Федеральное государственное автономное
образовательное учреждение
Высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий
институт
Вычислительная техника
кафедра

УТВЕРЖДАЮ
Заведующий кафедрой
_____ О.В. Непомнящий
подпись инициалы, фамилия
«_____» _____ 2022 г.

БАКАЛАВАРСКАЯ РАБОТА

09.03.01 Информатика и вычислительная техника
код и наименование специальности

Библиотека тестирования телекоммуникационного оборудования
тема

Руководитель	_____	Старший преподаватель	И.Н. Рыженко
	подпись, дата	должность, ученая степень	инициалы, фамилия
Выпускник	_____		А.Н. Цуба
	подпись, дата		инициалы, фамилия
Нормоконтролер	_____	Старший преподаватель	И.Н. Рыженко
	подпись, дата	должность, ученая степень	инициалы, фамилия

Красноярск 2022

Министерство науки и высшего образования РФ
Федеральное государственное автономное образовательное учреждение высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий
институт
Вычислительная техника
кафедра

УТВЕРЖДАЮ
Заведующий кафедрой
_____ О.В. Непомнящий
подпись инициалы, фамилия
«_____» _____ 2022 г.

ЗАДАНИЕ
НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ
в форме бакалаврской работы

Красноярск 2022

Студенту Цуба Андрею Николаевичу
фамилия, имя, отчество

Группа КИ18-08Б направление (специальность) 09.03.01
номер код

Информатика и вычислительная техника
наименование

Тема выпускной квалификационной работы Библиотека тестирования телекоммуникационного оборудования

Утверждена приказом по университету № _____ от _____

Руководитель ВКР И.Н. Рыженко, старший преподаватель
инициалы, фамилия, должность, ученое звание и место работы

Исходные данные для ВКР: сформулировать цели и задачи, провести анализ существующих аналогов в предметной области, реализовать библиотеку тестирования телекоммуникационного оборудования.

Перечень разделов ВКР: анализ задания на выполнения и обзор аналогов, проектирование библиотеки тестирования телекоммуникационного оборудования, программная реализация.

Перечень графического материала: презентация в формате PowerPoint.

Руководитель _____ И. Н. Рыженко
Подпись, дата Инициалы и фамилия

Задание принял к исполнению А. Н. Цуба
Подпись, инициалы и фамилия студента

«____» _____ 2022 г.

РЕФЕРАТ

Выпускная квалификационная работа по теме «Библиотека тестирования телекоммуникационного оборудования» содержит 37 страниц, исследовано 11 источников, использован 21 рисунок.

БИБЛИОТЕКА, КЛИЕНТ, СЕРВЕР, SSH, ТЕЛЕКОММУНИКАЦИОННОЕ ОБОРУДОВАНИЕ.

Цель работы: написание программного обеспечения для удаленного тестирования и конфигурирования телекоммуникационного оборудования на этапе производства.

При выполнении данной работы был произведен обзор предметной области, задания на выпускную квалификационную работу, изучены существующие аналоги и сформированы требования, предъявляемые к программному обеспечению.

Объект работы – библиотека, позволяющая осуществить удаленное подключение к спутниковому модему, посредством SSH-протокола, для дальнейшего конфигурирования оборудования и управления им.

Задачи:

- осуществить выбор программных средств моделирования и разработки программного обеспечения;
- выполнить моделирование разрабатываемого программного обеспечения;
- выполнить программную реализацию библиотеки для тестирования телекоммуникационного оборудования;

проанализировать полученные результаты работы;

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Что такое телекоммуникация[8] и посредством чего она осуществляется?

Телекоммуникация — это связь, которая осуществляется при помощи электронного оборудования такого, как телефоны, компьютерные модемы, спутники, волоконно-оптические кабели и т.д.

В период с начала до середины 20 века появились такие нововведения, как телефонный обмен, электромеханические коммутаторные системы, кабели, ретрансляторы, несущие системы, микроволновое оборудование, а потом в индустриально развитых районах мира начали распространяться телекоммуникационные системы.

С 1950-х годов до 1984 года в этой отрасли продолжали развиваться новые технологии. Например, спутниковые и усовершенствованные кабельные системы, цифровая и волоконно-оптическая технологии и видеотелефонная связь.

Отрасль коммуникаций была полностью компьютеризирована. Все эти модификации способствовали распространению телекоммуникационных систем по всем странам мира.

В 1984 году решением суда в Соединенных Штатах была разрушена монополия корпорации Американский телеграф и телефон (AT&T). Это событие совпало со многими крупными изменениями в технологии самой телекоммуникационной отрасли.

До 1980-х годов практически во всех странах считалось, что телекоммуникационные службы являются службами общественными и работают в законодательных рамках, обеспечивающих монопольное положение.

Вместе с ростом экономической активности наступление новых технологий привело к приватизации телекоммуникационной индустрии.

Эта тенденция достигла своей кульминации, когда AT&T лишилась своего монопольного положения, и прекратилось государственное регулирование телекоммуникационных системам США. В некоторых других странах сейчас происходят похожие приватизационные процессы.

После 1984 года в результате технического прогресса распространились телекоммуникационные системы, способные обеспечить универсальные услуги людям по всему миру.

1. Анализ задания для выполнения

Необходимо разработать библиотеку для тестирования телекоммуникационного оборудования, при помощи конфигурации.

Для обеспечения конкурентоспособности программного обеспечения, выполнен анализ сильных и слабых сторон аналогичных систем.

С их учетом составлены спецификации требований к программному обеспечению.

Данное программное обеспечение представляет собой модель системы – клиент-серверного взаимодействия[6], которое выступает в качестве API.

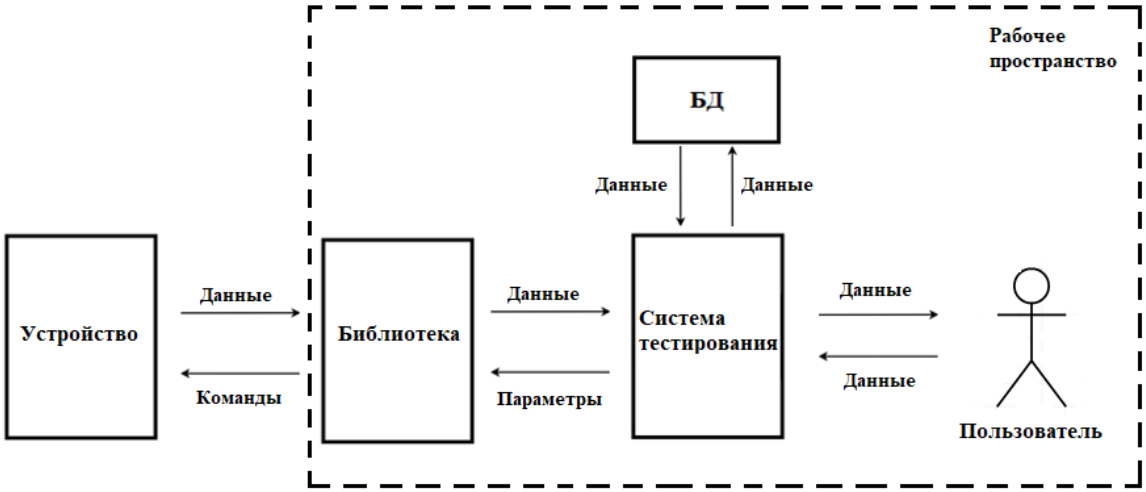

Рисунок 1 – Клиент-серверное взаимодействие

1.1 Анализ существующих аналогов

На данный момент уже существуют действующие автоматизированные системы управления и отслеживания статусов телекоммуникационного оборудования, рассмотрим некоторые из них, чтобы проанализировать преимущества и недостатки.

1.1.1 Информационно-графическая система «Кросс-про»

«Кросс-про» - отечественный кроссплатформенный программный продукт, способный обеспечить централизованный технический и

инвентаризационный учет технологических систем (телекоммуникационных сетей связи, инженерные сети, электрические сети и т.д.) с возможностью их мониторинга и управления.

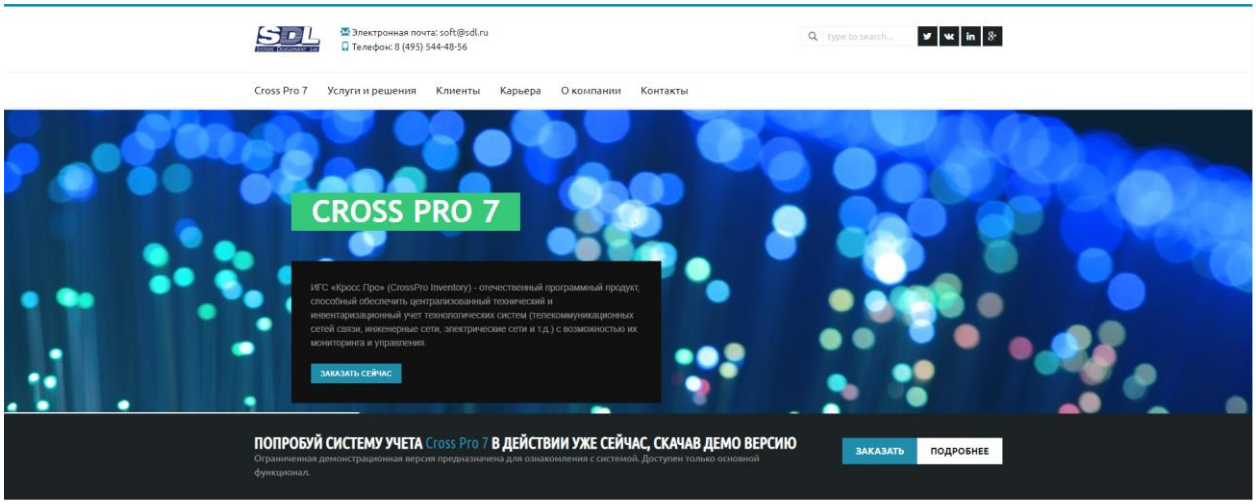

Рисунок 2 – Сайт сервиса «Кросс-про»

1.1.2 Система мониторинга и отслеживания «Zabbix»

свободная система мониторинга и отслеживания статусов разнообразных сервисов компьютерной сети, серверов и сетевого оборудования

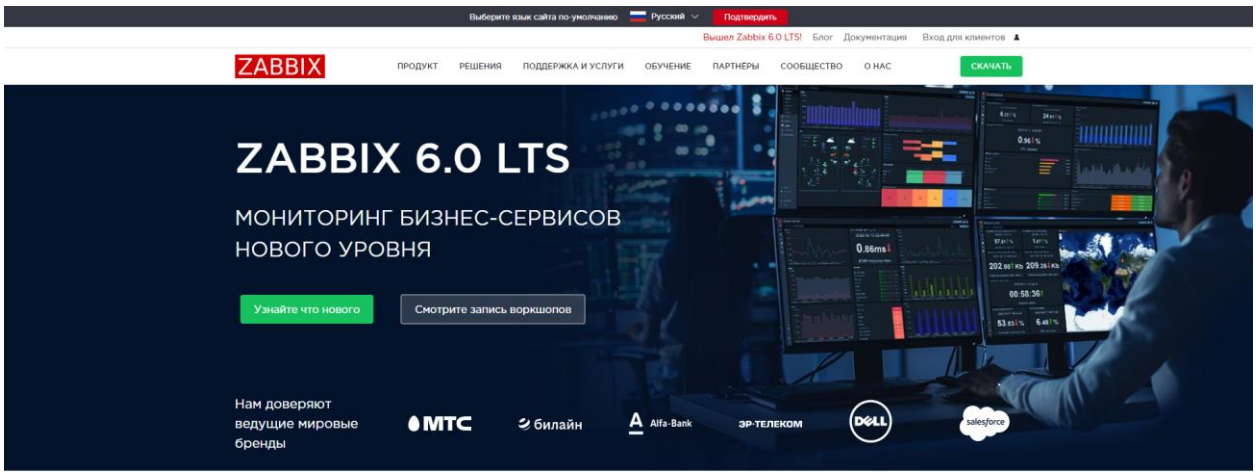

Рисунок 3 – Сайт сервиса «Zabbix»

1.1.3 программа «PRTG – Network Monitor»

программа, предназначенная для мониторинга использования сети, работает в семействе операционных систем Windows. Возможности программы сбор информации о потоках данных, проходящих через конкретные устройства, с сохранением её в базе данных программы.

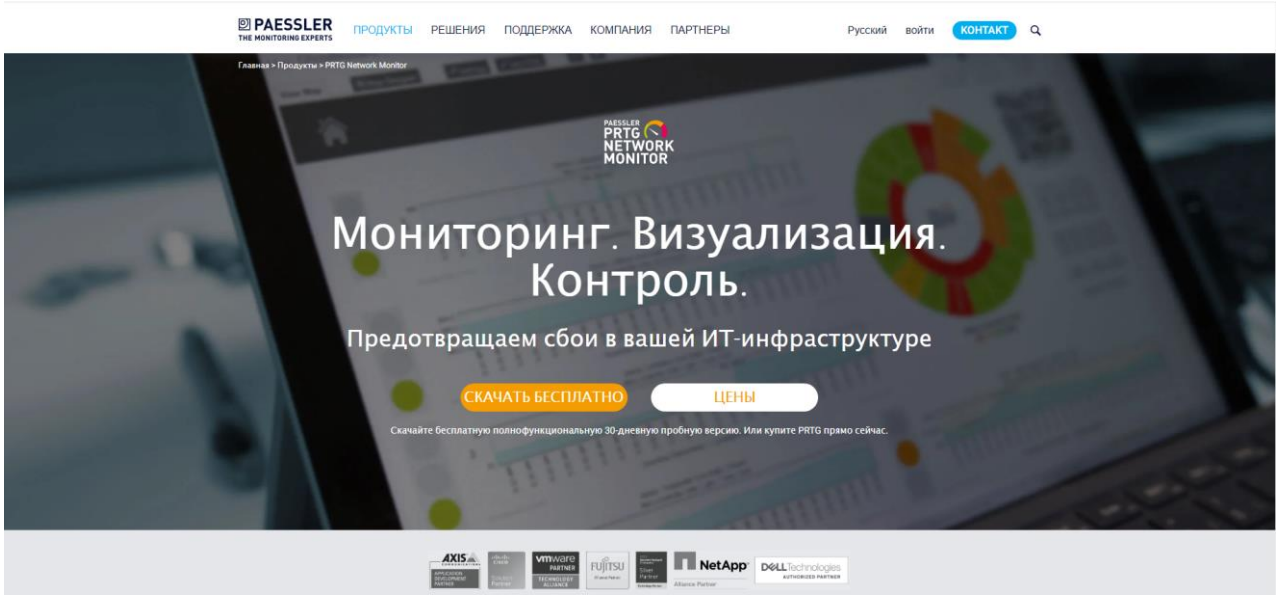

Рисунок 4 – Сайт сервиса «PRTG»

Вывод: были проанализированы сервисы, предоставляющие автоматизированные системы тестирования телекоммуникационного оборудования.

Минусом данных систем является то, что данное программное обеспечение не поставляется в бесплатном виде, а также требует определенных навыков, что влечет за собой обучение сотрудников для дальнейшей работы, также данные системы используются в основном для мониторинга на этапе эксплуатации.

Преимуществом разрабатываемой мной библиотеки является то, что она позволяет осуществлять тестирование на этапе производства телекоммуникационного оборудования, так же нет привязки к конкретному устройству, для использования на большинстве современных устройствах достаточно изменить команды в файле тестирования.

1.2 Спецификация требований на программное обеспечение

1.2.1 Функциональные требования

Библиотека тестирования является составной частью целой системы для мониторинга и тестирования телекоммуникационного оборудования.

Необходимо разработать библиотеку тестирования телекоммуникационного оборудования, реализующую следующий функционал:

- Установка, поддерживание и завершение соединения;
- Выполнение команд и получение результата;
- Редактирование полученных данных;
- Переключение между режимами управления (обычным и конфигурирования);

1.3 Выбор инструментов

1.3.1 Выбор языка программирования для разработки библиотеки

Для реализации поставленной передо мной задачи подходит множество современных языков программирования, например таких как: C,C++,C#,Python и т.д.

Все представленные языки являются объектно-ориентированными, поддерживают различные платформы, и имеют огромную базу библиотек и фреймворков для улучшения разработки.

При выборе языка для разработки, я исходил из следующих факторов:

- Простота реализации поставленной задачи;
- Поддерживаемость языка сообществом;
- Переносимость написанных программ.

Исходя из выше перечисленных факторов, в качестве языка программирования для реализации данной библиотеки мной был выбран Python[4].

Данный язык является объектно-ориентированным, высокоуровневый языком с динамической строгой типизацией и автоматическим управлением памятью, который ориентирован на повышение производительности разработчика, читаемости кода и его качества, а также на обеспечение переносимости написанных на нём программ.

Python очень популярен, благодаря чему имеет огромную поддержку сообществом, что позволяет использовать огромное количество подключаемых как стандартных библиотек и модулей, так и сторонних для оптимизации процесса разработки.

Например, таких как: NumPy, Pandas, TensorFlow и многие другие. Не маловажным фактором также является то, что данный язык поддерживает

регулярные выражения, благодаря которым очень легко редактировать огромное количество данных.

Так же язык является интерпретируемым и используется в том числе для написания скриптов, что очень упростит написание библиотеки тестирования телекоммуникационного оборудования.

Недостатками языка являются зачастую более низкая скорость работы и более высокое потребление памяти написанных на нём программ по сравнению с аналогичным кодом, написанным на компилируемых языках, например таких как C или C++.

1.3.2 Модуль Paramiko

Также для упрощения реализации, в качестве основного модуля для реализации сетевого взаимодействия мной был выбран модуль Paramiko, который является реализацией протокола SSHv2 на Python.

Так же данный модуль предоставляет функциональность клиента и сервера и не входит в стандартную библиотеку модулей Python, что позволит сконцентрировать силы на написание и решение поставленных задач.

1.4 Вывод по главе

В результате анализа поставленной задачи, были сформулированы четкие требования к разрабатываемому программному обеспечению.

Так же был произведен анализ существующих решений и выявлены их преимущества и недостатки.

Для того, чтобы упростить разработку был сформирован стек используемых технологий, а именно Python как основной язык, Paramiko как основное средство, позволяющее реализовать сетевое взаимодействие.

2. Проектирование

На рисунке 5 приведён основной алгоритм работы библиотеки в виде блок-схемы.

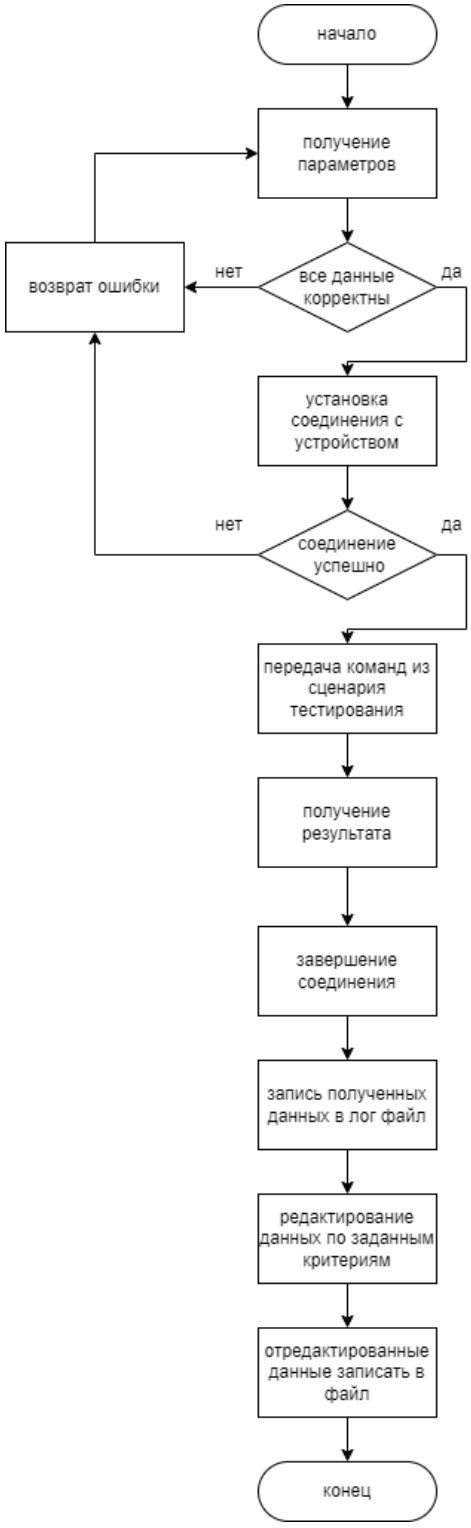

Рисунок 5 – Блок-схема работы библиотеки

2.1 Диаграмма последовательности

Вся логика работы библиотеки завязана на взаимодействии системы тестирования с тестируемым оборудованием, в разделе представлены диаграммы последовательностей, для наиболее значимых прецедентов.

На рисунке 6 приведена диаграмма последовательности, наглядно представляющая процесс осуществления удаленного доступа к тестируемому оборудованию.

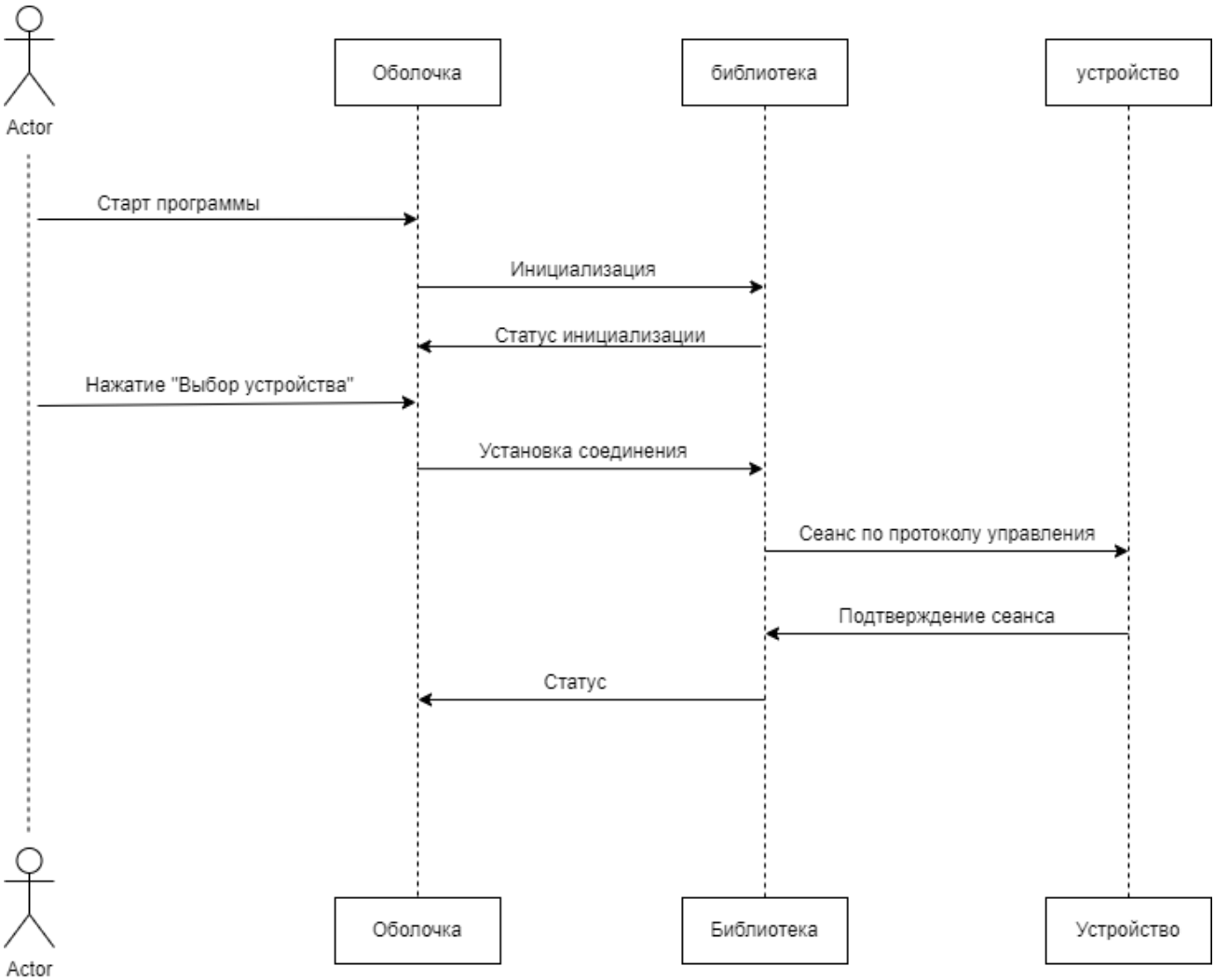

Рисунок 6 – Диаграмма последовательности «Подключение»

После удачного подключения есть два варианта тестирования одинарное и циклическое тестирование, на рисунке 7 представлена диаграмма

последовательности для одиночного варианта тестирования, на рисунке 8 для циклического.

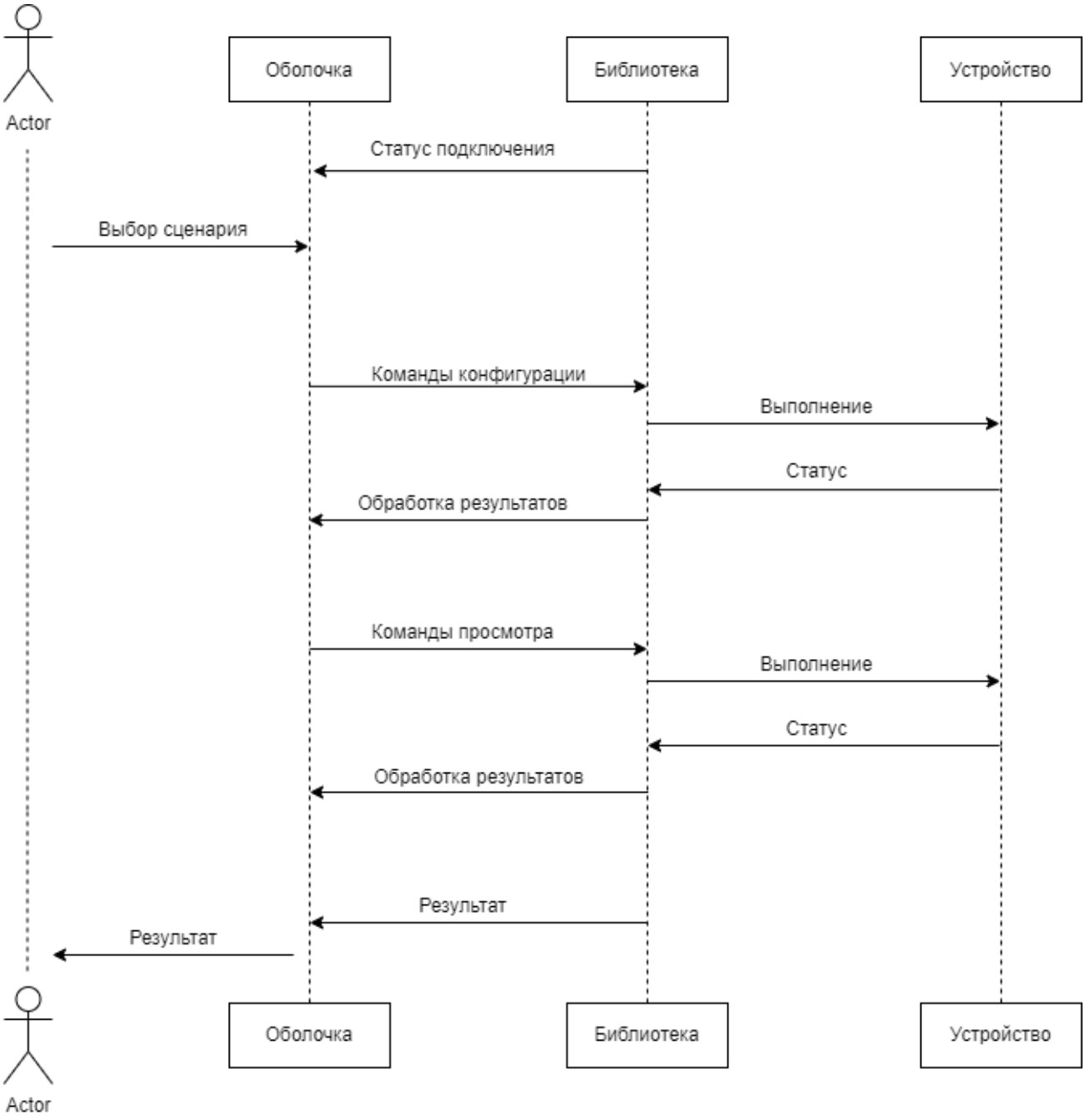

Рисунок 7 – Диаграмма последовательности «Одинарное тестирование»

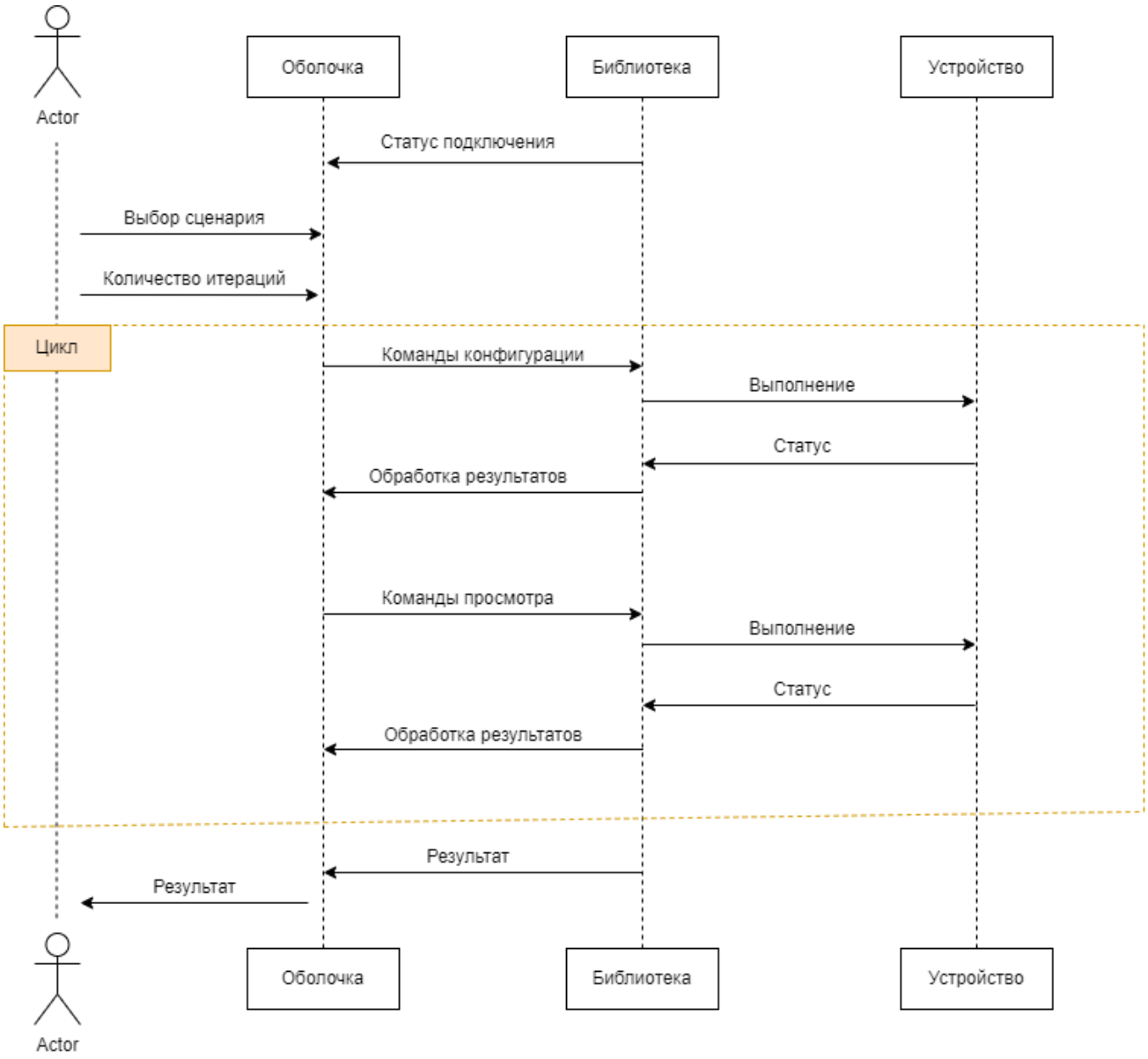

Рисунок 8 – Диаграмма последовательности «Циклическое тестирование»

2.2 Прецеденты

Взаимодействие библиотеки с пользователем происходит не на прямую, а через основную систему тестирования телекоммуникационного оборудования.

В данном разделе приведены наиболее значимые прецеденты, связанные со взаимодействием различных компонентов программы.

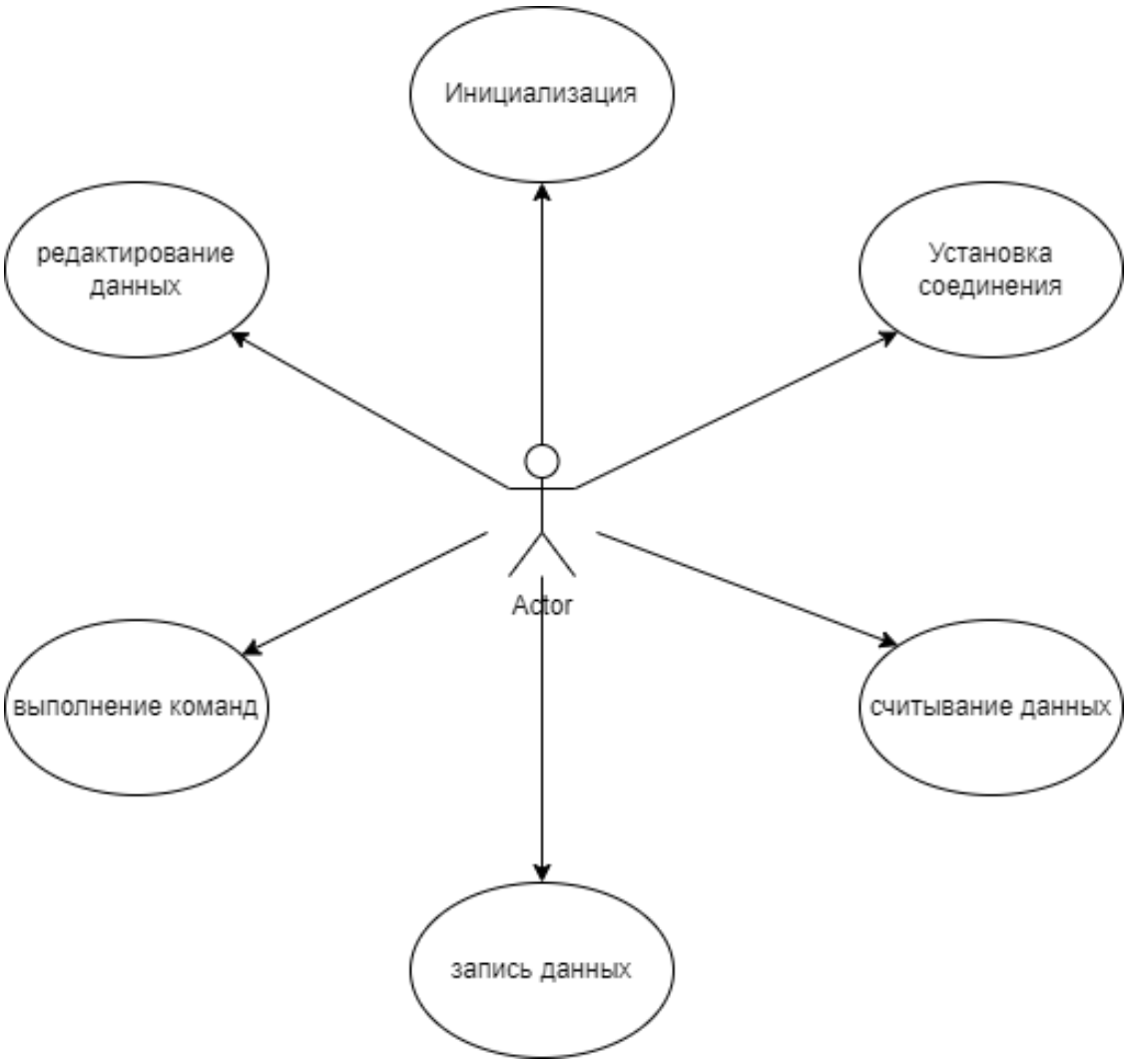

Рисунок 9 – Диаграмма вариантов использования

Прецедент 1. Получение файла конфигурации и данных об оборудовании

Цель сценария: увидеть список телекоммуникационного оборудования и сценарии тестирования.

Предусловия: Пользователь нажал кнопку “Начать тестирование” в системе тестирования телекоммуникационного оборудования.

Основной сценарий: получение файла конфигурации и считывание информации о модемах (IP_address, login, password, port) и команд конфигурации (Connect 192.8.8.8 login yyt pass [pass/ASK], Show

[all/sat/Ethernet] interfaces GET [bytes/packets/errors/status/ip], Save, disconnect и т.д.)

Условия ввода в действие альтернативных сценариев

Условие 1. Ошибка соединения.

Условие 2. Не верный сценарий тестирования.

Прецедент 2. Установка соединения с тестируемым оборудованием

Цель сценария: организовать удаленное подключение к телекоммуникационному оборудованию

Предусловия: все полученные данные корректны

Основной сценарий: осуществляется удаленное подключение посредством SSH протокола

Условия ввода в действие альтернативных сценариев

Условие 1. Сессия занята и в данный момент организовать подключение невозможно.

Условие 2. В процессе подключения возникла ошибка сети.

Прецедент 3. Тестирование телекоммуникационного оборудования

Цель сценария: протестировать подключенное оборудование и получить результат

Основной сценарий: отправка последовательности команд, полученных из файла сценария тестирования

Условия ввода в действие альтернативных сценариев

Условие 1. Не верные команды

Условие 2. В процессе тестирования произошла ошибка

Прецедент 4. Получение и редактирование результатов

Цель сценария: получение и обработка информации по заданным критериям

Предусловия: все полученные данные корректны

Основной сценарий: полученный результат тестирования записывается отдельно в лог файл, после чего посредством редактирования полученной информации по критериям, также записываются в отдельный файл

Условия ввода в действие альтернативных сценариев

Условие 1. В процессе тестирования оборудование не вернуло никаких результатов

Прецедент 5. Переключение режимов

Цель сценария: изменить режим на модеме

Основной сценарий: отправляется команда на изменение режима конфигурации.

Условия ввода в действие альтернативных сценариев

Условие 1. Модем вернул ошибку. Отображается информация с описанием ошибки.

Прецедент 6. возврат полученной информации

Цель сценария: после тестирования и редактирования результатов вернуть их системе тестирования для записи в базу данных

Основной сценарий: полученные данные собираются в файл и отправляются в систему тестирования для дальнейшего сохранения

Условия ввода в действие альтернативных сценариев

Условие 1. В процессе формирования выходных данных произошла ошибка. Отображается информация с описанием ошибки.

2.3 Выводы по главе

В соответствии с технических заданием была предложен архитектура системы, был описан подход и алгоритм работы программы, так же описаны функции, которые должна реализовать библиотека

- получение файла конфигурации и информации об оборудовании;
- выполнение команд на оборудовании и получение результата;
- обработка поученных результатов;
- переключение режимов (конфигурационный и пользовательский);
- возврат результата.

3. Программная реализация

В процессе программной реализации были решены следующие задачи:

- Реализовано удаленное подключение к тестируемому оборудованию с использованием протокола удаленного подключения.

- Реализован алгоритм тестирования;

- Реализовано получение и редактирование полученных данных;

- Реализовано сохранение поученных результатов.

3.1 Инициализация

Перед началом работы библиотеки, необходимо передать в нее параметры запуска такие как, информация об устройстве, а именно IP-адрес, логин, пароль, порт подключения, файл со сценарием тестирования, и при необходимости ключевое слово, по которому будет происходить редактирование полученных данных.

```
python v1.0.py example.txt 192.168.0.113 login pass 22
```

Рисунок 10 – Пример параметров вызова библиотеки

После чего происходит инициализация всех функций, и в случае неудачи возвращается ошибка.

3.2 Подключение

После успешной инициализации, необходимо организовать удаленное подключение к тестируемому оборудованию. За это отвечает функция **start_connection()**.

Входными параметрами данной функции является информация о устройстве, полученная из входных параметров вызова библиотеки.

```
PS C:\Users\babok\Desktop\sshconection> python v1.0.py example.txt 192.168.0.113 some_name 1 22 test
['192.168.0.113', 'some_name', '1', '22']
PS C:\Users\babok\Desktop\sshconection>
```

Рисунок 11 – Информация об устройстве

```
def start_connection(host,user,secret,port):
```

Рисунок 12 – Наименование функции и принимаемые параметры

Далее происходит проверка полученных параметров на корректность, в случае ввода неверных данных возвращается ошибка, в ином случае выполняется подключение.

Удаленное подключение к оборудованию происходит посредством управляющего протокола, в данном случае SSH для его реализации используются следующие методы:

- client = paramiko. SSHClient () для работы с ssh — создается объект SSHClient

- client.set_missing_host_key_policy (paramiko. AutoAddPolicy ()) в данной строке добавляется ключ сервера в список известных хостов — файл

.ssh/known_hosts. Если при соединении с сервером ключа в нем не найдено, то по умолчанию ключ «отбивается» и вызввается SSHException.

- client. connect () – осуществляется соединение с сервером.

- chan = client. invoke_shell () гарантирует стабильность сессии

В результате выполнения функции получаем следующие варианты:

- Функция выполнила подключение как и ожидалось;

- В процессе подключения произошел сбой на стороне клиента;

- В процессе подключения произошел сбой на стороне оборудования.

Функция прекращает работу при успешном выполнении тестирования, либо если в процессе подключения произошла ошибка.

3.3 Считывание информации

Для получения команд конфигурации из файла тестирования была реализована функция **read_from_file()**, принимающая в качестве параметра путь с названием файла.

Рисунок 13 - Наименование функции и принимаемые параметры

Построчно считывается информация из сценария тестирования и в зависимости от параметров передается в функцию **single_test()** или **cycle_test()**, в качестве параметра.

Варианты завершения выполнения функции:

- Функция отработала корректно и отправила полученную информацию;

- В процессе произошла ошибка.

3.4 Тестирование

Библиотека предоставляет два варианта тестирования телекоммуникационного оборудования:

- Одинарное тестирование;
- Циклическое тестирование.

3.4.1 Одинарное тестирование

После успешного соединения, необходимо выполнить сценарий тестирования, для одинарного тестирования реализована функция: **single_test()**

В качестве входных параметров данная функция принимает статус подключения, данные полученные из сценария тестирования.

Рисунок 14 - Наименование функции и принимаемые параметры

Посредством метода **send()** отправляем указанную строку, полученную из сценария тестирования в сессию и возвращает количество отправленных байт или ноль если сессия закрыта и не удалось отправить команду.

На рисунке 15 представлен самый простой сценарий тестирования.

```
show interface GigabitEthernet
ping 192.168.0.111
show temperature
exit
```

Рисунок15 – Пример сценария тестирования

Варианты завершения выполнения функции:

- заканчивает выполнение тогда, когда была отправлена последняя команда и получен ответ от тестируемого оборудования;

- В процессе тестирования произошла ошибка.

3.4.1.1 Не интерактивные команды

Получаемые команды подразделяются на интерактивные и не интерактивные, в случае, когда приходят не интерактивные команды(команды просмотра), библиотека последовательно посылает их в командную сроку устройства с задержкой в 3 секунды, получая ответ.

Пример не интерактивных команд:

- группа команд просмотра конфигурационных, оперативных параметров и статистики show;

- группа команд удаления данных и сброса clear;

- команда копирования файлов, загрузки и сохранения конфигурации, установки программного обеспечения copy;

- команды тестирования ping и traceroute;

- команда перезагрузки reboot;

- команда выхода exit.

3.4.1.2 Интерактивные команды и команды конфигурации

В случае получения интерактивных команд и команды конфигурации, есть два варианта работы с ними.

Первый вариант подразумевает под собой также как и в первом случае устанавливаем таймер, что не подходит, так как заранее не известно сколько времени займет выполнение той или иной команды, поэтому был реализован второй вариант.

Второй вариант реализации заключается в том, что мы ожидаем, когда модем вернет результат выполнения предыдущей команды, посредством того, что постоянно проверяется последняя полученная строка, и в случае если оборудование вернуло приглашение командной строки, тогда отправляется следующая команда.

Примером данных команд являются:

- команда перехода к специализированным режимам управления сетевыми интерфейсами interface;

- команда назначения изделию имени сетевого узла hostname;

- команды редактирования таблицы учётных записей пользователей username и no username;

- команды редактирования таблицы маршрутизации ip route и no ip route;

- команда установки таймера на перезагрузку set watchdog;

- группа команд установки текущих даты, времени, часового пояса clock;

- группа команд установки географических координат земной станции и орбитальной позиции геостационарного космического аппарата position;

- группа команд настройки пользовательских интерфейсов управления ui;

- команды выхода end и exit.

- команда ping.

3.4.2 Циклическое тестирование

Для реализации циклического тестирования была реализована функция **sycle_test()**, принимающая в качестве входных параметров статус подключения и сценарий тестирования.

```
def sycle_test(status,data):
```

Рисунок 16 – Наименование функции и принимаемые параметры

Для прохождения циклического тестирования кроме самих команд, также необходимо получить количество итераций.

Посредством метода **send()** отправляет указанную строку, полученную из сценария тестирования в сессию и возвращает количество отправленных байт или ноль если сессия закрыта и не удалось отправить команду.

Варианты завершения:

- выполнено требуемое количество итераций;

- в процессе выполнения произошел сбой с оборудованием;

- в процессе выполнения произошел разрыв соединения.

3.5 Получение данных

За получение данных отвечает функция **get_data()**, после выполнения сценария тестирования все полученные данные собираются с помощью метода **recv(),** и записываются в новый файл. Пример полученных данных представлен в приложении А.

```
def read_from_file(path):
```

Рисунок 17 - Наименование функции и принимаемые параметры

Варианты завершения выполнения функции:

- В процессе считывания произошла ошибка;
- Была получена последняя строка, функция завершилась корректно.

3.6 Редактирование данных

После получения данных о тестировании вся информация передается в функцию **parse().**

Библиотека имеет возможность фильтровать полученный результат по нескольким критериям, таким как:

- Пользователь может указать фильтрацию по группам;
- Отдельно по ключевому слову.

Для выделения групп обработка происходит посредством использования регулярных выражений.

```
def parse(out):
```

Рисунок 18 - Наименование функции и принимаемые параметры

Пример данных отредактированных по группам представлен на рисунке 19, по ключевому слову на рисунке 20.

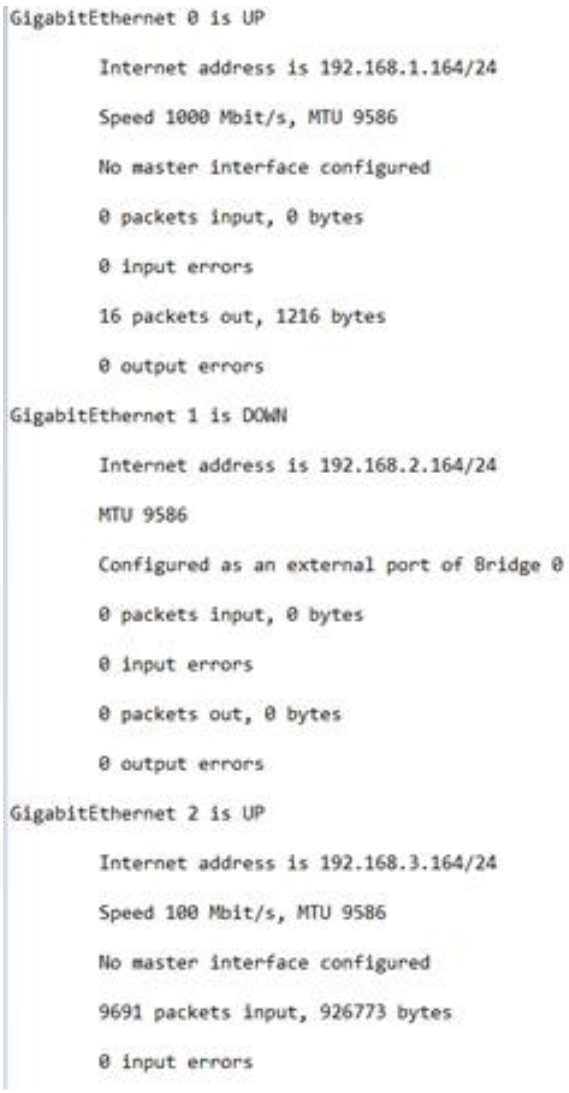

Рисунок 19 – Пример фильтрации данных по группам

```
GigabitEthernet 0 is UP
        Internet address is 192.168.1.164/24
GigabitEthernet 1 is DOWN
        Internet address is 192.168.2.164/24
GigabitEthernet 2 is UP
        Internet address is 192.168.3.164/24
```

Рисунок 20 – Пример фильтрации данных по ключевому слову

3.7 Завершение работы

Варианты завершения работы библиотеки:

- Истечение таймера на выполнение команд;

- Не получен ответ от тестируемого оборудования;

- Аварийное завершение в случае отказа работы оборудования или в процессе подключения;

-После успешного выполнения всего цикла тестирования, собранные файлы возвращаются в систему тестирования, библиотека завершает свою работу.

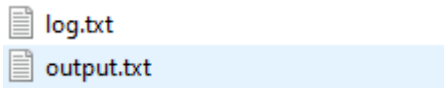

Рисунок 21 – Возвращаемые файлы

ЗАКЛЮЧЕНИЕ

В результате выполнения выпускной квалификационной работы была изучена предметная область и существующие на данный момент аналоги.

После изучения аналогов был сформулирован ряд требований, предъявленных к программному обеспечению. На основе сформулированных требований были определенны технологии разработки. Вся библиотека реализована на языке программирования Python с использованием модуля Paramiko.

В результате была разработана библиотека тестирования телекоммуникационного оборудования поддерживающая различные варианты тестов, поддерживающая подключение посредством протокола удаленного управления SSH.

В дальнейшем возможно расширение функциональной части, в частности добавление поддержки других протоколов управления, что расширит область применения.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. «Кросс-Про» [Электронный ресурс]: – Режим доступа: http://www.sdl.ru/cross-pro
2. Zabbix [Электронный ресурс]: – Режим доступа: https://www.zabbix.com/ru
3. PRTG [Электронный ресурс]: – Режим доступа: https://www.paessler.com/ru/prtg
4. Python [Электронный ресурс]: – Режим доступа: https://www.python.org/
5. Paramiko [Электронный ресурс]: – Режим доступа: https://docs.paramiko.org/en/stable/index.html
6. Клиент – серверная архитектура [Электронный ресурс]: - Режим доступа: https://zametkinapolyah.ru/servera-i-protokoly/o-modeli-vzaimodejstviya-klient-server-prostymi-slovami-arxitektura-klient-server-s-primerami.html
7. SSH [Электронный ресурс]: - Режим доступа: https://freehost.com.ua/faq/wiki/chto-takoe-ssh/
8. Телекоммуникация [Электронный ресурс]: - Режим доступа: http://base.safework.ru/
9. СТУ 7.5–07–2021 [Электронный ресурс]: - Режим доступа: https://about.sfu-kras.ru/docs/8127/pdf/119063
10. C# [Электронный ресурс]: - Режим доступа: http://web.spt42.ru/index.php/
11. C++ [Электронный ресурс]: - Режим доступа: https://docs.microsoft.com/ru-ru/cpp/cpp/cpp-language-reference?view=msvc-170

ПРИЛОЖЕНИЕ А
Не отредактированные данные, полученные после тестирования:

```
[?2004hadmin@yar1040> show interface GigabitEthernet

[?2004l
GigabitEthernet 0 is UP

        Internet address is 192.168.1.164/24

        Speed 1000 Mbit/s, MTU 9586

        No master interface configured

        0 packets input, 0 bytes

        0 input errors

        16 packets out, 1216 bytes

        0 output errors

GigabitEthernet 1 is DOWN

        Internet address is 192.168.2.164/24

        MTU 9586

        Configured as an external port of Bridge 0

        0 packets input, 0 bytes

        0 input errors

        0 packets out, 0 bytes

        0 output errors

GigabitEthernet 2 is UP

        Internet address is 192.168.3.164/24

        Speed 100 Mbit/s, MTU 9586

        No master interface configured

        9691 packets input, 926773 bytes

        0 input errors
```

```
[?2004l
[?2004hadmin@yar1040> ping 192.168.0.111
[?2004l
PING 192.168.0.111  56(84) bytes of data.

64 bytes from li-in-f102.1e100.net (64.233.162.102): icmp_seq=1 ttl=57 time=74.6 ms

64 bytes from li-in-f102.1e100.net (64.233.162.102): icmp_seq=2 ttl=57 time=75.0 ms

64 bytes from li-in-f102.1e100.net (64.233.162.102): icmp_seq=3 ttl=57 time=75.3 ms

64 bytes from li-in-f102.1e100.net (64.233.162.102): icmp_seq=4 ttl=57 time=84.1 ms

64 bytes from li-in-f102.1e100.net (64.233.162.102): icmp_seq=5 ttl=57 time=74.8 ms



--- 192.168.0.111 ping statistics ---

5 packets transmitted, 5 received, 0% packet loss, time 4005ms

rtt min/avg/max/mdev = 74.636/76.778/84.087/3.661 ms
[?2004hadmin@yar1040> show temperature

[?2004l
9 sensors present



Sensor 1: CPU board

  Temperature: 46.000 C

  Min crit.: -40.000 C

  Min alarm: -20.000 C

  Max alarm: 115.000 C

  Max crit.: 125.000 C

  Location: bottom of the CPU board, near the unit's rear side
```

```
Sensor 2: CPU chip

  Temperature: 58.875 C

  Min crit.: -40.000 C

  Min alarm: -20.000 C

  Max alarm: 115.000 C

  Max crit.: 125.000 C

  Location: top of the CPU board, near the unit's rear side


Sensor 3: RAM chips

  Temperature: 43.375 C

  Min crit.: -40.000 C

  Min alarm: -20.000 C

  Max alarm: 85.000 C

  Max crit.: 95.000 C

  Location: top of the CPU board, near the unit's rear side


Sensor 4: BMC

  Temperature: 38.500 C

  Min crit.: -40.000 C

  Min alarm: -20.000 C

  Max alarm: 85.000 C

  Max crit.: 105.000 C

  Location: bottom of the DSP board, center of the unit


Sensor 5: FPGA 1 power supply

  Temperature: 54.375 C
```

```
Sensor 5: FPGA 1 power supply

  Temperature: 54.375 C

  Min crit.: -40.000 C

  Min alarm: -20.000 C

  Max alarm: 80.000 C

  Max crit.: 85.000 C

  Location: top of the DSP board, towards left fan


Sensor 6: FPGA 2 power supply

  Temperature: 57.625 C

  Min crit.: -40.000 C

  Min alarm: -20.000 C

  Max alarm: 80.000 C

  Max crit.: 85.000 C

  Location: bottom of the DSP board, towards right fan


Sensor 7: FPGA 1

  Temperature: 55.123 C

  Min crit.: -40.000 C

  Min alarm: -20.000 C

  Max alarm: 80.000 C

  Max crit.: 85.000 C

  Location: top of the DSP board, towards left fan
```

```
Sensor 8: FPGA 2

  Temperature: 61.644 C

  Min crit.: -40.000 C

  Min alarm: -20.000 C

  Max alarm: 80.000 C

  Max crit.: 85.000 C

  Location: top of the DSP board, towards right fan


Sensor 9: RF transceiver

  Temperature: 41.000 C

  Min crit.: -40.000 C

  Min alarm: -20.000 C

  Max alarm: 80.000 C

  Max crit.: 85.000 C

  Location: top of the RF transceiver board, center of the unit

[?2004hadmin@yar1040>

[?2004l
[?2004hadmin@yar1040> exit

[?2004l
```

Федеральное государственное автономное
образовательное учреждение
Высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий

институт

Вычислительная техника

кафедра

УТВЕРЖДАЮ
Заведующий кафедрой
_____ О.В. Непомнящий
подпись инициалы, фамилия
«20» 06 2022 г.

БАКАЛАВАРСКАЯ РАБОТА

09.03.01 Информатика и вычислительная техника

код и наименование специальности

Библиотека тестирования телекоммуникационного оборудования

тема

Руководитель	_____	Старший преподаватель	И.Н. Рыженко
	подпись, дата	должность, ученая степень	инициалы, фамилия
Выпускник	24.06.22		А.Н. Цуба
	подпись, дата		инициалы, фамилия
Нормоконтролер	_____	Старший преподаватель	И.Н. Рыженко
	подпись, дата	должность, ученая степень	инициалы, фамилия

Красноярск 2022