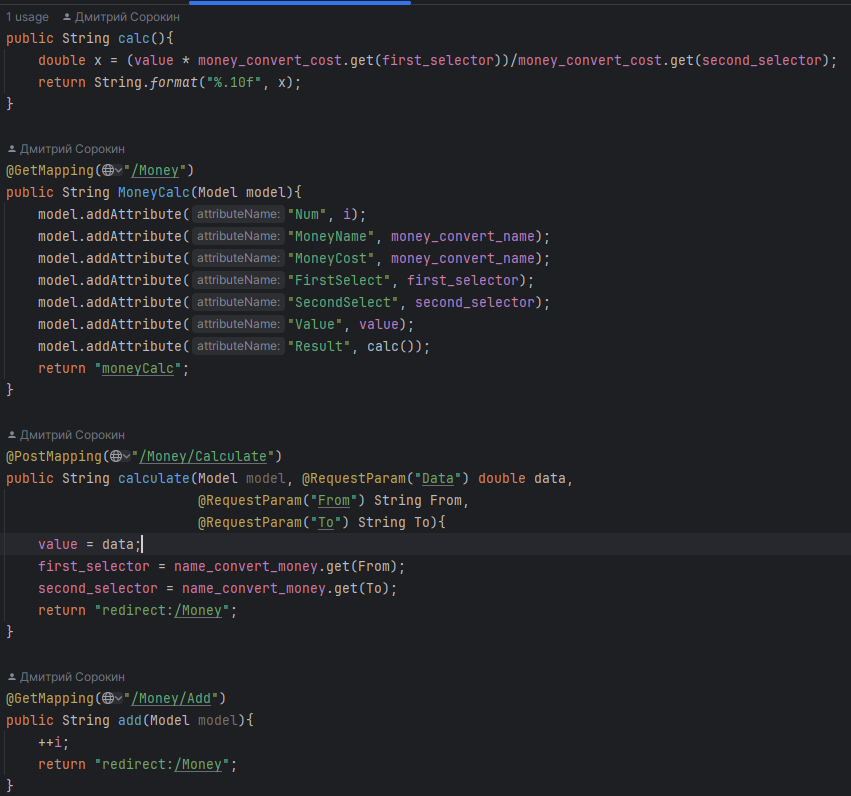


Рисунок 8 Контроллер конвертации валют

1. Результат работы

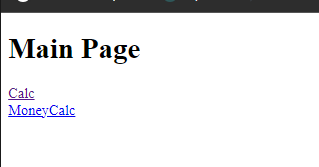


Рисунок 9 Главная страница

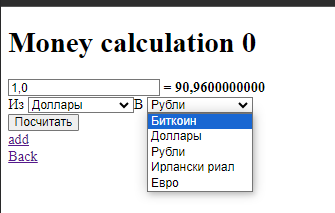


Рисунок 10 Перевод валют

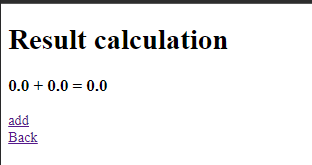


Рисунок 11 Страница вывод ответа калькулятора



Рисунок 12 Страница указания параметров вычисления

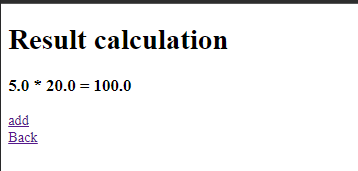


Рисунок 13 Вывод ответа

Вывод: В ходе практической работы №1 было создано приложение включающее в себя калькулятор и перевод валют.

Практическая работа №2

Работа с паттерном DAO

Цель: В данной практической необходимо реализовать приложение, в котором будет реализован паттерн DAO.

Требования:

1) На 3 необходимо создать 5 моделей по 4 поля в каждой, и реализовать базовый паттерн DAO(Повторить действия из примера)

2) На 4 необходимо сделать навигацию по своему сайту. Так же необходимо подтянуть стили на страницы.

Особо углубляться в frontend не обязательно, можете просто подключить Bootstrap

3) На 5 необходимо добавить универсальный класс для обработки CRUD-операций, который будет вызываться для каждой модели(чтобы не было много одинакового кода)

Ход работы:

1. Создание классов



Рисунок 14 Создание абстрактного класса для указания методов которые будут у наследников



Рисунок 15 Пример реализации одного из наследников абстрактного класса

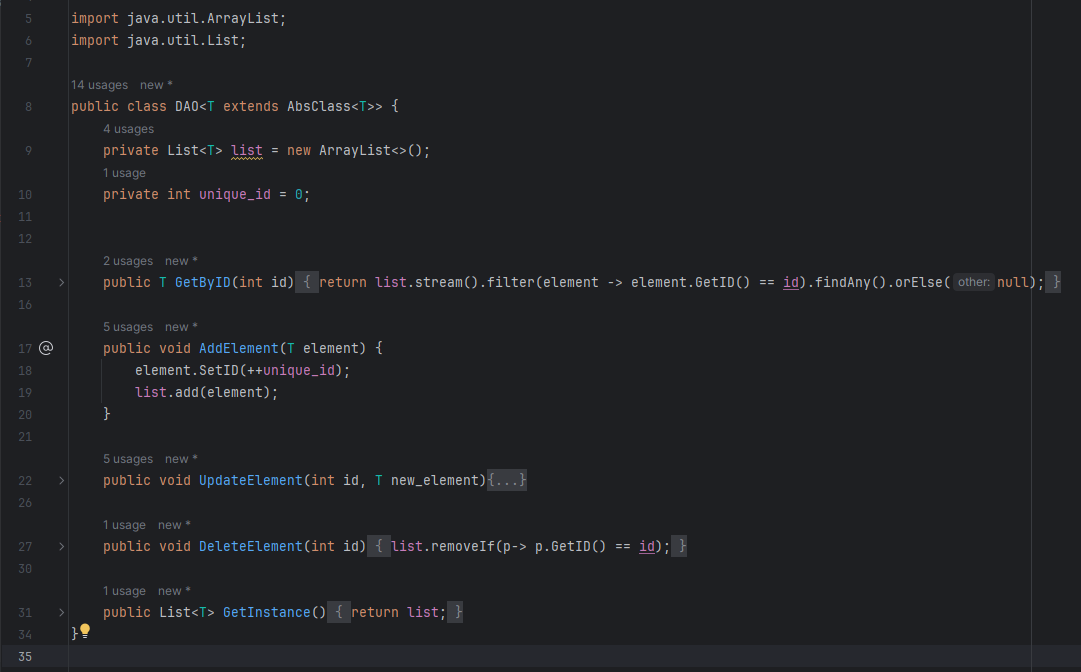


Рисунок 16 Создание шаблонного класса DAO который будет хранить и взаимодействовать с данными

1. Создание контроллеров

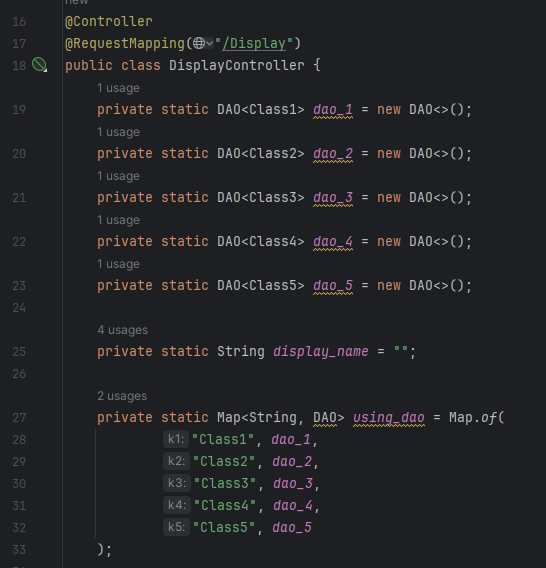


Рисунок 17 Создание переменных типа DAO для указания хранимых дынных

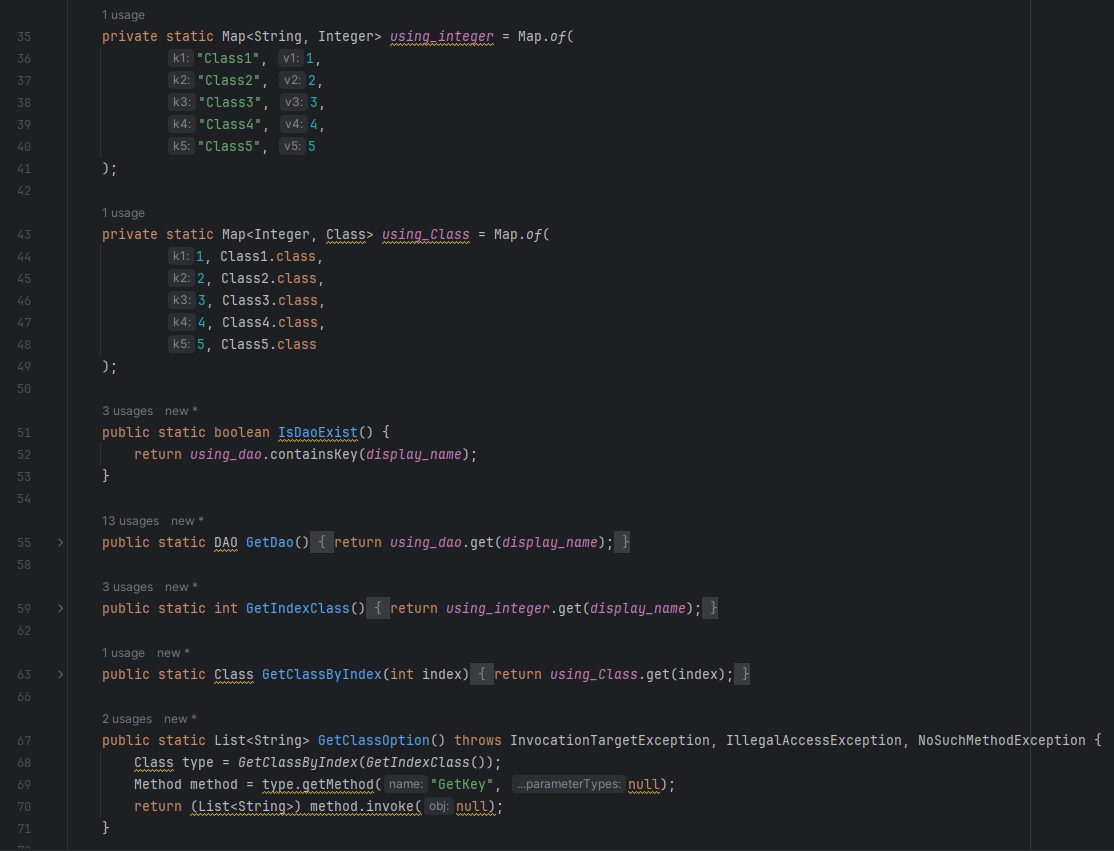


Рисунок 18 Создание статических полей и методов для быстрого взаимодействия с данными

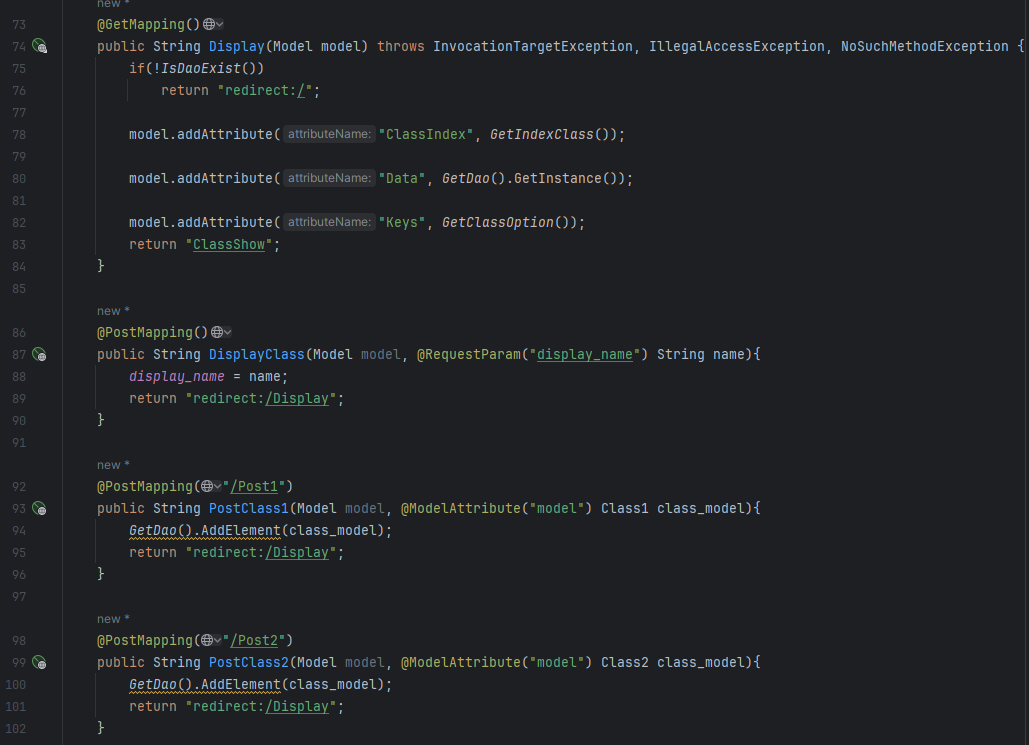


Рисунок 19 Создание запросов отображения данных и примеры добавления новых данных в поля DAO

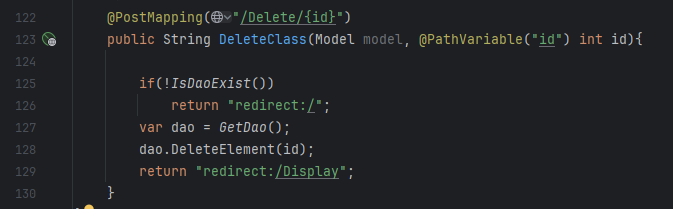


Рисунок 20 Создание запроса удаления записи из выбранного поля DAO

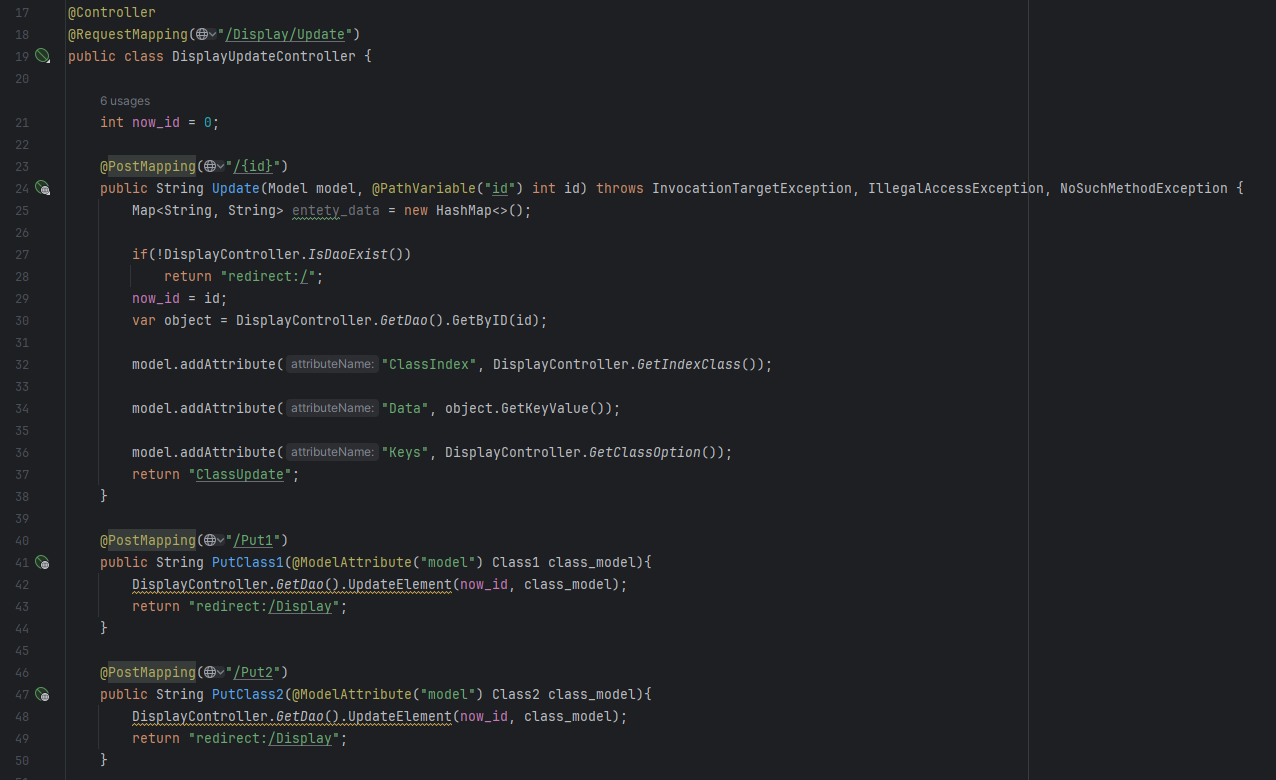


Рисунок 21 Создание контроллера изменения данных полей DAO

1. Создание разметки

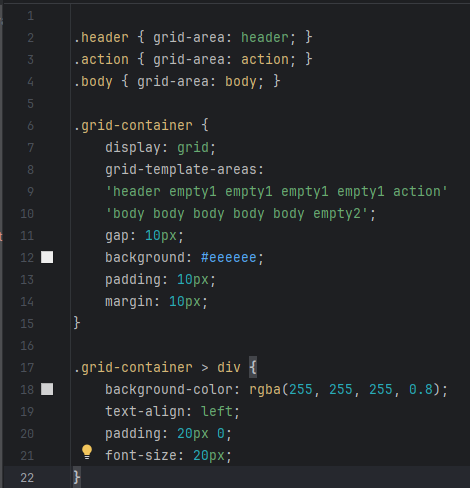


Рисунок 22 Создание стиля для отображения данных в виде ячеек

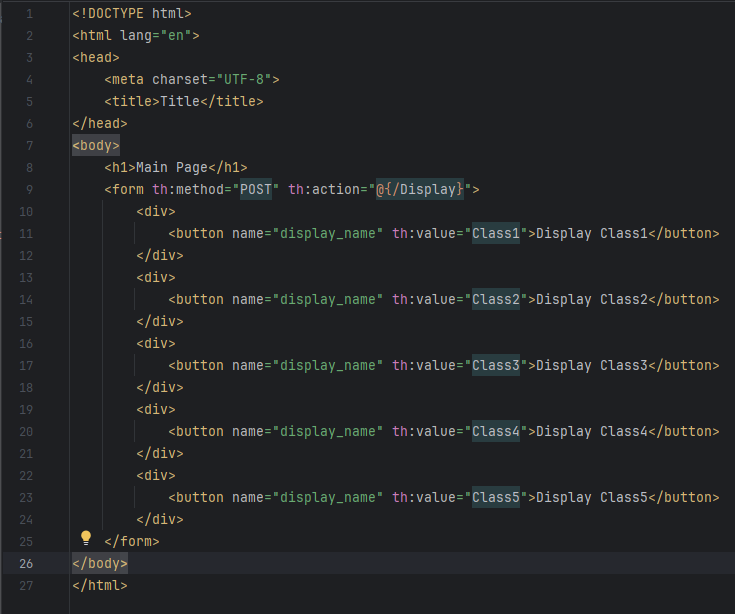


Рисунок 23 Разметка начальной страницы (Перехода на страницы с отображением информации)

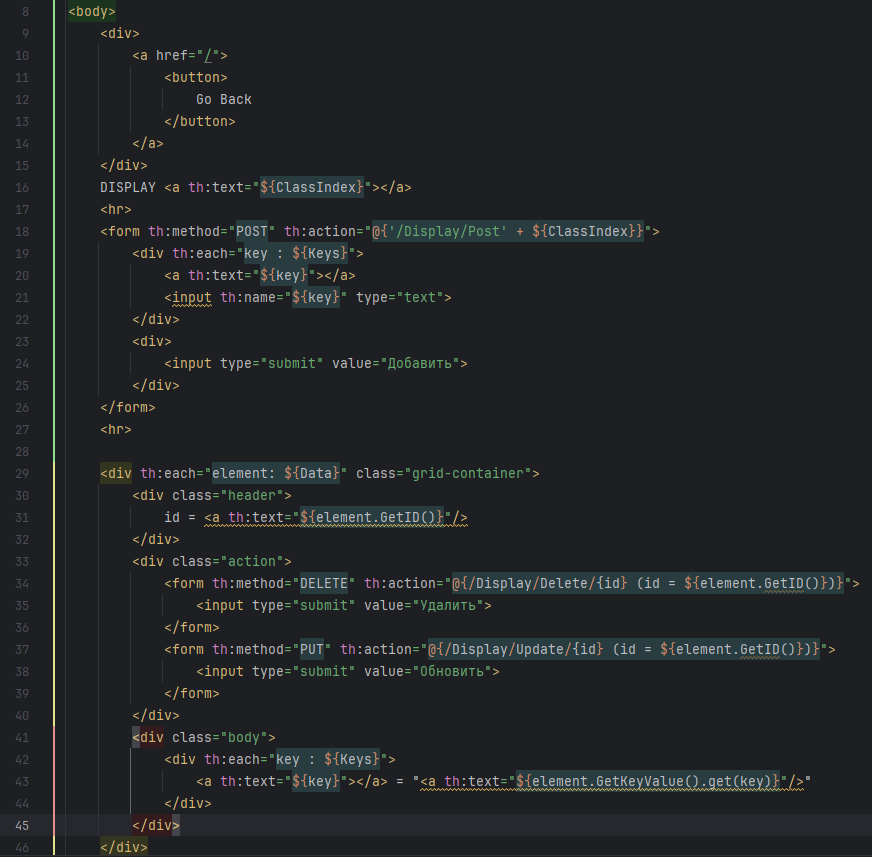


Рисунок 24 Создание разметки для отображения данных выбранного DAO (С выводом названия полей данного класса)

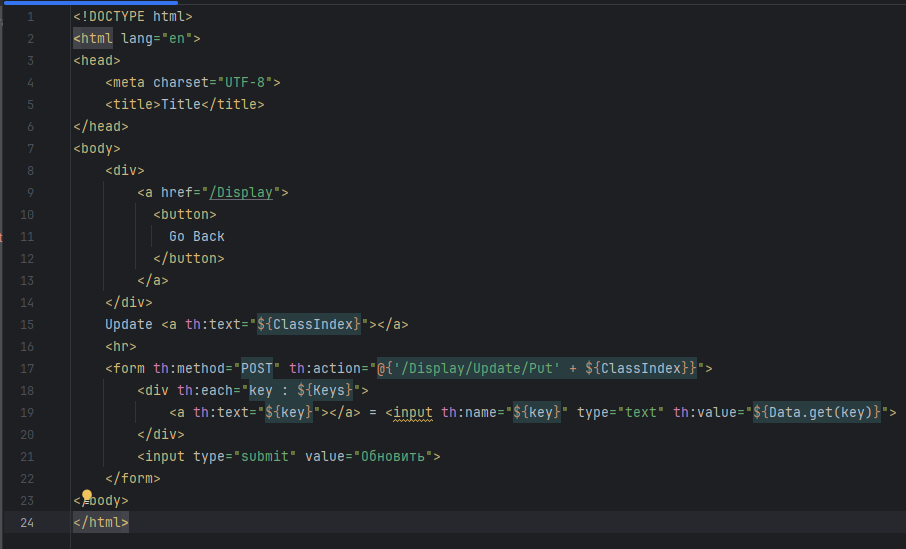


Рисунок 25 Разметка для вывода и редактирования данных одной записи

1. Результат

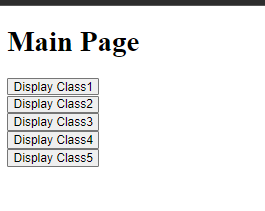


Рисунок 26 Начальная страница

Для каждого класса используется один файл отображения и окно изменения. Используется статический метод вывод названия всех полей через List, и абстрактный метод который выводит значения записи по ключам содержащихся в List. Создаваемые поля ввода генерируются в fpom что позволяет использовать @ModelAttribute для генерации нужного класса.

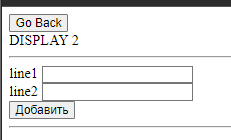


Рисунок 27 Страница отображения (пустая (Класс №2))

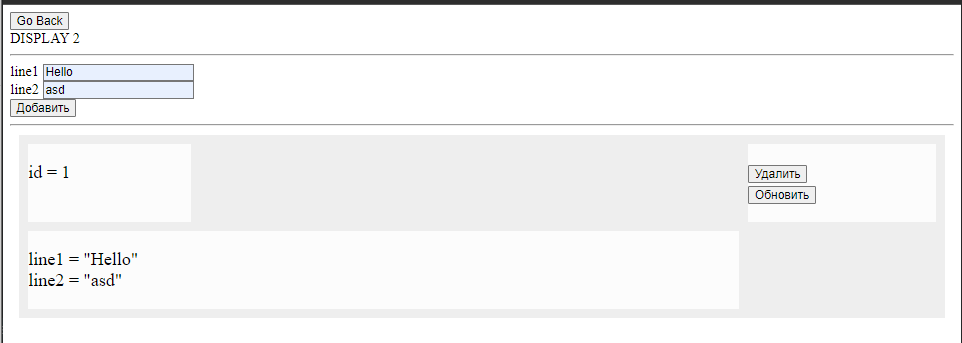


Рисунок 28 Страница отображения (не пустая (Класс №2))

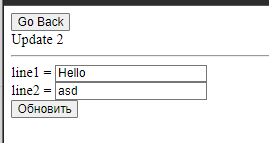


Рисунок 29 Переход на страницу изменения данных

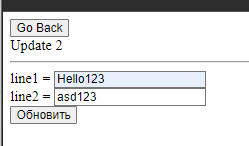


Рисунок 30 Изменение данных

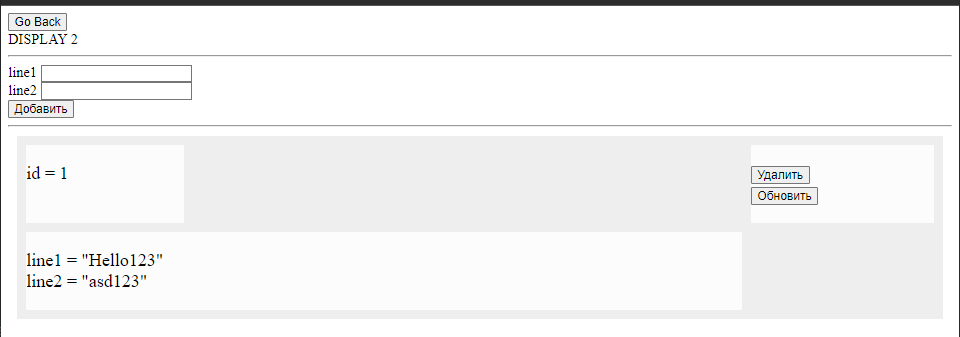


Рисунок 31 Переход обратно на страницу отображения

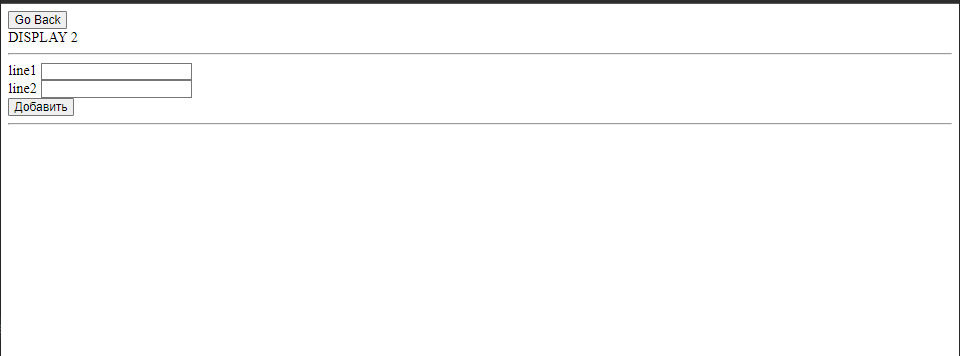


Рисунок 32 Удалить

Вывод: В ходе практической работы было создано приложения для хранения, изменения, удаления различных классов используя вспомогательный класс DAO.