A gold eagle with two heads and a blue circle with a white circle and a blue circle with a blue circle with a blue circle with a white circle with a blue circle with a blue circle with

AI-generated content may be incorrect.

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

**«МИРЭА – Российский технологический университет»**

**РТУ МИРЭА**

Институт Информационных технологий

Кафедра Инструментального и прикладного программного обеспечения

**ПРАКТИЧЕСКАЯ РАБОТА №2**

по дисциплине «Проектирование и разработка серверных частей интернет- ресурсов»

**Студент группы** ИКБО-21-23 Муравьев А.О.

(подпись студента)

**Руководитель практической работы** Благирев М.М.

(подпись руководителя)

Работа представлена « » 2025 г.

Допущен к работе « » 2025 г.

Москва 2025

# СОДЕРЖАНИЕ

[ЦЕЛЬ РАБОТЫ 3](#_Toc208567963)

[ХОД РАБОТЫ 4](#_Toc208567964)

[ВЫВОД 12](#_Toc208567965)

[ОТВЕТЫ НА КОНТРОЛЬНЫЕ ВОПРОСЫ Ошибка! Закладка не определена.](#_Toc208567966)

# ЦЕЛЬ РАБОТЫ

Предлагается, используя серверную конфигурацию, разработанную в прошлой практической работе выполнить следующие упражнения. Предполагается создать 3 независимых сервиса, устойчивых к минимальному набору самых простых ошибок. Предполагается создание 1 общего проекта с разделенными сервисами, разделением проекта на файлы для разделения функционала и переиспользования файлов. Каждый сервис должен состоять как минимум из 2 файлов.

# ХОД РАБОТЫ

## Упражнение 1 - Drawer

Задача: создать веб-сервис Drawer для рисования svg объектов. Ему передается один параметр - целое число, представляющее закодированная фигура для рисования.

Суть реализации: при отправке GET-запроса с информацией о фигуре сервер возвращает код svg-фигуры с запрашиваемыми данными.

Обработкой запросов занимается класс WebServer. Он принимает и обрабатывает сырые запросы, используя другие компоненты программы. Фрагмент класса WebServer показан на рисунке 1.

### Рисунок 1 – Фрагмент класса WebServer

На рисунке 2 продемонстрирован фрагмент обработчика GET-запроса, который отвечает за сборку сырого ответа с svg-картинкой (картинка собирается в методе SvgRenderer.Render()).

### Рисунок 2 – Фрагмент обработчика GET-запросов

Основной компонент кода – класс SvgRenderer. С помощью тега <svg>, неймспейса w3.org/2000/svg и тегов circle и rect создаются круг и прямоугольник соответственно. Более сложные фигуры (треугольник и звезда) создаются с помощью polygon и вычислением точек (вершин) фигур. Фрагмент класса SvgRenderer показан на рисунках 3-4.

### Рисунок 3 – Фрагмент класса SvgRenderer, часть 1 из 2

### Рисунок 4 – Фрагмент класса SvgRenderer, часть 2 из 2

В результате при отправке GET-запроса с данными фигуры мы получаем запрошенную svg-картинку.

### Рисунок 5 – Результат работы Drawer

## Упражнение 2 - Sorter

Задача: реализовать сортировку слиянием.

Массив передается скрипту как параметр строка: состоящий из значений элементов массива, разделенных запятыми. Итогом является вебстраница, содержащая отсортированный массив.

Суть решения: класс WebServer принимает сырой GET-запрос, разархивирует данные и отправляет их на сортировку классу MergeSorter, а затем возвращает страничку с отсортированным массивом.

Фрагмент класса WebServer, отвечающий за вызов сортировки показан на рисунке 6.

### Рисунок 6 – Фрагмент WebServer, отвечающий за вызов сортировки

На рисунке 7 показан статический метод сортировки в классе MergeSorter.

### Рисунок 7 – Метод сортировки слиянием

Теперь при отправке GET-запроса серверу с перечислением элементов массива мы получаем страничку с отсортированными элементами этого массива (рисунок 8).

### Рисунок 8 – Результат работы Sorter

## Упражнение 3 - AdminPanel

Задача: реализовать информационно-административную веб-страницу о сервере с помощью таких команд Unix как: ls, ps, whoami, id и так далее.

Суть решения: класс WebServer работает с сырыми запросами и отправляет данные в другие классы, ShellSafeBuilder принимает unix-команды, проверяет их и преобразовывает в зависимости от текущей системы (под windows shell и unix команды), Commands занимается очисткой и уточнением команд (флагами, параметрами и прочим), CommandRunner занимается запуском команд в зависимости от текущей системы.

Для реализации пародии на SSH-доступ был добавлен токен доступа, хранящийся в окружении (конфиге docker-compose). В рамках данной работы токен также будет передаваться через GET-запрос.

Фрагмент класса WebServer, работающий с Shell, показан на рисунке 9.

### Рисунок 9 – Фрагмент класса WebServer, отвечающий за сборку команды и её вызов

Фрагмент класса ShellSafeBuilder, проверяющий команды, показан на рисунке 10.

### Рисунок 10 – Фрагмент класса ShellSaveBuilder

Фрагмент класса Commands, уточняющий команды, показан на рисунке 11.

### Рисунок 11 – Фрагмент класса Commands

Фрагмент класса CommandRunner, отвечающий за запуск команд, показан на рисунке 12.

### Рисунок 12 – Фрагмент класса CommandRunner

# ВЫВОД

Таким образом, были разработаны 3 независимых сервиса: Drawer, Sorter и AdminPanel.

Исходный код проекта расположен по адресу:

<https://github.com/alexomur/MireaBackend/tree/master/Prac2>