ООО «Гикбраинс»

Факультет: Разработчик

Технологическая специализация: Веб-разработка на Java

**ДИПЛОМНАЯ РАБОТА**

на тему:

«Создание веб-приложения для расчета контура заземления подстанции»

Исполнитель:

Студент факультета «Разработчик»

специализации «Веб-разработка на Java»

Ивлев А.Н.

Москва

2024

**СОДЕРЖАНИЕ**

|  |  |
| --- | --- |
| Наименование | стр. |
| Введение | 3 |
| Глава 1. Основные технические решения. | 5 |
| 1.1. Требования к приложению | 5 |
| Глава 2. Проектные решения | 7 |
| 2.1. Описание класса Grounding | 7 |
| 2.2. Описание класса GroundingController | 8 |
| 2.3. Описание класса GroundingServis | 9 |
| 2.4. Описание WEB –интерфейса. | 13 |
| Заключение | 17 |
| Список используемых источников | 18 |
| **Приложения** | 19 |
| Приложение 1. Пример выполнения заземления электрической подстанции напряжением 110/10кВ | 20 |
| Приложение 2. Интерфейс приложения. Исходные данные. | 21 |
| Приложение 3. Интерфейс приложения. Ввод данных, поправочные коэффициенты | 22 |
| Приложение 4. Интерфейс приложения. Расчетные значения. | 23 |
| Приложение 5. Код страницы Web-интерфейса | 24 |
| Приложение 6. Код класса main | 30 |
| Приложение 7. Код класса GroundingServis | 31 |
| Приложение 8. Код класса Grounding | 36 |
| Приложение 9. Код класса GroundingController | 38 |

**ВВЕДЕНИЕ**

Дипломная работа посвящена разработке веб-приложения, предназначенного для автоматизированного расчета контура заземления подстанции. Расчет контура заземления применяется в проектировании электрических подстанций с напряжением от 0.4 до 750кВ. Контур заземления на электроподстанциях используется для обеспечения безопасности электротехнического оборудования и персонала. Он предназначен для отведения токов утечки, коротких замыканий или молниевых разрядов к земле, минимизируя риск поражения электрическим током. Таким образом, контур заземления играет ключевую роль в обеспечении надежной и безопасной работы электроподстанции.

С развитием технологий и повышением требований к безопасности в энергетике, расчет контура заземления становится все более значимым процессом. Создание веб-приложения для проведения подобных расчетов позволит инженерам и специалистам в области электроэнергетики удобно и быстро оценивать необходимые параметры заземления, что повысит эффективность работы и обеспечит более точные результаты.

Проект будет использовать следующие технологии и инструменты разработки для создания удобного и эффективного инструмента:

- Spring Framework: предоставляет обширный набор инструментов и функций для разработки Java-приложений. Он используется для упрощения создания масштабируемых, надежных и поддерживаемых приложений. Spring Framework предлагает решения для управления зависимостями, обеспечения безопасности, обработки транзакций, взаимодействия с базами данных и многое другое.

- Lombok: библиотека для языка программирования Java, которая позволяет ускорить процесс разработки, уменьшая количество шаблонного кода. Она предоставляет аннотации для создания геттеров, сеттеров, конструкторов, методов equals и hashCode, а также других стандартных методов JavaBean, автоматически генерирует байт-код для этих методов во время компиляции. Ломбок позволяет улучшить читаемость и структуру кода, делая его более лаконичным и чистым..

- Java: Универсальный язык программирования, выбранный для написания веб-приложения, обеспечивающий надежность и масштабируемость. Он является объектно-ориентированным языком, который позволяет разработчикам создавать разнообразные приложения, начиная от мобильных приложений и веб-приложений до больших корпоративных систем.

- Thymeleaf: шаблонизатор для языка Java, который используется для создания веб-приложений и веб-страниц. Он позволяет разработчикам создавать динамические веб-страницы, которые могут взаимодействовать с данными Java. Thymeleaf позволяет использовать HTML в качестве шаблонов и вставлять в них данные, управлять отображением и форматированием данных, условными итерациями, а также другими операциями, что делает процесс создания веб-страниц более удобным и эффективным..

Актуальность темы связана с моей работой в области проектирования электрических подстанций. Данный калькулятор поможет быстро подобрать оптимальный состав контура заземления (вертикальный и горизонтальный заземлитель), уменьшить затраты на материалы и работы по монтажу контура заземления.

**Глава 1. Основные технические решения.**

**1.1. Требования к приложению**

- Пользователь должен иметь возможность изменять справочные коэффициенты.

-Пользователь при вводе данных должен наблюдать значения справочных коэффициентов.

- Не должно быть ограничений по количеству применяемый материалов (количество вертикальных электродов, количество горизонтального заземлителя).

- Пользователь должен наблюдать сопротивление отдельных элементов контура заземления.

1.2. Структура приложения.

Для разработки серверной части приложения используется IntelliJ IDEA. Этот стек технологий обеспечивает разработку приложения с учетом требований к функциональности, производительности и безопасности.

Архитектура приложения в рамках JavaSpring основывается на паттерне проектирования MVC (Model-View-Controller), который позволяет разделить приложение на три основных компонента: модель (Model), представление (View) и контроллер (Controller). Наше приложение состоит из данных составных частей:

Модель (Model):

Модель представляет собой структуру данных приложения.

Включает классы, описывающие сущности приложения ("Пользователь", "Запись чувств" ).

Представление (View):

Представление отвечает за отображение данных пользователю.

В веб-приложениях на JavaSpring представление часто реализуется с использованием шаблонизаторов, таких как Thymeleaf или JSP, которые позволяют вставлять данные из контроллера непосредственно в HTML-шаблоны.

Контроллер (Controller):

Контроллер обрабатывает запросы от пользователя и взаимодействует с моделью и представлением.

В JavaSpring контроллеры представлены классами, аннотированными @Controller или @RestController.

Контроллеры содержат методы для обработки различных типов запросов (GET, POST, PUT, DELETE) и вызывают соответствующие сервисы для выполнения бизнес-логики.

Сервис (Service):

Сервис содержит бизнес-логику приложения.

Он обычно используется для выполнения операций с данными, координирования действий между различными компонентами приложения и обеспечения единого доступа к функциональности приложения.

Сервер (Server):

Сервер представляет собой окружение, в котором работает приложение.

В JavaSpring приложение обычно запускается на встроенном сервере, таком как Tomcat или Jetty, который обрабатывает HTTP-запросы и управляет жизненным циклом приложения.

Каждый из этих компонентов играет важную роль в построении архитектуры приложения на JavaSpring, обеспечивая его модульность, гибкость и масштабируемость. Они позволяют разделить различные аспекты приложения и обеспечить их независимую разработку и тестирование.

**Глава 2. Проектные решения**

**2.1. Описание класса Grounding.**

В представленном классе Grounding, который содержит поля переменных, справочных коэффициентов для расчета параметров заземления. В классе используется библиотека Lombok для генерации геттеров и сеттеров, а также других методов, таких как toString(), hashCode(), и equals(). Поля класса Grounding представлены на рисунках 1 и 2.

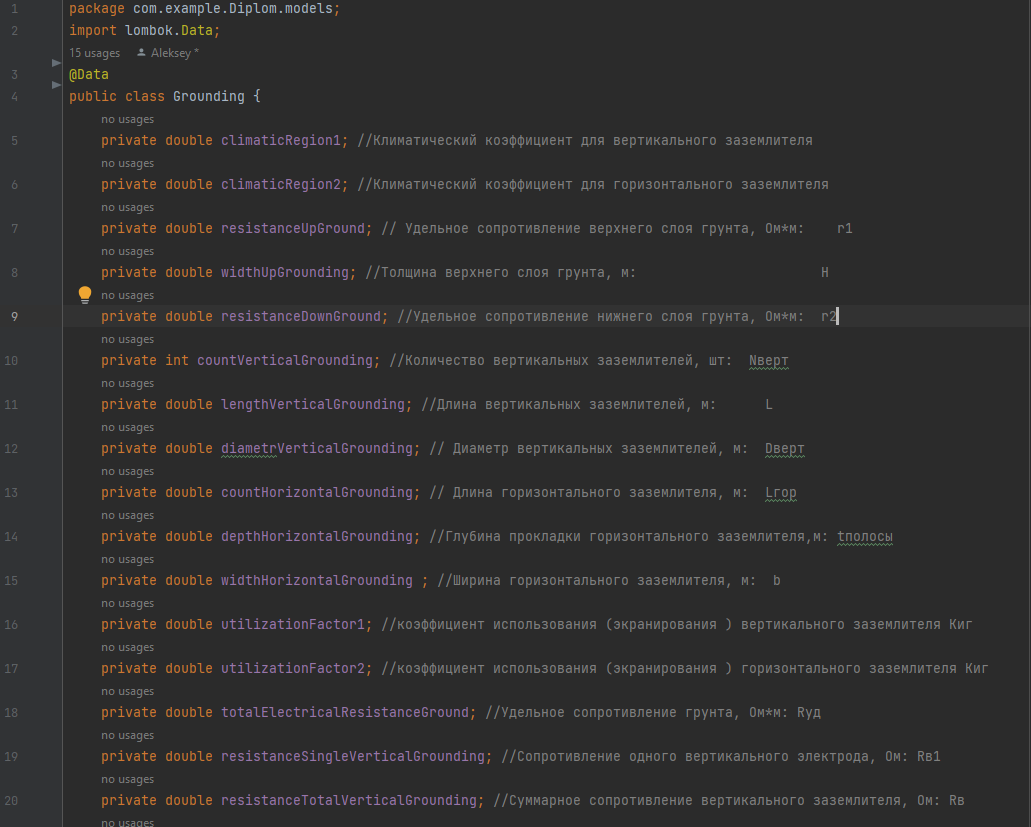
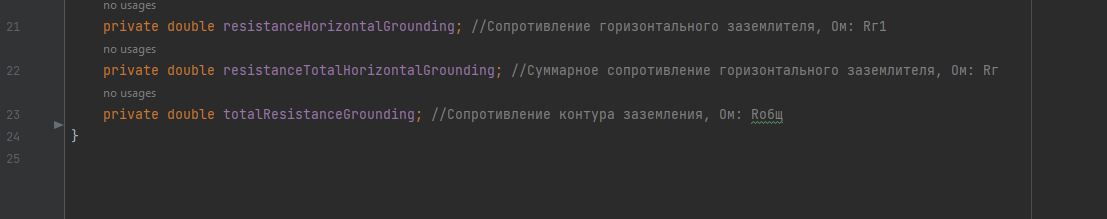


Рисунок 1. Поля класса Grounding. Начало

 Рисунок 2 . Поля класса Grounding. Окончание

**2.2 Описание класса GroundingController**

GroundingController - используется для управления запросами от клиента и взаимодействия с другими компонентами приложения. Он является частью паттерна проектирования MVC (Model-View-Controller) и обрабатывает HTTP запросы от клиента, вызывая соответствующие методы сервисов и возвращая клиенту необходимую информацию. Контроллер предоставляет различные методы для обработки данных и взаимодействия с сервисом, использует сервис GroundingService и для выполнения расчетов.

Методы контроллера @GetMapping и @PostMapping используются для обработки HTTP GET и POST запросов соответственно.

Метод getGroundings отображает все объекты Grounding в представлении "groundings".

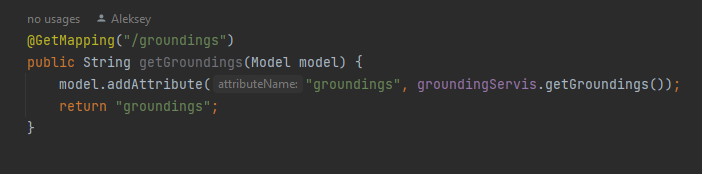


Рисунок 2.2.1. метод Get.

Метод PostGroundings добавляет значения вводимых переменных в Grounding и вызывает методы GroundingServis для расчета параметров заземления. См. рис. 2.2.1.

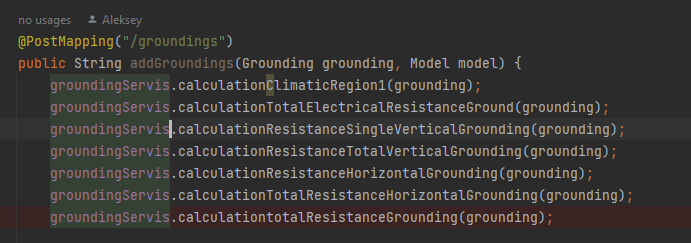


Рисунок 2.2.2. метод Post.

Метод addGroundings добавляет переданный объект Grounding в сервис. Метод возвращает представление "groundings" с обновленным списком объектов Grounding. См. рисунок 2.2.3

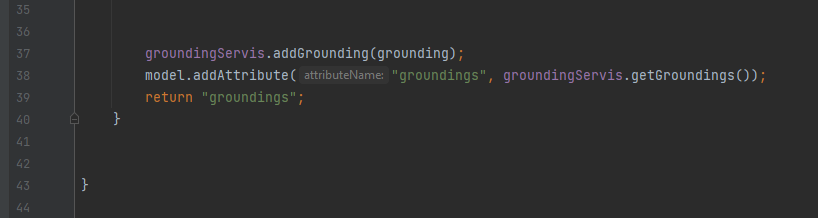


Рисунок 2.2.3 метод addGroundings.

Логгер Lombok @AllArgsConstructor генерирует конструктор с аргументами для всех полей в классе.

**2.3 Описание класса GroundingServis**

Аннотация @Service в Spring обозначает, что класс является сервисом, который содержит логику приложения. Она используется для пометки классов, которые предоставляют различные сервисы, такие как обработка данных, выполнение логики и управление транзакциями.

Класс GroundingServis реализует следующие методы :

addGrounding- добавление полученных значений в список. На данном этапе мы выводим только последний расчет. Для этого перед записью удаляем предыдущие значения. В дальнейшем мы сможем выводить несколько расчетов для сравнения и выбора оптимального значения. В методе реализована проверка на Null**.** См. рисунок 2.3.1.

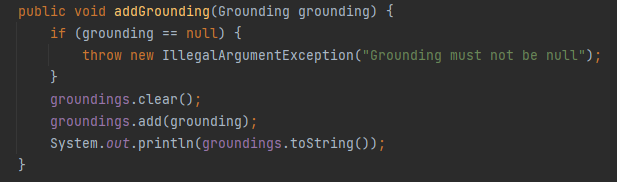


Рисунок 2.3.1. Метод addGrounding.

calculationСlimaticRegion1 – выбор климатического коэффициента из списка. См. рисунок 2.3.2.

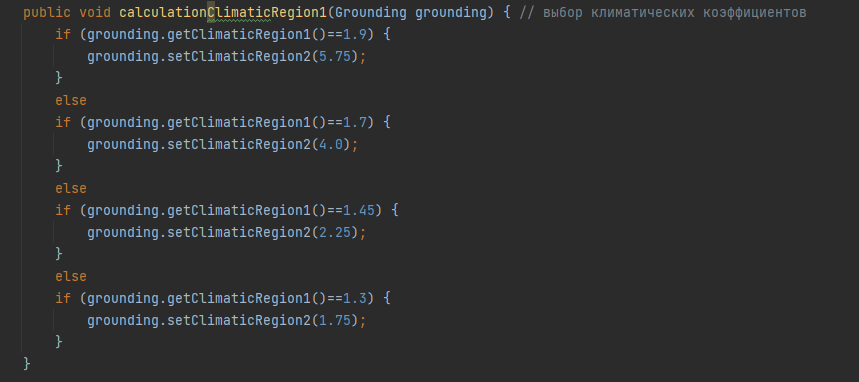


Рисунок 2.3.2. Метод calculationСlimaticRegion1.

calculationTotalElectricalResistanceGround – расчет удельного сопротивления грунта. В методе реализована проверка деления на «0». См. рисунок 2.3.3.

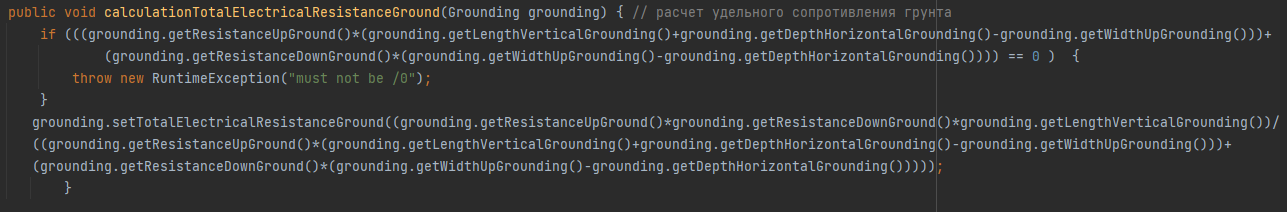


Рисунок 2.3.3. Метод calculationTotalElectricalResistanceGround.

calculationResistanceSingleVerticalGrounding – расчет сопротивления одиночного вертикального заземлителя. В методе реализована проверка деления на «0». См. рисунок 2.3.4.

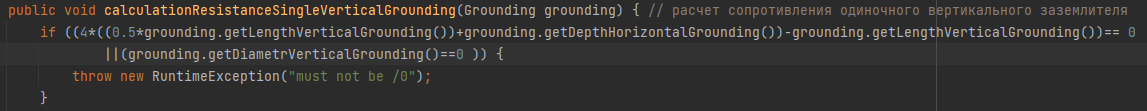


Рисунок 2.3.4. Метод calculationResistanceSingleVerticalGrounding.

calculationResistanceTotalVerticalGrounding – расчет общего сопротивления вертикальных заземлителей. В методе реализована проверка деления на «0». См. рисунок 2.3.5.

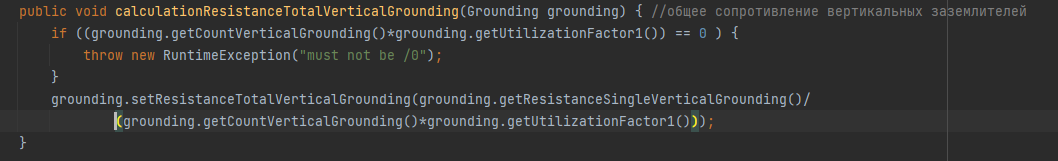


Рисунок 2.3.5. Метод calculationResistanceSingleVerticalGrounding.

calculationResistanceHorizontalGrounding – расчет сопротивления горизонтального заземлителя. В методе реализована проверка деления на «0». См. рисунок 2.3.6.

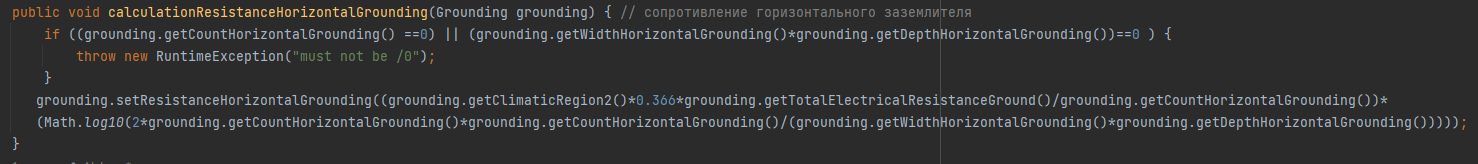


Рисунок 2.3.6. Метод calculationResistanceHorizontalGrounding.

calculationTotalResistanceHorizontalGrounding - расчет общего заземления горизонтального заземлителя. В методе реализована проверка деления на «0». См. рисунок 2.3.7.

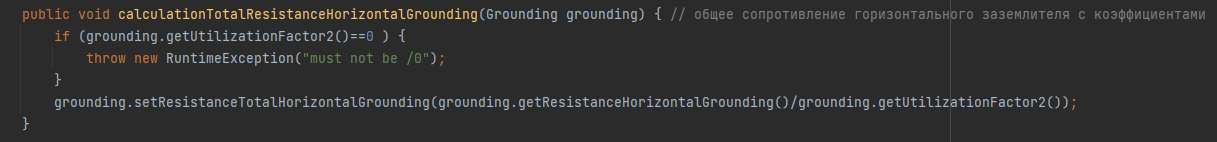


Рисунок 2.3.7. Метод calculationTotalResistanceHorizontalGrounding.

calculationtotalResistanceGrounding - общее сопротивление контура заземления. В методе реализована проверка деления на «0». См. рисунок 2.3.8.

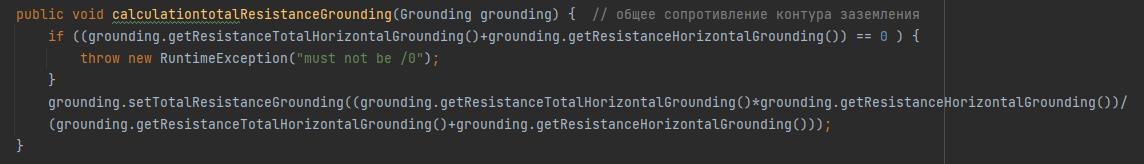


Рисунок 2.3.8. Метод calculationtotalResistanceGrounding.

getGroundings  - возвращает новую копию списка groundings, чтобы избежать изменений по ссылке на оригинальный список. В методе реализована проверка деления на «0». См. рисунок 2.3.9.

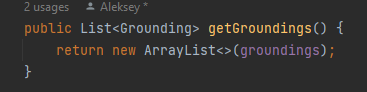


Рисунок 2.3.9. Метод getGroundings.

**2.4. Описание WEB –интерфейса.**

Для разработки WEB – интерфейса был использован язык гипертекстовой разметки- HTML. Основная цель HTML — структурировать и оформлять контент на сайте. Он создаёт иерархическую структуру веб-страницы, используя заголовки, абзацы, списки и таблицы. Такая структура помогает пользователю легче ориентироваться на сайте.

Данный HTML-код представляет собой шаблон для веб-страницы, использующей Thymeleaf в качестве шаблонизатора.Thymeleaf — современный серверный механизм Java-шаблонов для веб- и автономных сред, способный обрабатывать HTML, XML, JavaScript, CSS и даже простой текст. Основной целью Thymeleaf является создание элегантного и удобного способа шаблонизации.

Рассмотрим HTML файл groundings.html, реализующий ввод данных, отображение результатов расчетов, а также вывод таблиц с поправочными коэффициентами.

Рассмотрим основные компоненты страницы html:

<html lang="ru" – определение языка

xmlns:th=<http://www.thymeleaf.org>> подключение шаблонизатора thymeleaf

<title>Расчет сопротивления контура заземления</title> - заголовок страницы

<body> </body> - Тело страницы.

<h1> Расчет сопротивления контура заземления </h1> - Заголовок.

<table> </table> - Таблица.

<tr> </tr> - служит контейнером для создания строки таблицы

<td></td> - служит для создания

Рассмотрим основные узлы страницы :

1. Вставка картинки с поясняющей схемой . размеры картинки 900\*450. См.рисунок 2.4.1.

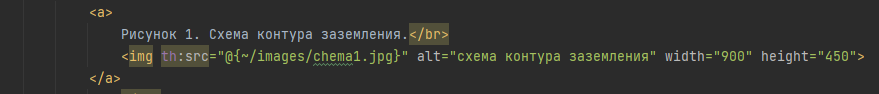


Рисунок 2.4.1. Вставка картинки на станицу.

1. Таблица выбора климатических коэффициентов в зависимости от среднегодовой температуры окружающего воздуха района. См.рисунок 2.4.2.

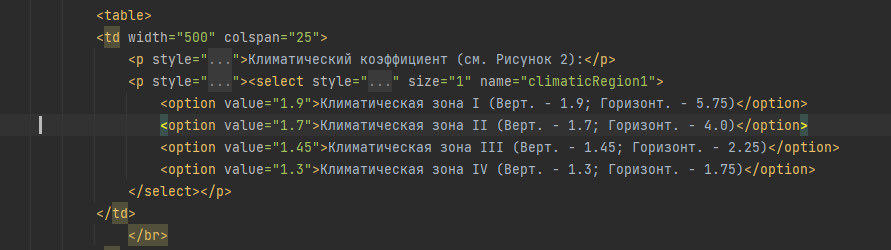


Рисунок 2.4.2. Таблица выбора климатических коэффициентов.

1. Таблица ввода значений для расчета контура заземления. Размеры ячейки таблицы 500\*25. См.рисунок 2.4.3.

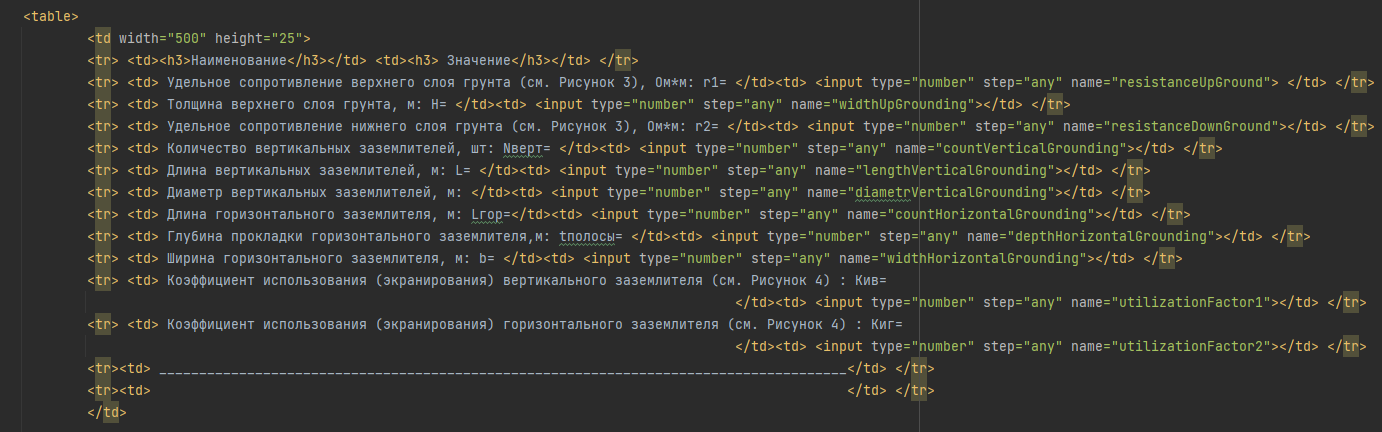


Рисунок 2.4.3. Таблица ввода значений для расчета контура заземления.

1. Кнопка для выполнения расчета . См.рисунок 2.4.4.



Рисунок 2.4.4. Кнопка для выполнения расчета.

1. Таблица вывода результатов расчета на экран. См.рисунок 2.4.5

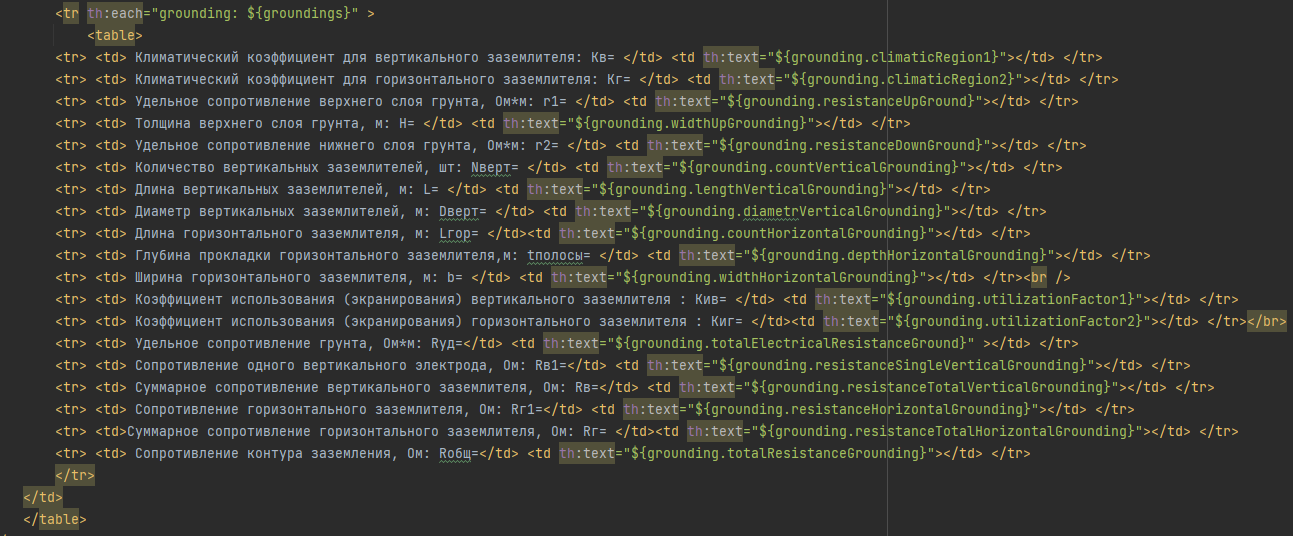


Рисунок 2.4.5. Таблица вывода результатов расчета на экран.

**ЗАКЛЮЧЕНИЕ**

В рамках данного дипломного проекта было разработано веб-приложение «Калькулятор расчета конура заземления», предназначенное для расчета, выбора материала требуемого контура заземления объекта. Проект был реализован с использованием современных технологий веб-разработки, таких как Java Spring Framework, Thymeleaf.

В процессе разработки были достигнуты следующие цели:

* Создание удобного и интуитивно понятного интерфейса для пользователей приложения.
* Использование принципов SOLID и паттернов проектирования для обеспечения гибкости и расширяемости кода.

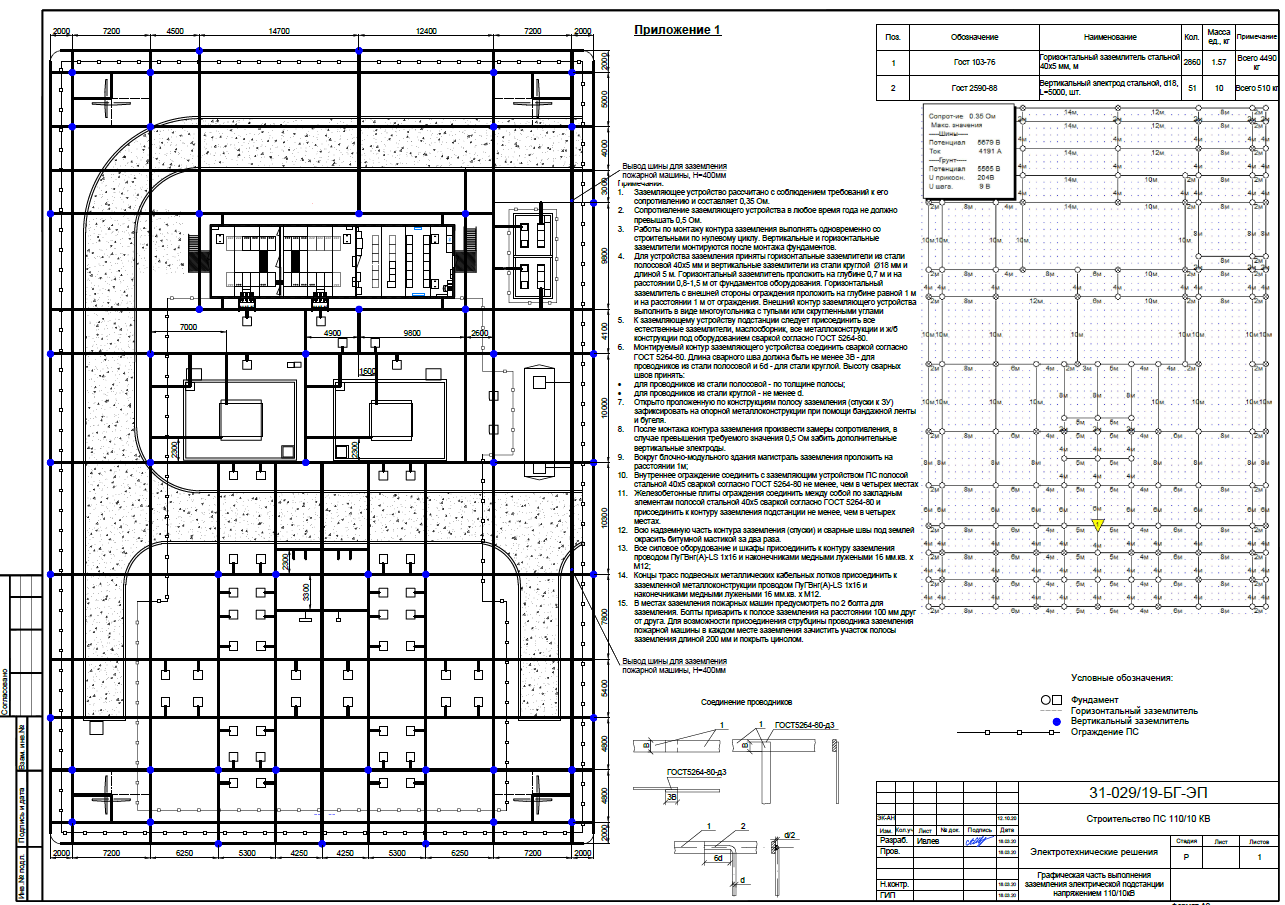
Дальнейшее развитие приложения может включать в себя добавление нового функционала, такого как вывод на экран формул, расчетов (решения задачи) для добавления в проект.

В целом, дипломный проект "Калькулятор расчетка контура заземления" позволил закрепить навыки разработки веб-приложений с использованием современных технологий и понять важность удобства использования для конечного пользователя.

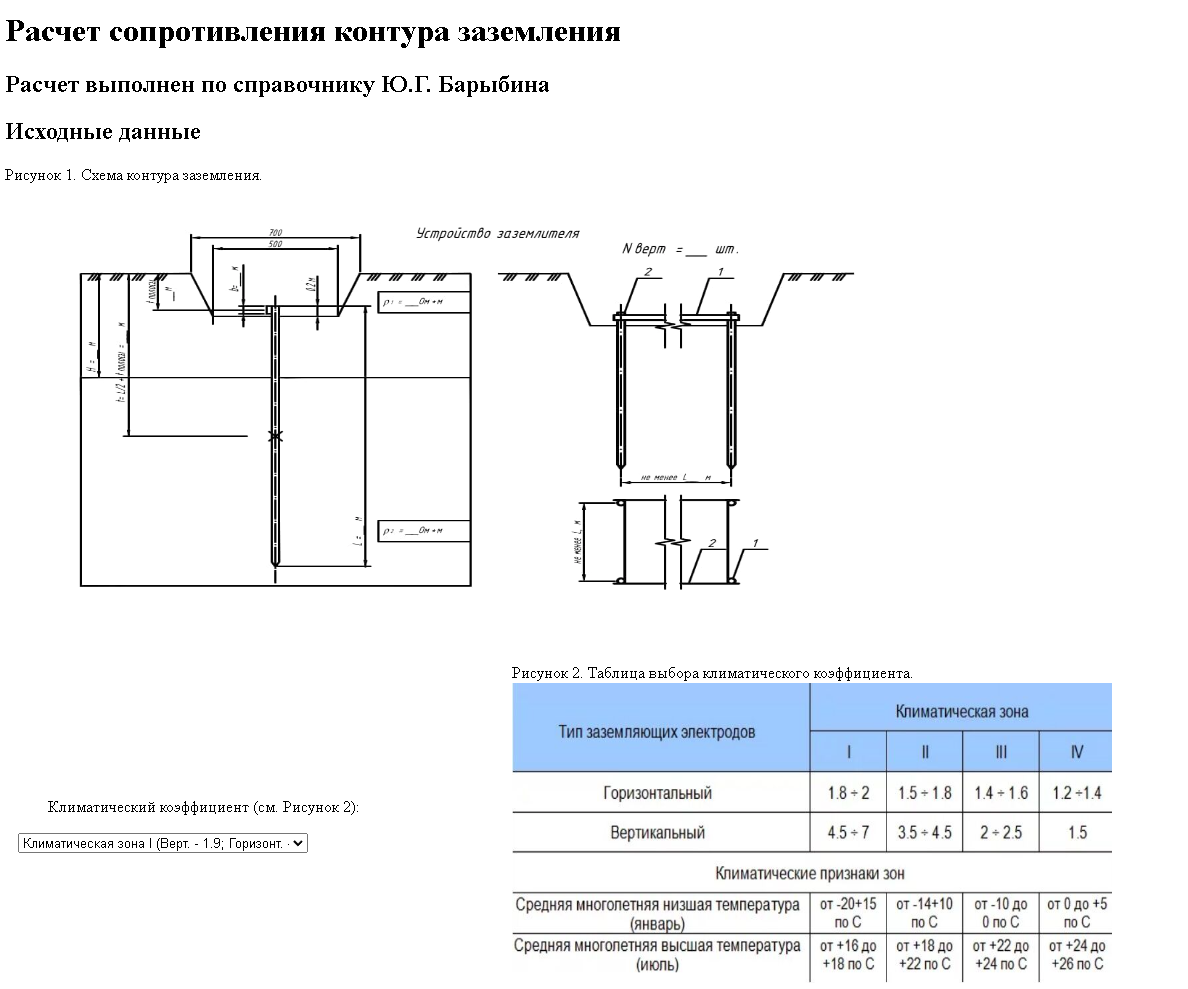
**Список использованной литературы**

1. Шилдт Герберт Java. Полное руководство. 12-е изд. : перевод с англ. – Ю.Н. Артеменко; СПБ: ООО «Диалектика», 2023. – 1344 с. : ил. – Парал. тит. англ.
2. Spring в действии**,** Крейг Уоллс (6-е издание, 2022 год).
3. Учебный материал Гикбрейнс.
4. Статьи и видео-материалы из сети интернет.
5. Справочник по проектированию электрических сетей и электрооборудования под редакцией Ю.Г. Барыбина, Л.Е. Федорова, М.Г. Зименкова, А.Г. Смрнова. Москва. Энергоатомиздат 1991г.

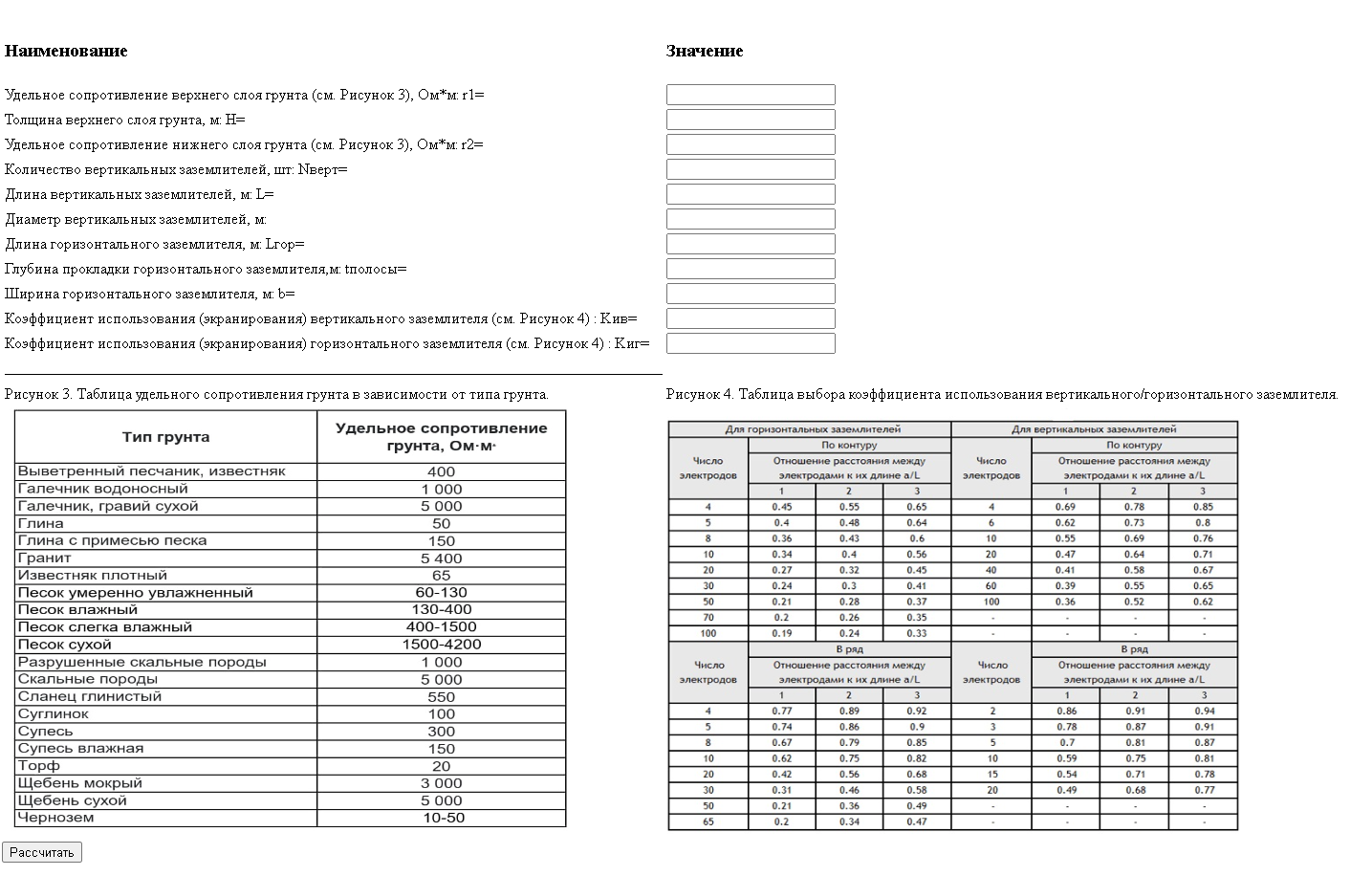
**Приложения**



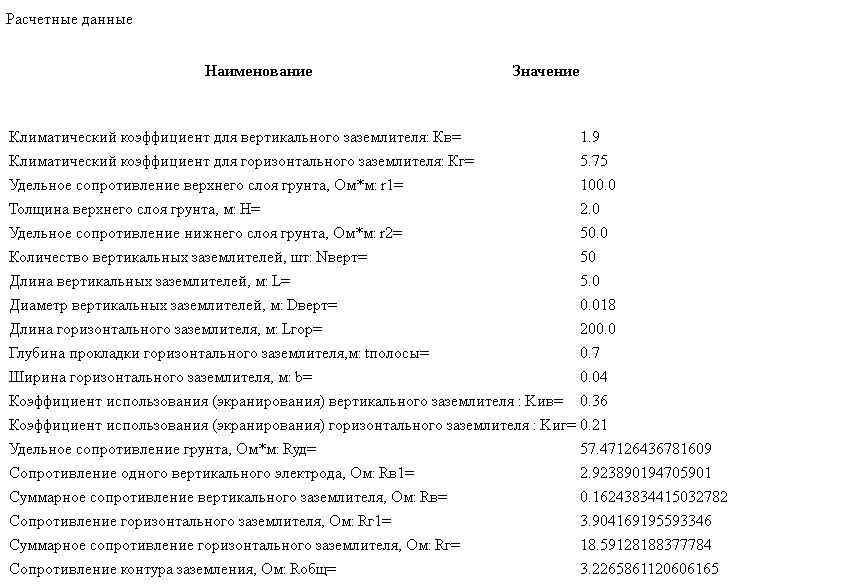
**Приложение 2. Интерфейс приложения. Исходные данные.**



**Приложение 3. Интерфейс приложения. Ввод данных, поправочные коэффициенты**



**Приложение 4. Интерфейс приложения. Расчетные значения.**



**Приложение 5. Код страницы Web-интерфейса**

<!DOCTYPE html>  
<html lang="ru" xmlns:th="http://www.thymeleaf.org" >  
 <head>  
 <meta charset="UTF-8">  
 <title>Расчет сопротивления контура заземления</title>  
 </head>  
 <body>  
 <h1>Расчет сопротивления контура заземления </h1>  
 <h2>Расчет выполнен по справочнику Ю.Г. Барыбина</h2>  
 <h2>Исходные данные</h2>  
 <form action="/groundings" method="post">  
 <a>  
 Рисунок 1. Схема контура заземления.</br>  
 <img th:src="@{~/images/chema1.jpg}" alt="схема контура заземления" width="900" height="450">  
 </a>  
 </br>  
  
 <table>  
 <td width="500" colspan="25">  
 <p style="margin-left: 40px">Климатический коэффициент (см. Рисунок 2):</p>  
 <p style="margin-left: 10px"><select style="width:290px" size="1" name="climaticRegion1">  
 <option value="1.9">Климатическая зона I (Верт. - 1.9; Горизонт. - 5.75)</option>  
 <option value="1.7">Климатическая зона II (Верт. - 1.7; Горизонт. - 4.0)</option>  
 <option value="1.45">Климатическая зона III (Верт. - 1.45; Горизонт. - 2.25)</option>  
 <option value="1.3">Климатическая зона IV (Верт. - 1.3; Горизонт. - 1.75)</option>  
 </select></p>  
 </td>  
 </br>  
 <td>  
 Рисунок 2. Таблица выбора климатического коэффициента.</br>  
 <img th:src="@{~/images/chema2.jpg}" alt="Таблица выбора климатического коэффициента" width="600" height="300"/>  
 </td>  
 </table>  
  
 <table>  
 <td width="500" height="25">  
 <tr> <td><h3>Наименование</h3></td> <td><h3> Значение</h3></td> </tr>  
 <tr> <td> Удельное сопротивление верхнего слоя грунта (см. Рисунок 3), Ом\*м: r1= </td><td> <input type="number" step="any" name="resistanceUpGround"> </td> </tr>  
 <tr> <td> Толщина верхнего слоя грунта, м: H= </td><td> <input type="number" step="any" name="widthUpGrounding"></td> </tr>  
 <tr> <td> Удельное сопротивление нижнего слоя грунта (см. Рисунок 3), Ом\*м: r2= </td><td> <input type="number" step="any" name="resistanceDownGround"></td> </tr>  
 <tr> <td> Количество вертикальных заземлителей, шт: Nверт= </td><td> <input type="number" step="any" name="countVerticalGrounding"></td> </tr>  
 <tr> <td> Длина вертикальных заземлителей, м: L= </td><td> <input type="number" step="any" name="lengthVerticalGrounding"></td> </tr>  
 <tr> <td> Диаметр вертикальных заземлителей, м: </td><td> <input type="number" step="any" name="diametrVerticalGrounding"></td> </tr>  
 <tr> <td> Длина горизонтального заземлителя, м: Lгор=</td><td> <input type="number" step="any" name="countHorizontalGrounding"></td> </tr>  
 <tr> <td> Глубина прокладки горизонтального заземлителя,м: tполосы= </td><td> <input type="number" step="any" name="depthHorizontalGrounding"></td> </tr>  
 <tr> <td> Ширина горизонтального заземлителя, м: b= </td><td> <input type="number" step="any" name="widthHorizontalGrounding"></td> </tr>  
 <tr> <td> Коэффициент использования (экранирования) вертикального заземлителя (см. Рисунок 4) : Kив=  
 </td><td> <input type="number" step="any" name="utilizationFactor1"></td> </tr>  
 <tr> <td> Коэффициент использования (экранирования) горизонтального заземлителя (см. Рисунок 4) : Kиг=  
 </td><td> <input type="number" step="any" name="utilizationFactor2"></td> </tr>  
 <tr><td> \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_</td> </tr>  
 <tr><td> </td> </tr>  
 </td>  
 <td>  
 Рисунок 3. Таблица удельного сопротивления грунта в зависимости от типа грунта.</br>  
 <img th:src="@{~/images/chema3.jpg}" alt="Таблица удельного сопротивления грунта в зависимости от типа грунта" width="600" height="450"/>  
 </td>  
 <td>  
 Рисунок 4. Таблица выбора коэффициента использования вертикального/горизонтального заземлителя.</br>  
 <img th:src="@{~/images/chema4.jpg}" alt="схема контура заземления" width="600" height="450"/>  
 </td>  
 </table>  
  
 <tr>  
 <button type="submit">Рассчитать</button>  
 </form>  
 </br>  
 <tr> Расчетные данные</tr>  
 <table>  
 <td width="500" height="25">  
 <tr>  
 <th> Наименование </th>  
 <th> Значение</th>  
 </tr>  
 </td>  
 </table>  
 <tr th:each="grounding: ${groundings}" >  
 <table>  
 <tr> <td> Климатический коэффициент для вертикального заземлителя: Кв= </td> <td th:text="${grounding.climaticRegion1}"></td> </tr>  
 <tr> <td> Климатический коэффициент для горизонтального заземлителя: Кг= </td> <td th:text="${grounding.climaticRegion2}"></td> </tr>  
 <tr> <td> Удельное сопротивление верхнего слоя грунта, Ом\*м: r1= </td> <td th:text="${grounding.resistanceUpGround}"></td> </tr>  
 <tr> <td> Толщина верхнего слоя грунта, м: H= </td> <td th:text="${grounding.widthUpGrounding}"></td> </tr>  
 <tr> <td> Удельное сопротивление нижнего слоя грунта, Ом\*м: r2= </td> <td th:text="${grounding.resistanceDownGround}"></td> </tr>  
 <tr> <td> Количество вертикальных заземлителей, шт: Nверт= </td> <td th:text="${grounding.countVerticalGrounding}"></td> </tr>  
 <tr> <td> Длина вертикальных заземлителей, м: L= </td> <td th:text="${grounding.lengthVerticalGrounding}"></td> </tr>  
 <tr> <td> Диаметр вертикальных заземлителей, м: Dверт= </td> <td th:text="${grounding.diametrVerticalGrounding}"></td> </tr>  
 <tr> <td> Длина горизонтального заземлителя, м: Lгор= </td><td th:text="${grounding.countHorizontalGrounding}"></td> </tr>  
 <tr> <td> Глубина прокладки горизонтального заземлителя,м: tполосы= </td> <td th:text="${grounding.depthHorizontalGrounding}"></td> </tr>  
 <tr> <td> Ширина горизонтального заземлителя, м: b= </td> <td th:text="${grounding.widthHorizontalGrounding}"></td> </tr><br />  
 <tr> <td> Коэффициент использования (экранирования) вертикального заземлителя : Kив= </td> <td th:text="${grounding.utilizationFactor1}"></td> </tr>  
 <tr> <td> Коэффициент использования (экранирования) горизонтального заземлителя : Kиг= </td><td th:text="${grounding.utilizationFactor2}"></td> </tr></br>  
 <tr> <td> Удельное сопротивление грунта, Ом\*м: Rуд=</td> <td th:text="${grounding.totalElectricalResistanceGround}" ></td> </tr>  
 <tr> <td> Сопротивление одного вертикального электрода, Ом: Rв1=</td> <td th:text="${grounding.resistanceSingleVerticalGrounding}"></td> </tr>  
 <tr> <td> Суммарное сопротивление вертикального заземлителя, Ом: Rв=</td> <td th:text="${grounding.resistanceTotalVerticalGrounding}"></td> </tr>  
 <tr> <td> Сопротивление горизонтального заземлителя, Ом: Rг1=</td> <td th:text="${grounding.resistanceHorizontalGrounding}"></td> </tr>  
 <tr> <td>Суммарное сопротивление горизонтального заземлителя, Ом: Rг= </td><td th:text="${grounding.resistanceTotalHorizontalGrounding}"></td> </tr>  
 <tr> <td> Сопротивление контура заземления, Ом: Rобщ=</td> <td th:text="${grounding.totalResistanceGrounding}"></td> </tr>  
 </tr>  
 </td>  
 </table>  
</body> </html>

**Приложение 6. Код модуля main**

package com.example.Diplom;  
  
import org.springframework.boot.SpringApplication;  
import org.springframework.boot.autoconfigure.SpringBootApplication;  
  
@SpringBootApplication  
public class Diplom extends Exception{  
  
 public static void main(String[] args) {  
 SpringApplication.*run*(Diplom.class, args);  
 }  
}

**Приложение 7. Код класса GroundingServis**

package com.example.Diplom.servis;  
import com.example.Diplom.models.Grounding;  
import org.springframework.stereotype.Service;  
import java.util.ArrayList;  
import java.util.List;  
@Service  
public class GroundingServis {  
 private final Grounding grounding = new Grounding();  
 List<Grounding> groundings = new ArrayList<>();  
 public void addGrounding(Grounding grounding) {  
 if (grounding == null) {  
 throw new IllegalArgumentException("Grounding must not be null");  
 }  
 groundings.clear();  
 groundings.add(grounding);  
 System.*out*.println(groundings.toString());  
 }  
 public void calculationСlimaticRegion1(Grounding grounding) { // выбор климатических коэффициентов  
 if (grounding.getClimaticRegion1()==1.9) {  
 grounding.setClimaticRegion2(5.75);  
 }  
 else  
 if (grounding.getClimaticRegion1()==1.7) {  
 grounding.setClimaticRegion2(4.0);  
 }  
 else  
 if (grounding.getClimaticRegion1()==1.45) {  
 grounding.setClimaticRegion2(2.25);  
 }  
 else  
 if (grounding.getClimaticRegion1()==1.3) {  
 grounding.setClimaticRegion2(1.75);  
 }  
 }  
 public void calculationTotalElectricalResistanceGround(Grounding grounding) { // расчет удельного сопротивления грунта  
 if (((grounding.getResistanceUpGround()\*(grounding.getLengthVerticalGrounding()+grounding.getDepthHorizontalGrounding()-grounding.getWidthUpGrounding()))+  
 (grounding.getResistanceDownGround()\*(grounding.getWidthUpGrounding()-grounding.getDepthHorizontalGrounding()))) == 0 ) {  
 throw new RuntimeException("must not be /0");  
 }  
 grounding.setTotalElectricalResistanceGround((grounding.getResistanceUpGround()\*grounding.getResistanceDownGround()\*grounding.getLengthVerticalGrounding())/  
 ((grounding.getResistanceUpGround()\*(grounding.getLengthVerticalGrounding()+grounding.getDepthHorizontalGrounding()-grounding.getWidthUpGrounding()))+  
 (grounding.getResistanceDownGround()\*(grounding.getWidthUpGrounding()-grounding.getDepthHorizontalGrounding()))));  
 }  
  
 public void calculationResistanceSingleVerticalGrounding(Grounding grounding) { // расчет сопротивления одиночного вертикального заземлителя  
 if ((4\*((0.5\*grounding.getLengthVerticalGrounding())+grounding.getDepthHorizontalGrounding())-grounding.getLengthVerticalGrounding())== 0  
 ||(grounding.getDiametrVerticalGrounding()==0 )) {  
 throw new RuntimeException("must not be /0");  
 }  
 grounding.setResistanceSingleVerticalGrounding((Math.*log10*(2\*grounding.getLengthVerticalGrounding()/grounding.getDiametrVerticalGrounding())+  
 0.5\*(Math.*log10*((4\*((0.5\*grounding.getLengthVerticalGrounding())+grounding.getDepthHorizontalGrounding())+grounding.getLengthVerticalGrounding())/  
 (4\*((0.5\*grounding.getLengthVerticalGrounding())+grounding.getDepthHorizontalGrounding())-grounding.getLengthVerticalGrounding())))));  
 }  
 public void calculationResistanceTotalVerticalGrounding(Grounding grounding) { //общее сопротивление вертикальных заземлителей  
 if ((grounding.getCountVerticalGrounding()\*grounding.getUtilizationFactor1()) == 0 ) {  
 throw new RuntimeException("must not be /0");  
 }  
 grounding.setResistanceTotalVerticalGrounding(grounding.getResistanceSingleVerticalGrounding()/ (grounding.getCountVerticalGrounding()\*grounding.getUtilizationFactor1()));  
 }  
 public void calculationResistanceHorizontalGrounding(Grounding grounding) { // сопротивление горизонтального заземлителя  
 if ((grounding.getCountHorizontalGrounding() ==0) || (grounding.getWidthHorizontalGrounding()\*grounding.getDepthHorizontalGrounding())==0 ) {  
 throw new RuntimeException("must not be /0");  
 }  
 grounding.setResistanceHorizontalGrounding((grounding.getClimaticRegion2()\*0.366\*grounding.getTotalElectricalResistanceGround()/grounding.getCountHorizontalGrounding())\*  
 (Math.*log10*(2\*grounding.getCountHorizontalGrounding()\*grounding.getCountHorizontalGrounding()/(grounding.getWidthHorizontalGrounding()\*grounding.getDepthHorizontalGrounding()))));  
 }  
 public void calculationTotalResistanceHorizontalGrounding(Grounding grounding) { // общее сопротивление горизонтального заземлителя с коэффициентами  
 if (grounding.getUtilizationFactor2()==0 ) {  
 throw new RuntimeException("must not be /0");  
 }  
 grounding.setResistanceTotalHorizontalGrounding(grounding.getResistanceHorizontalGrounding()/grounding.getUtilizationFactor2());  
 }  
 public void calculationtotalResistanceGrounding(Grounding grounding) { // общее сопротивление контура заземления  
 if ((grounding.getResistanceTotalHorizontalGrounding()+grounding.getResistanceHorizontalGrounding()) == 0 ) {  
 throw new RuntimeException("must not be /0");  
 }  
 grounding.setTotalResistanceGrounding((grounding.getResistanceTotalHorizontalGrounding()\*grounding.getResistanceHorizontalGrounding())/  
 (grounding.getResistanceTotalHorizontalGrounding()+grounding.getResistanceHorizontalGrounding()));  
 }  
 public List<Grounding> getGroundings() {  
 return new ArrayList<>(groundings);  
 }  
}

**Приложение 8. Код класа Grounding**

package com.example.Diplom.models;  
import lombok.Data;  
@Data  
public class Grounding {  
 private double climaticRegion1; //Климатический коэффициент для вертикального заземлителя  
 private double climaticRegion2; //Климатический коэффициент для горизонтального заземлителя  
 private double resistanceUpGround; // Удельное сопротивление верхнего слоя грунта, Ом\*м: r1  
 private double widthUpGrounding; //Толщина верхнего слоя грунта, м: H  
 private double resistanceDownGround; //Удельное сопротивление нижнего слоя грунта, Ом\*м: r2  
 private int countVerticalGrounding; //Количество вертикальных заземлителей, шт: Nверт  
 private double lengthVerticalGrounding; //Длина вертикальных заземлителей, м: L  
 private double diametrVerticalGrounding; // Диаметр вертикальных заземлителей, м: Dверт  
 private double countHorizontalGrounding; // Длина горизонтального заземлителя, м: Lгор  
 private double depthHorizontalGrounding; //Глубина прокладки горизонтального заземлителя,м: tполосы  
 private double widthHorizontalGrounding ; //Ширина горизонтального заземлителя, м: b  
 private double utilizationFactor1; //коэффициент использования (экранирования ) вертикального заземлителя Kиг  
 private double utilizationFactor2; //коэффициент использования (экранирования ) горизонтального заземлителя Kиг  
 private double totalElectricalResistanceGround; //Удельное сопротивление грунта, Ом\*м: Rуд  
 private double resistanceSingleVerticalGrounding; //Сопротивление одного вертикального электрода, Ом: Rв1  
 private double resistanceTotalVerticalGrounding; //Суммарное сопротивление вертикального заземлителя, Ом: Rв  
 private double resistanceHorizontalGrounding; //Сопротивление горизонтального заземлителя, Ом: Rг1  
 private double resistanceTotalHorizontalGrounding; //Суммарное сопротивление горизонтального заземлителя, Ом: Rг  
 private double totalResistanceGrounding; //Сопротивление контура заземления, Ом: Rобщ  
}

**Приложение 9. Код класса GroundingController**

package com.example.Diplom.controllers;  
  
import lombok.AllArgsConstructor;  
import org.springframework.stereotype.Controller;  
import org.springframework.ui.Model;  
import org.springframework.web.bind.annotation.GetMapping;  
import org.springframework.web.bind.annotation.PostMapping;  
import com.example.Diplom.models.Grounding;  
import com.example.Diplom.servis.GroundingServis;  
  
@Controller  
@AllArgsConstructor  
public class GroundingController extends Exception{  
 public final GroundingServis groundingServis;  
  
 @GetMapping("/groundings")  
 public String getGroundings(Model model) {  
 model.addAttribute("groundings", groundingServis.getGroundings());  
 return "groundings";  
 }  
 @PostMapping("/groundings")  
 public String addGroundings(Grounding grounding, Model model) {  
 groundingServis.calculationСlimaticRegion1(grounding);  
 groundingServis.calculationTotalElectricalResistanceGround(grounding);  
 groundingServis.calculationResistanceSingleVerticalGrounding(grounding);  
 groundingServis.calculationResistanceTotalVerticalGrounding(grounding);  
 groundingServis.calculationResistanceHorizontalGrounding(grounding);  
 groundingServis.calculationTotalResistanceHorizontalGrounding(grounding);  
 groundingServis.calculationtotalResistanceGrounding(grounding);  
 groundingServis.addGrounding(grounding);  
 model.addAttribute("groundings", groundingServis.getGroundings());  
 return "groundings";  
 }  
}