

Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение высшего образования  
«Российский экономический университет имени Г.В. Плеханова»  
**МОСКОВСКИЙ ПРИБОРОСТРОИТЕЛЬНЫЙ ТЕХНИКУМ**

Специальность 09.02.07 Информационные системы и программирование  
Квалификация: Программист

**ПРАКТИЧЕСКАЯ РАБОТА №1**  
**УП 04.01**

Листов: 5

Выполнил: студент  
группы П50-6-22  
Ахвердиева С.Н.

Проверил преподаватель  
Д. В. Серяк  
«\_\_\_\_» \_\_\_\_\_ 2025 года

Москва 2025

Цель работы: создать приложение, которое включает в себя главную страницу, калькулятор и конвертер валют, используя @GetMapping, @PostMapping и @RequestParam.

Ход работы:

### 1. Калькулятор

По адресу /calculator загружается страница с формой ввода двух чисел и выбора операции. После отправки формы запрос отправляется методом POST на /calculate, где контроллер с помощью @RequestParam получает данные, выполняет вычисление и передаёт результат на страницу calc-result.html через ModelAndView.

### 2. Конвертер валют

Страница выбора валют и суммы открывается по адресу /currency, а обработка данных выполняется по адресу /convert. Контроллер принимает сумму и выбранные валюты, рассчитывает итог по заданным курсам и выводит результат на currency-result.html, также используя ModelAndView.

Для отображения страниц применяются методы с аннотацией @GetMapping, для обработки форм – @PostMapping, а значения из формы передаются в контроллер через @RequestParam.

```
@Controller no usages
public class MainController {

    @GetMapping("/") no usages
    public String index() {
        return "index";
    }

    @GetMapping("/calculator") no usages
    public String calculator() {
        return "calculator";
    }

    @PostMapping("/calculate") no usages
    public String calculate(
        @RequestParam("a") double a,
        @RequestParam("b") double b,
        @RequestParam("op") String op,
        Model model
    ) {
        double result = 0;

        switch (op) {
            case "+" -> result = a + b;
            case "-" -> result = a - b;
            case "*" -> result = a * b;
            case "/" -> result = (b != 0) ? a / b : Double.NaN;
        }

        model.addAttribute( attributeName: "a", a);
        model.addAttribute( attributeName: "b", b);
        model.addAttribute( attributeName: "op", op);
        model.addAttribute( attributeName: "result", result);
    }

    return "calc-result";
}
```

Рисунок 1 – Код

```
@PostMapping("/convert") no usages
public String convert(
    @RequestParam("from") String from,
    @RequestParam("to") String to,
    @RequestParam("amount") double amount,
    Model model
) {
    double rate = getRate(from, to);
    double result = amount * rate;

    model.addAttribute( attributeName: "from", from);
    model.addAttribute( attributeName: "to", to);
    model.addAttribute( attributeName: "amount", amount);
    model.addAttribute( attributeName: "rate", rate);
    model.addAttribute( attributeName: "result", result);

    return "convert-result"; // шаблон convert-result.html
}

private double getRate(String from, String to) { 1 usage
    if (from.equals(to)) return 1.0;

    if (from.equals("RUB") && to.equals("USD")) return 0.011;
    if (from.equals("USD") && to.equals("RUB")) return 90.0;

    if (from.equals("RUB") && to.equals("EUR")) return 0.01;
    if (from.equals("EUR") && to.equals("RUB")) return 100.0;

    if (from.equals("USD") && to.equals("EUR")) return 0.9;
    if (from.equals("EUR") && to.equals("USD")) return 1.1;

    return 1.0;
}
}
```

Рисунок 2 - Код

1. Результат.

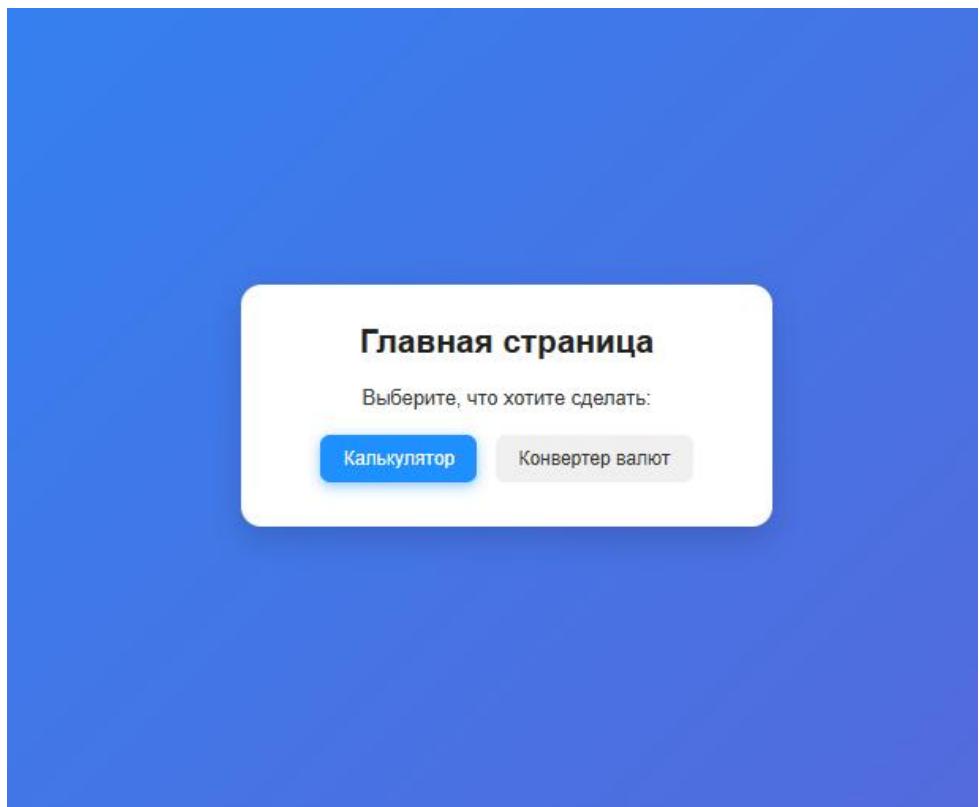


Рисунок 3 - Результат

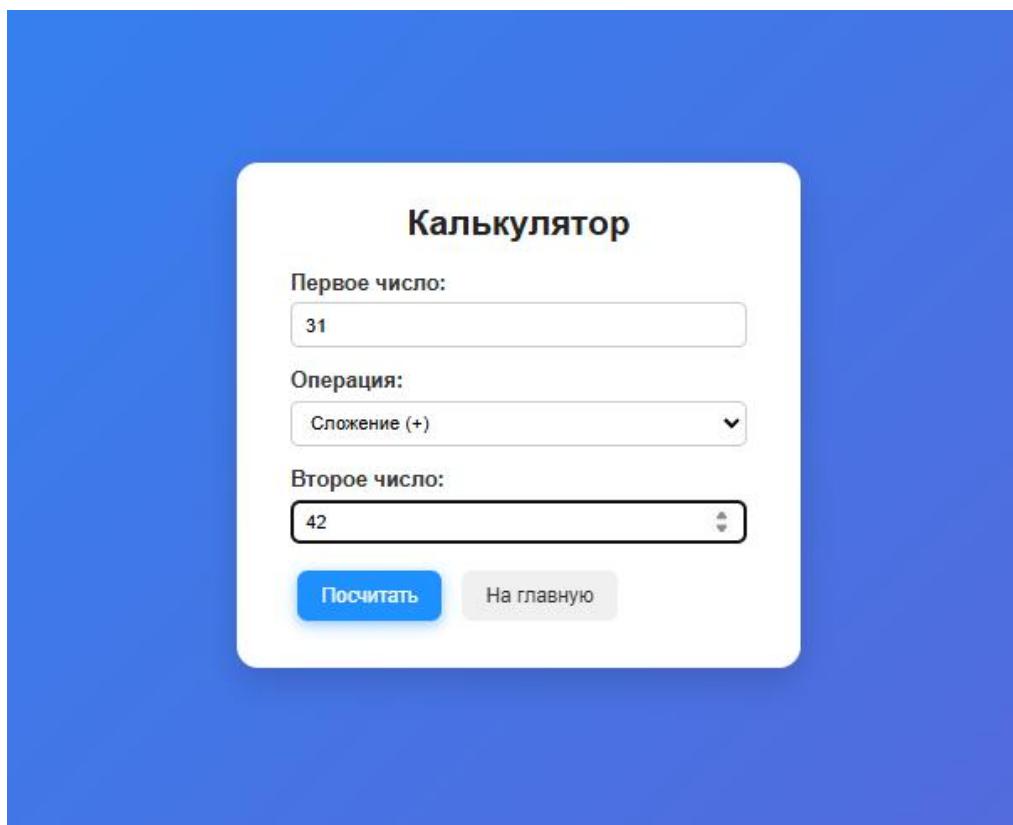


Рисунок 4 – Результат

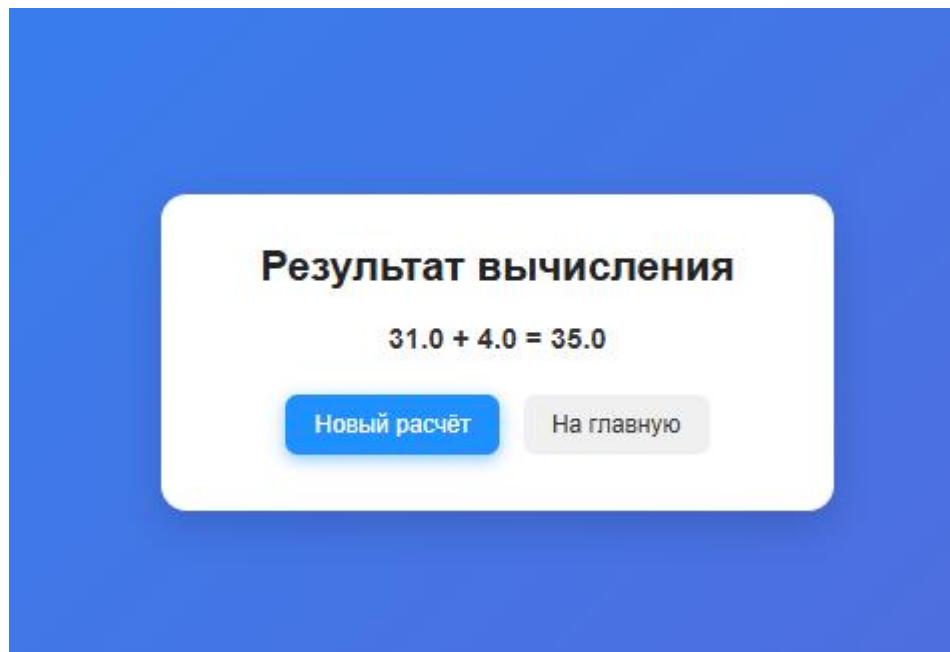


Рисунок 5 - Результат

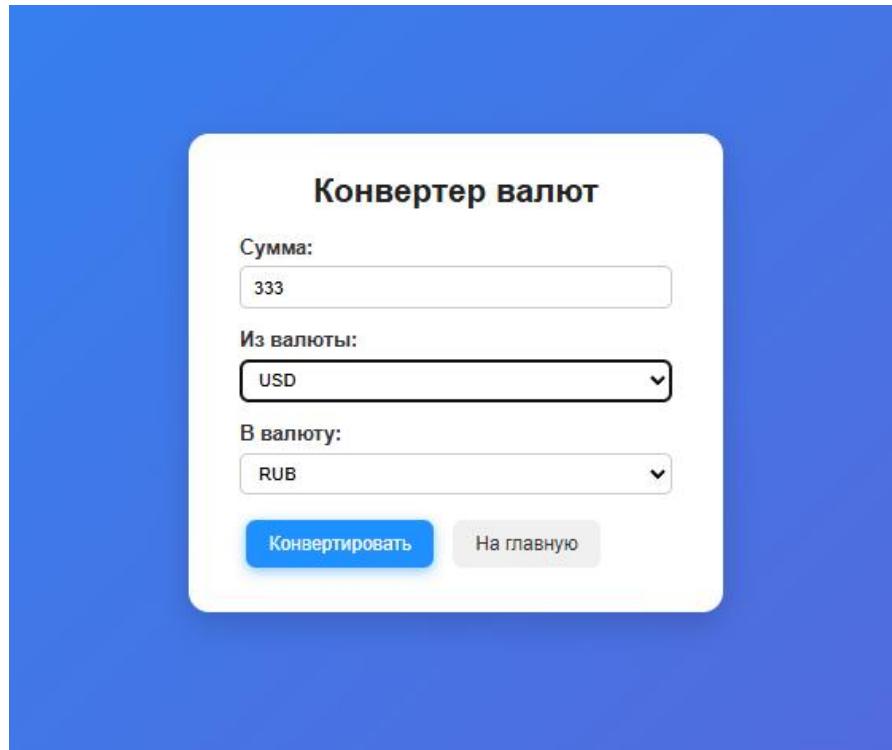


Рисунок 6 – Результат

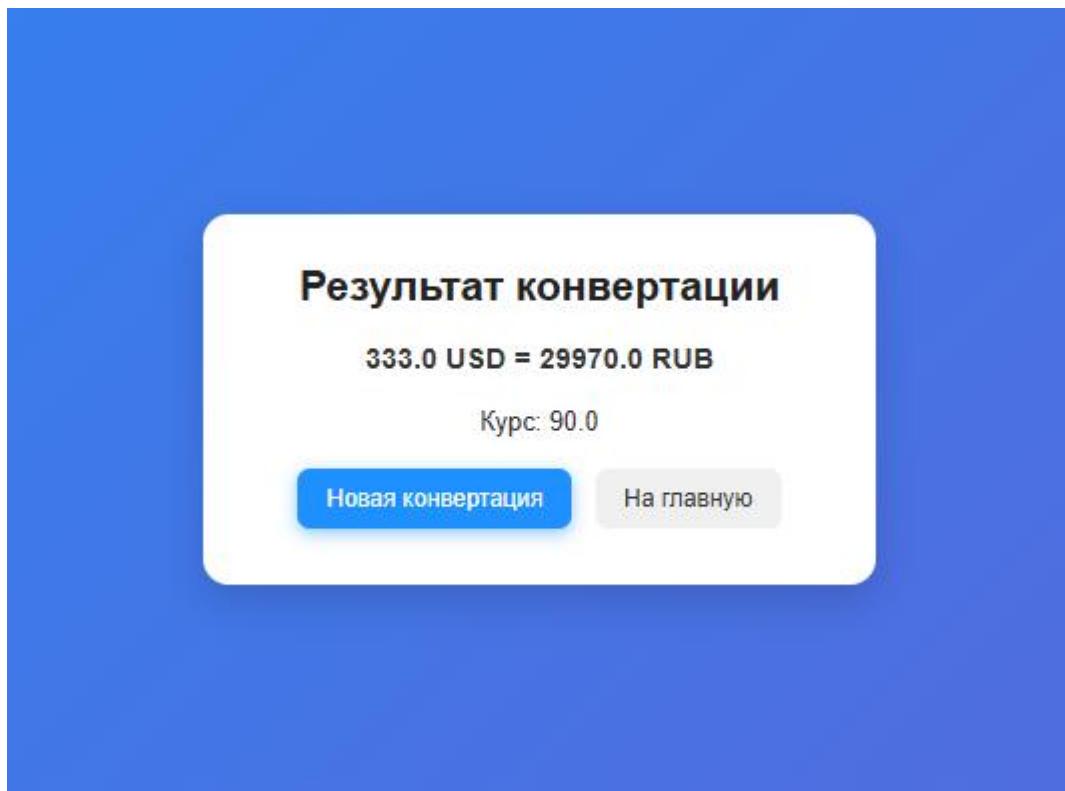


Рисунок 7 - Результат

Вывод: в ходе практической работы была создано приложение, которое включает в себя главную страницу, калькулятор и конвертер валют, используя `@GetMapping`, `@PostMapping` и `@RequestParam`.