Определения. Обозначения и сокращения

Введение (актуальность)

(Основная часть, три под части)

1. Постановка задачи и обоснование метода решения задачи

1.1 Описание предметной области и постановка задачи (цели и задачи)

1.1.1 OSI

1.1.2 Классификация угроз

1.1.3 Анализ вектора инцидентов

1.2 Анализ существующих методов решения задач, привести конкретные устройства фаерволов/сканеров, их классификация

1.3 CISCO, Microtic

2. проектирование и программная реализация метода решения задачи

2.1 Обоснование и выбор комплекса программных средств, ЯП, ОС.

2.2 Описание информационно-логической модели системы

2.2.1 Архитектура системы

2.2.4 Диаграмма размещения UML

2.3 Проектирование интерфейса пользователя

2.3.1 Инструменты и технологии

2.3.2 Прототип пользовательского интерфейса

2.4 Программная реализация

3. Вычислительные эксперименты, проверка эффективности реализованного метода решения задачи.

3.1 Разработка контрольного примера реальной обработки данных

3.2 Проверка эффективности разработанной системы для решения поставленной задачи

Заключение

Список используемых источников

Приложение 1 (листинг (можно только main))

Титульник

Тема:

**Разработка веб-интерфейса для авторизации и управления сетевым устройством**

Оглавление

ВВЕДЕНИЕ

1. Постановка задачи и обоснование метода решения задачи

1.1 Описание предметной области и постановка задачи

1. 2. Анализ предметной области

2. Проектирование и программная реализация метода решения задачи

2.1 Обоснование и выбор комплекса программных средств, ЯП, ОС.

2.2 …

2.3 Проектирование интерфейса пользователя

2.3.1 Инструменты и технологии

2.3.2 Прототип пользовательского интерфейса

2.4 Программная реализация

**ВВЕДЕНИЕ**

На сегодняшний день существует множество сетевых устройств, созданных для различных целей. Сложно представить современную жизнь без таких устройств, как маршрутизаторов, коммутаторов, межсетевых экранов.

В большинстве своем, многие сетевые устройства работают по своим определенным протоколам, у них как правило имеется своя файловая система, операционная система и пользовательский интерфейс.

Многие производители сетевых устройств предпочитают не тратить лишние средства на оформления пользовательского интерфейса, а зачастую реализуют его в виде команд в терминале. Это, конечно, имеет свои преимущества для тех, кто хорошо разбирается в администрировании таких устройств и как правило для обычных пользователей этот интерфейс является не очень понятным.

Поэтому всё чаще стали появляться устройства, управление которыми осуществляется через графический интерфейс, это может быть окно в Windows или Linux системах, или WEB-страница. В основном, в качестве графического интерфейса используются web-страницы, т.к они универсальны, их можно просматривать с любого компьютера, телефона, планшета, под любой операционной системой, в любом браузере.

**2. Проектирование и программная реализация метода решения задачи**

**2.1 Обоснование и выбор комплекса программных средств, ЯП, ОС.**

В качестве языка программирования был выбран язык Java. В отличие от программ написанных на Python программы на Java обладают большей производительностью, по сравнению с языком Go Java давно используется программистами, имеет большое сообщество, хорошую документацию, множество популярных фреймворков. JavaScript проигрывает Java в безопасности. Для упрощения и ускорения процесса разработки был выбран фреймворк Spring. Данный фрейворк является универсальным и используется для создания бизнес-приложений.

Операционная система – Debian.

Debian отличается широким спектром возможностей. В текущую стабильную версию включено свыше пятидесяти девяти тысяч пакетов программ для десяти архитектур на основе ядра Linux.

Отличительными чертами Debian являются: система управления пакетами Advanced Packaging Tool (APT), жёсткая политика по отношению к пакетам, [репозитории](https://ru.wikipedia.org/wiki/%D0%A0%D0%B5%D0%BF%D0%BE%D0%B7%D0%B8%D1%82%D0%BE%D1%80%D0%B8%D0%B9) с огромным их количеством, а также высокое качество выпускаемых версий. Это сделало возможным простое обновление между версиями, а также автоматическую установку и удаление пакетов. Именно в Debian впервые был введён как единый стандарт механизм выбора предпочтительного ПО среди нескольких вариантов — Alternatives.

Также Debian является единственным дистрибутивом со смешанной системой обновлений — как традиционная с жизненным циклом, так и модель Rolling release. Переключение между моделями реализовано по кодовым именам в APT. Для Rolling Release прописывается в репозиторий имя ветки, например stable, а для традиционного жизненного цикла — имя дистрибутива, например stretch.

**2.3 Проектирование интерфейса пользователя**

**2.3.1 Инструменты и технологии**

**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. Чтобы достичь этого, Thymeleaf основывается на концепции Natural Templates, чтобы внедрить свою логику в файлы шаблонов таким образом, чтобы этот шаблон не влиял на отображение прототипа дизайна. Это улучшает коммуникацию в команде и уменьшает разрыв между дизайнерско-программистскими группами.  
  
Thymeleaf также был разработан с самого начала с учетом стандартов Web, особенно **HTML5**, что позволяет создавать полностью соответствующие стандарту шаблоны.

Из коробки, Thymeleaf позволяет работать с шестью типами шаблонов, каждый из которых называется «Формат шаблона/Template Mode»:

* HTML
* XML
* TEXT
* JAVASCRIPT
* CSS
* RAW

**Bootstrap** — свободный набор инструментов для создания сайтов и веб-приложений. Включает в себя HTML- и CSS-шаблоны оформления для типографики, веб-форм, кнопок, меток, блоков навигации и прочих компонентов веб-интерфейса, включая JavaScript-расширения.

На рынке много устройств, которые имеют веб интерфейс