|  |
| --- |
| **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.02.01

Разработка и администрирование баз данных .

Специальность 09.02.07

Информационные системы и программирование .

Студент Разумов Никита Дмитриевич.

*(фамилия, имя, отчество)*

Группа П50-3-20

Руководитель по практической подготовке от техникума

Пахомов Даниил Александрович.

*(фамилия, имя, отчество)*

«\_\_» \_\_\_\_\_\_ 2023 года

Цель работы: создать веб-приложение с использованием @GetMapping и @PostMapping @RequestParam. Приложение должно включать в себя главную страницу и две кнопки, отвечающие за вкладки с калькулятором и конвертером валют.

Ход работы:

Сначала создаем контроллер основной страницы и 3 файла html. Они будут отвечать за вкладки и переход между ними. В контроллере прописываем GetMapping и не забываем подключить все нужные фреймворки.



Рисунок 1 HomeController

Теперь нужно сделать разметку на первой странице. Добавляем стили, подключаем google шрифт. После этого принимаем значение Title, прописываем его и создаем две кнопки.



Рисунок 2 Верстка основной страницы

После этого нужно сделать вторую страницу. Это будет калькулятор. Также подключаем шрифт, пишем css. Далее создаем все кнопки калькулятора и прописываем их логику. Создаем дисплей калькулятора.

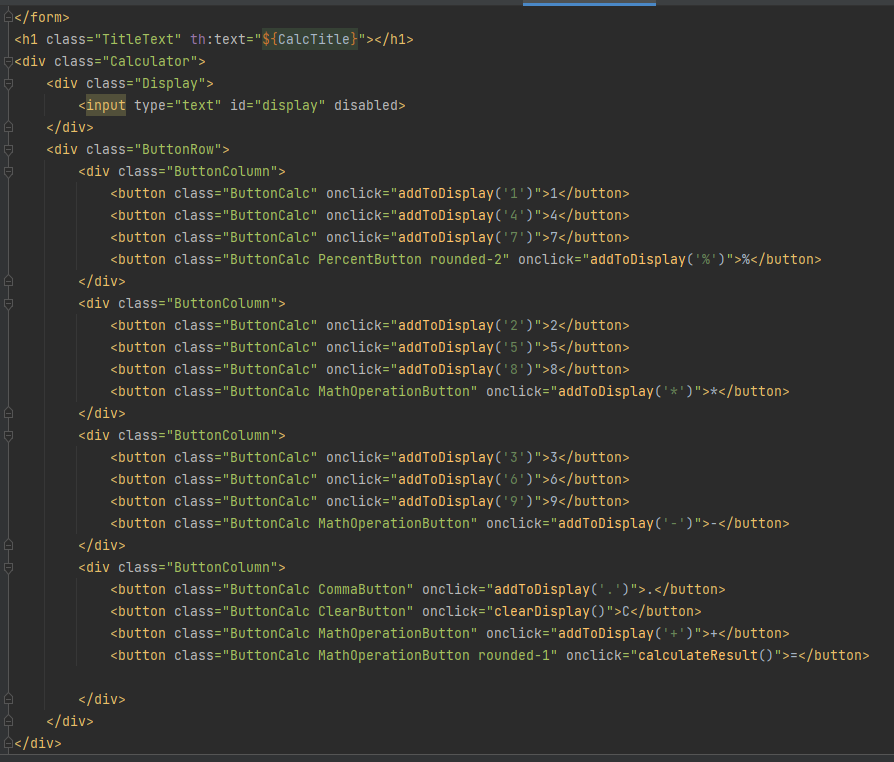


Рисунок 3 Разметка калькулятора

В скрипте файла пишем функцию процента, очистка поля ввода, функция добавления символов на дисплей и основная функция для выполнения математических вычислений.

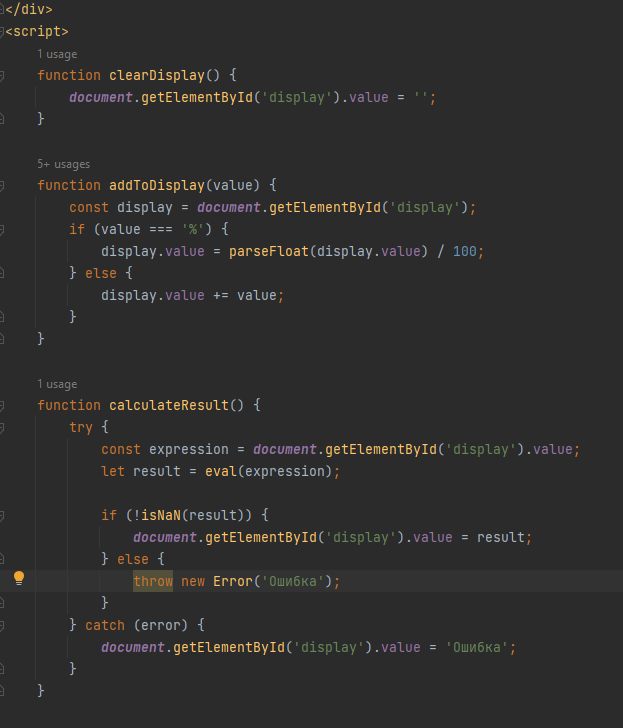


Рисунок 4 Логика страницы

После этого переходим к конвертеру. Создаем новый контроллер для конвертера. Переносим туда GetMapping, подключаем фреймворки и пишем код для хранения информации о валютах и их курсе. Для этого создаем соответствующие классы.

CurrencyInfo: Этот класс представляет информацию о валюте. У него есть приватные поля name (название валюты), rate (курс валюты) и nominal (номинал валюты). Конструктор класса принимает эти значения при создании объекта, и есть методы getName(), getRate() и getNominal(), чтобы получать соответствующие данные.

CurrencyData: Этот класс представляет коллекцию данных о валютах. Он использует HashMap, чтобы хранить информацию о различных валютах, используя коды валют в качестве ключей и объекты CurrencyInfo в качестве значений. Конструктор класса инициализирует начальные данные для нескольких валют (USD, RUB, EUR, BYR), но можно добавить больше данных о других валютах, если это необходимо. Класс также предоставляет методы getCurrencyInfo(String code), чтобы получить информацию о валюте по её коду, и getCurrencyCodes(), чтобы получить набор всех доступных кодов валют.

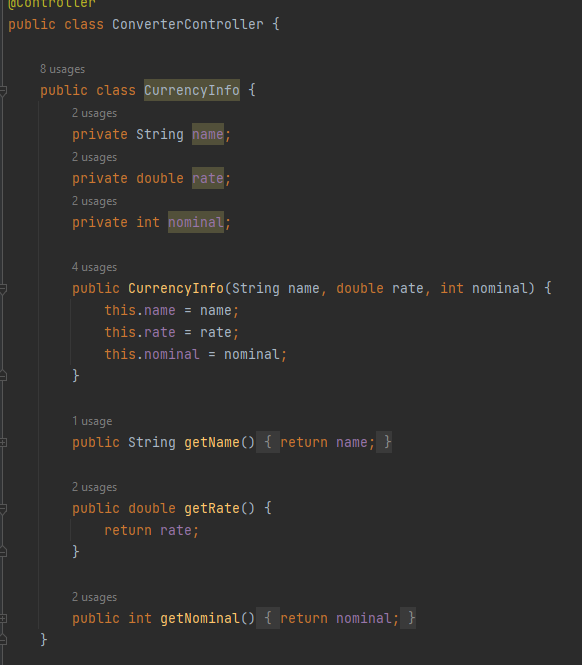


Рисунок 5 Класс CurrencyInfo

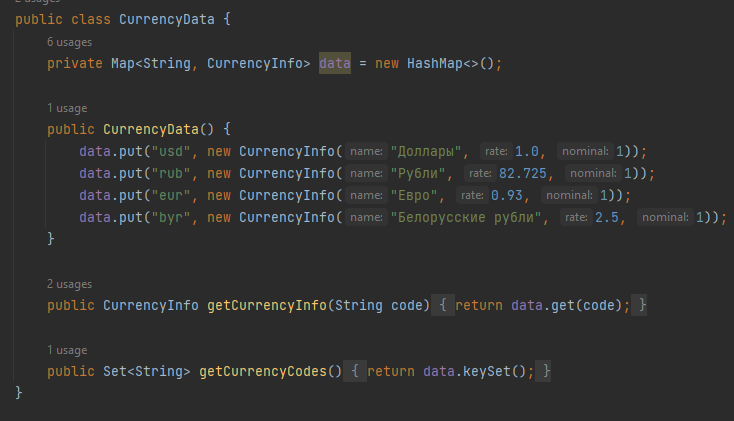


Рисунок 6 Класс CurrencyData

Теперь нужно использовать PostMapping и прописать логику конвертирования. Этот метод контроллера обрабатывает POST-запрос на "/Converter". Он принимает выбранную валюту и сумму, затем выполняет конвертацию суммы в выбранной валюте во все остальные валюты. Результаты конвертации добавляются в список и передаются в представление с другими данными, такими как заголовок и исходная сумма.

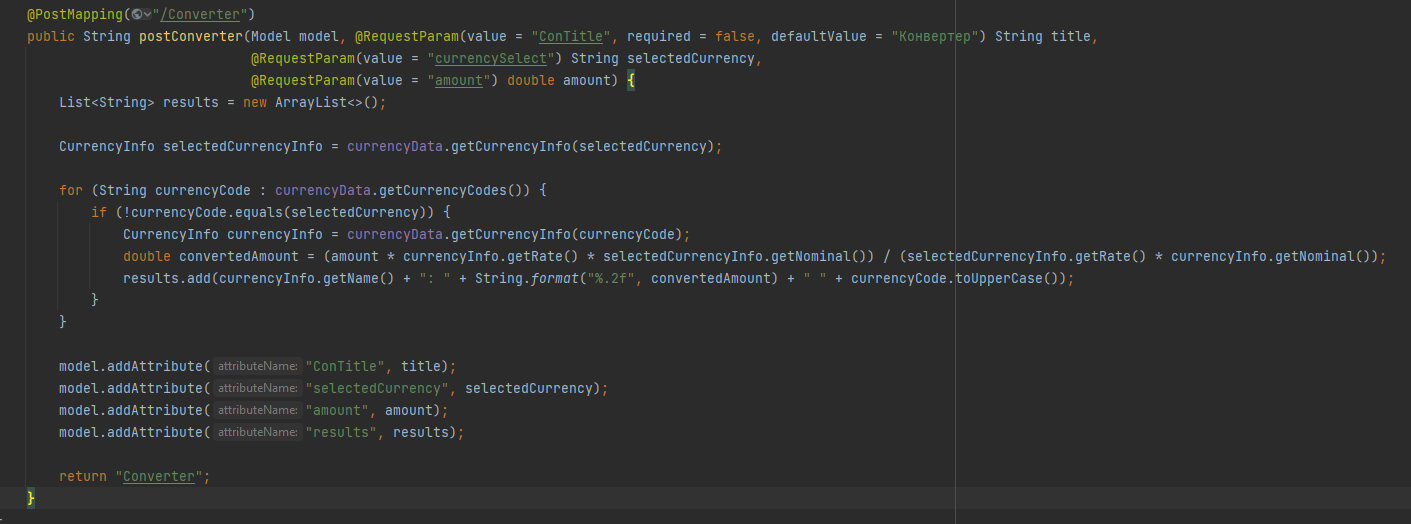


Рисунок 7 Логика конвертера

Теперь создаем разметку для конвертера. Делаем сначала все тоже самое для дизайна, после чего переходим к основному телу кода. Этот HTML-код представляет веб-страницу с двумя формами: первая форма отправляет GET-запрос на "/Home" при нажатии кнопки "Назад", а вторая форма содержит конвертер валют с выбором валюты в выпадающем списке, полем ввода суммы и кнопкой "Конвертировать". Результаты конвертации отображаются под формой.

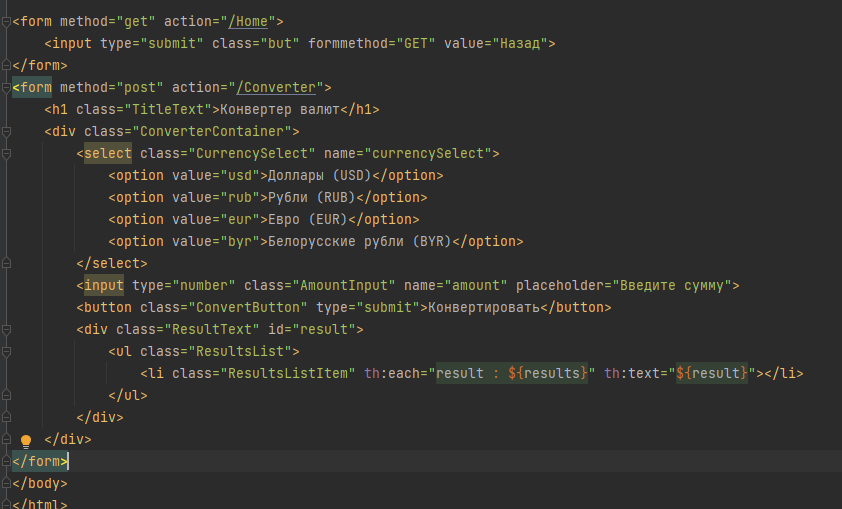


Рисунок 8 Разметка конвертера

Итогом работы являются три вкладки:

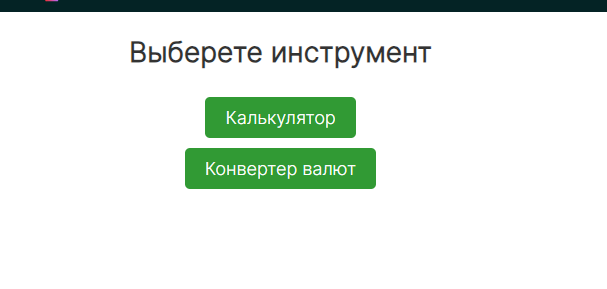


Рисунок 9 Основная страница

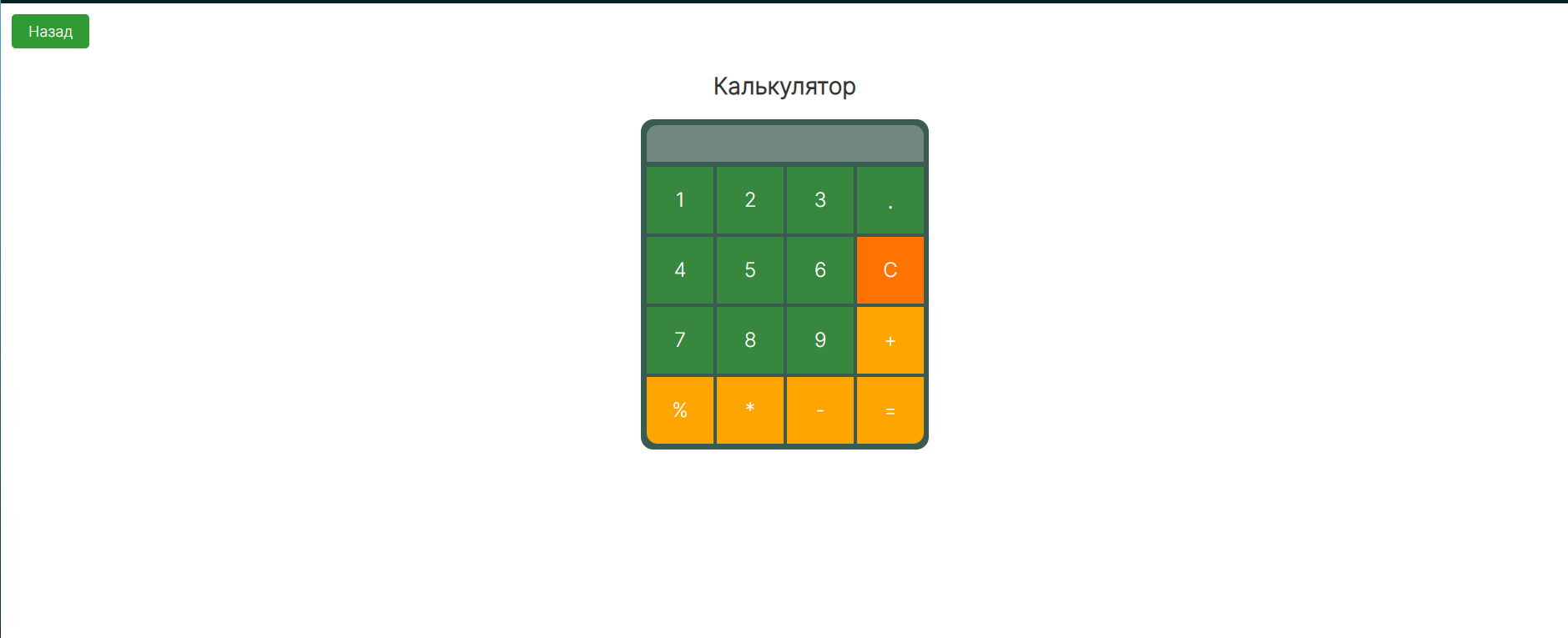


Рисунок 10 Калькулятор

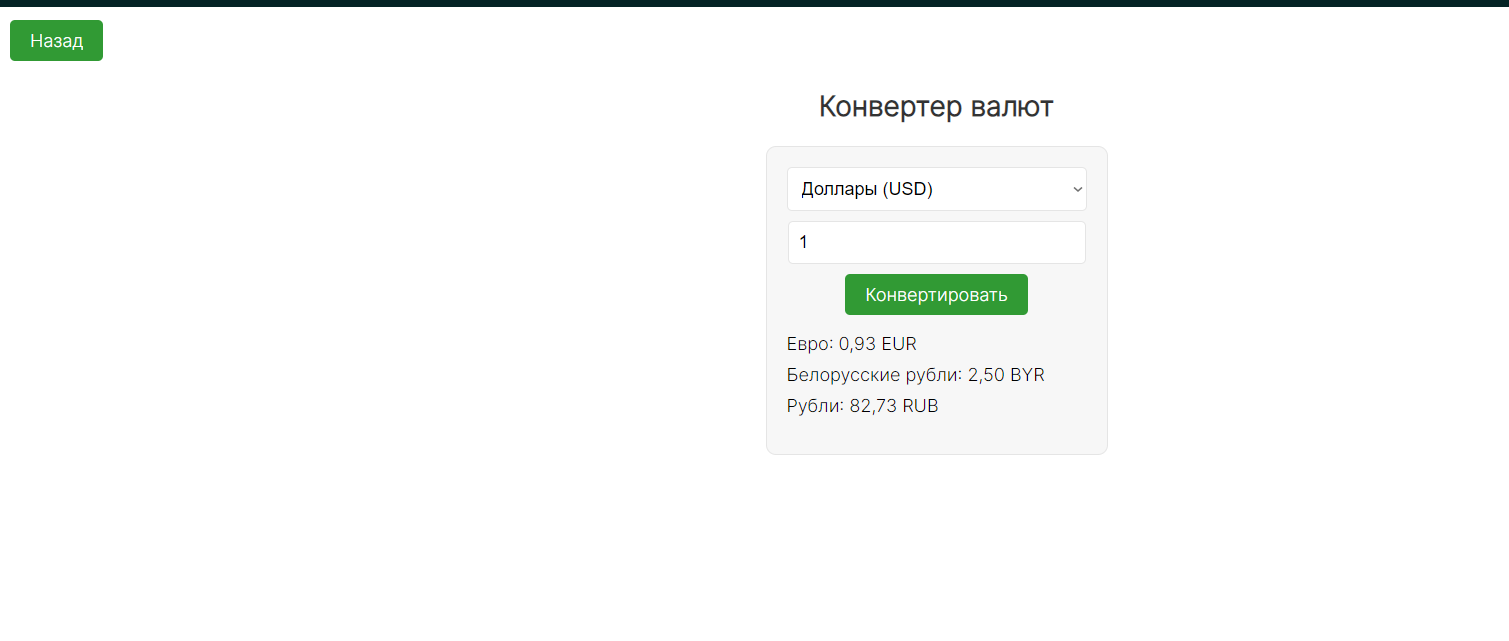


Рисунок 11 Конвертер

Вывод: была выполнена работа по созданию веб-приложения с двумя инструментами. В процессе создания использовались @GetMapping, @PostMapping и @RequestParam. Также были улучшены навыки верстки.