



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования

**"МИРЭА - Российский технологический университет"**

**РТУ МИРЭА**

---

Институт информационных технологий (ИТ)  
Кафедра математического обеспечения и стандартизации информационных  
технологий (МОСИТ)

**ОТЧЕТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ №1**  
**по дисциплине**  
**«Тестирование и верификация программного обеспечения»**

Тема: Тестирование программного продукта методом «Черного ящика»

Выполнили студенты группы ИКБО-50-23

Аникеев М.И.  
Корольков Г.Д.  
Лысачева М.М.  
Талапов Д.М.  
Ильичев Г.П.

Принял преподаватель

Москва 2025

## ОГЛАВЛЕНИЕ

<b>1 ТЕХНИЧЕСКОЕ ЗАДАНИЕ.....</b>	<b>3</b>
ВВЕДЕНИЕ.....	3
ОСНОВАНИЯ ДЛЯ РАЗРАБОТКИ.....	3
НАЗНАЧЕНИЕ РАЗРАБОТКИ.....	3
ТРЕБОВАНИЯ К ПРОГРАММЕ.....	4
ТРЕБОВАНИЯ К ИНТЕРФЕЙСУ .....	6
КРИТЕРИИ ПРИЕМКИ.....	6
ТРЕБОВАНИЯ К ДОКУМЕНТАЦИИ.....	6
ПОРЯДОК КОНТРОЛЯ И ПРИЕМКИ.....	6
ЭТАПЫ И СРОКИ РАЗРАБОТКИ.....	7
<b>2 ДОКУМЕНТАЦИЯ .....</b>	<b>8</b>
РАБОЧАЯ ДОКУМЕНТАЦИЯ.....	8
РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ.....	8
<b>3 ОПИСАНИЕ ОШИБОК.....</b>	<b>15</b>
<b>4 ТЕХНИЧЕСКОЕ ЗАДАНИЕ И ДОКУМЕНТАЦИЯ ПП ДРУГОЙ КОМАНДЫ.....</b>	<b>16</b>
<b>5 РЕЗУЛЬТАТЫ ТЕСТИРОВАНИЯ ПП ДРУГОЙ КОМАНДЫ.....</b>	<b>29</b>
<b>6 АНАЛИЗ ДОКУМЕНТАЦИИ ДРУГОЙ КОМАНДЫ .....</b>	<b>34</b>
<b>7 ЗАКЛЮЧЕНИЕ .....</b>	<b>35</b>

# **1 ТЕХНИЧЕСКОЕ ЗАДАНИЕ**

## **ВВЕДЕНИЕ**

Программа предназначена для оптимизации взаимодействия наставников и стажеров: постановка задач, объединение стажеров в команды, просмотр отчетов по задачам, уведомление о готовности задач, поиск по скиллам, просмотр резюме и другой информации. Данная программа может использоваться в подразделениях для стажеров, а также в других подразделениях для более удобного и быстрого взаимодействия между руководителями и подчиненными.

## **ОСНОВАНИЯ ДЛЯ РАЗРАБОТКИ**

Программа была разработана как рабочий проект, задача была поставлена руководителем. Целью было упрощение процесса постановки задач, как для одного подчиненного, так и для нескольких сразу.

## **НАЗНАЧЕНИЕ РАЗРАБОТКИ**

Система предназначена для оптимизации временных издержек при постановке задач, ответах на вопросы по задачам, для повышения скорости адаптации стажеров, сократив организационные затраты на рутинные операции и улучшив коммуникацию между всеми участниками процесса.

- Автоматизация рутинных процессов: сбор данных, ответы на частые вопросы, напоминания;
- Стандартизация процесса onboarding-a: каждый стажер получает одинаково полную и структурированную информацию;
- Централизация информации и документооборота: вся информация и все документы в одном месте;
- Ускорение коммуникации
- Обеспечения легкого доступа к задачам

## **ТРЕБОВАНИЯ К ПРОГРАММЕ**

### **Функциональные требования:**

#### **1. Роли в приложении:**

- Администратор – пользователь на руководящей должности, который имеет возможность просмотра и постановки задач, просмотра информации о стажерах и объединения их в группы;
- Стажер – пользователь-стажер, видит только себя, свои задачи и группы, в которые он добавлен;
- Суперадминистратор – контролирует регистрацию администраторов: одобряет или не одобряет регистрацию администратора, удаляет пользователей, следит за базой данных.

#### **2. Личные профили:**

- Каждый пользователь проходит регистрацию с указанием ФИО. Стажеры
- Username стажера должен быть передан администратору после регистрации.

#### **3. Отправка резюме**

- Функция отправки резюме при регистрации или обновления резюме при редактировании профиля;
- Пользователь должен иметь возможность загрузить резюме в формате PDF или DOCX.

#### **4. Управление задачами**

- Создание задач и описаний к ним;
- У каждой задачи есть установленный дедлайн;
- Администратор может создавать задачи для одного, или нескольких стажеров, или группы стажеров;
- Администратор может редактировать задачи;
- Стажеры получают уведомление о новой задаче и ее дедлайне.

#### **5. Управление группами**

- Администратор может создавать группы из нескольких стажеров;
- Администратор может редактировать состав и название группы;
- Стажеры получают уведомление о добавлении в группу.

#### 6. Отчет о результатах выполнения задачи

- Администратор может указывать конкретный формат отчета о выполнении задачи;
- Стажеры имеют возможность прикреплять файлы (документы, скриншоты и т.д.) и отправлять их через бот;
- Отчет доступен к просмотру администратору и может быть возвращен на доработку.

#### **Требования к надежности:**

- Бот должен отвечать на любые сообщения пользователя в течение 5 секунд;
- Бот быть отказоустойчивым: не должно быть ошибок, приводящих к полной остановке работы бота;
- Резервное копирование базы данных происходит каждый день и хранится в течение двух суток для быстрого восстановления работы бота в случае сбоев.

#### **Требования к эксплуатации:**

- Доступ к боту осуществляется через телеграм приложение;
- Размещение на сервере с доступом в интернет;
- Использование локальных баз данных для хранения информации о пользователях и задачах;
- Соблюдение правил рабочей документации.

#### **Требования к совместимости:**

- Совместимость с Telegram API и актуальными версиями мессенджера;
- Внешние сервисы: сервисы для хранения (базы данных);

## **ТРЕБОВАНИЯ К ИНТЕРФЕЙСУ**

- Доступ к функционалу бота осуществляется через команды (/command), через обычные кнопки (KeyboardButton), представленные в виде клавиатуры внизу экрана, через кнопки, прикрепленные к конкретному сообщению (InlineKeyboardButton);
- Для удобства и упрощения взаимодействия из большинства состояний можно вернуться назад (например, отменить создание задачи);
- При запуске и вызове команды /help выводится руководство пользователя: описание функционала бота, это сделает интерфейс более понятным пользователю.

## **КРИТЕРИИ ПРИЕМКИ**

- При оценке интерфейса у заказчика не возникает вопросов по использованию функционала бота за счет поясняющей команды /help и логического расположения кнопок;
- Основной рабочий функционал, успешно прошедший 95% всех тест-кейсов, реализован в установленные сроки.

## **ТРЕБОВАНИЯ К ДОКУМЕНТАЦИИ**

- Руководства пользователя, описывающего использование функционала бота;
- Рабочая документация.

## **ПОРЯДОК КОНТРОЛЯ И ПРИЕМКИ**

- Проверка наличия документации;
- Оценка интерфейса и тестирование методом «черного ящика», подразумевающего проверку функциональной работоспособности программы без знания способов реализации, выявление ошибок;
- Анализ внутренней структуры и тестирование методом «белого ящика», подразумевающего проверку внутренней реализации программы и

логики выполнения, обнаружение нарушений логики, приводящих к ошибкам в работе;

### **ЭТАПЫ И СРОКИ РАЗРАБОТКИ**

- Дизайн и проектирование архитектуры бота: до 01.03.2025;
- Реализация функционала: до 20.04.2025;
- Тестирование всех функций и исправление ошибок: до 12.05.2025.

## 2 ДОКУМЕНТАЦИЯ

### РАБОЧАЯ ДОКУМЕНТАЦИЯ

```
1  services:
2    app:
3      build: .
4      restart: always
5      depends_on:
6        - db
7      ports:
8        - 8000:8000
9      environment:
10       - DB_NAME=db/db/thw_db.db
11       - TOKEN=7602332436:AAEZ0Cw5QB4zHREMtteqIzBfUWM3kd8sMIM
12       - ADMIN_FATHER_PSW=1234567890
13       - ADMIN_PSW=1234567890
```

*Рисунок 1 – compose.yml*

Указание паролей для администратора и суперадминистратора в файле compose.yml для запуска в docker контейнере.

Запуск через командную строку: python main.py

### РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ

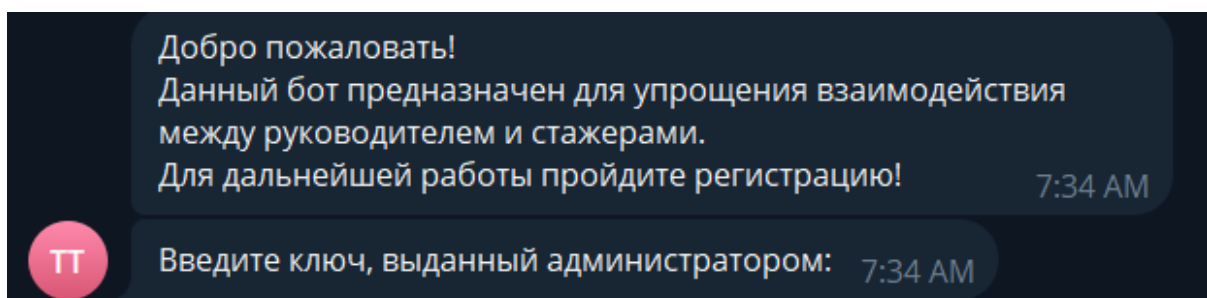
Запуск бота (одинаково для всех пользователей вне зависимости от уровня): команда /start

Указание ключа, выданного администратором (для каждого администратора ключ уникален, по этому ключу пользователь привязывается к своему руководителю, руководителю приходит уведомление, администратор вводит ключ, выданный суперадминистратором, подтверждая свою личность)

Прохождение регистрации:

Администратор указывает ФИО

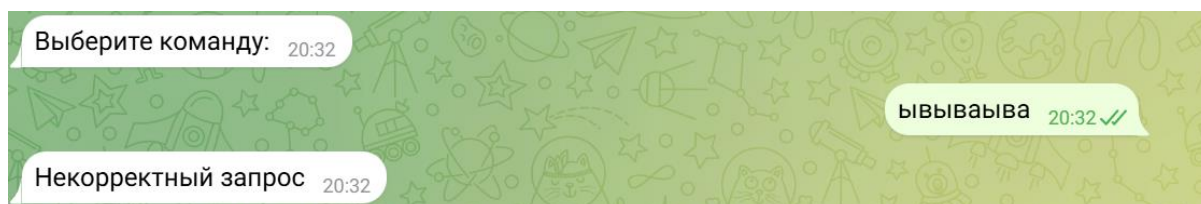
Стажер указывает ФИО, скиллы, прикрепляет файл с резюме.



*Рисунок 2 – Стартовое сообщение*



Большинство функций имеют обработку ошибок для стабильной работы бота с отправкой уведомления о некорректном вводе или выборе команды.

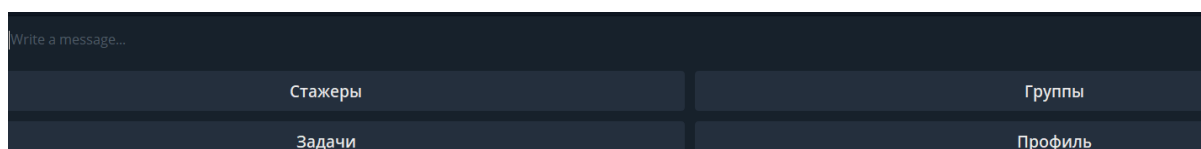


*Рисунок 3 – Обработка ошибок*

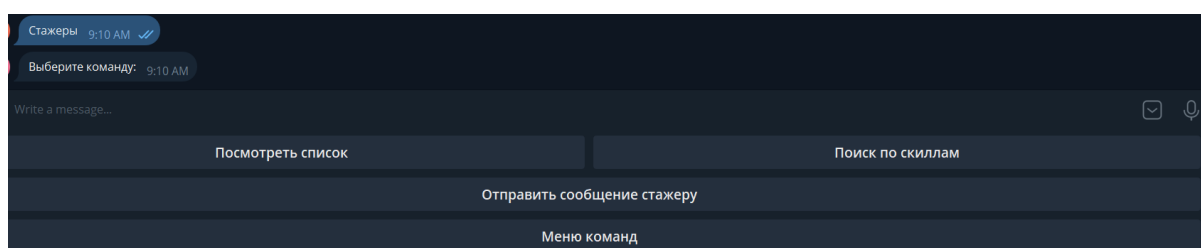
Функционал администратора:

Администратор может создавать, редактировать, удалять, просматривать задачи и группы из стажеров, просматривать список стажеров, информацию о них, осуществлять поиск по скиллам, удалять, просматривать резюме. Администратор может отправлять сообщения стажерам, просматривать отчеты о выполнении задач и принимать или возвращать на доработку задачи. При регистрации нового пользователя администратор получает уведомление. Администратор также может редактировать свой профиль, там же можно найти уникальный ключ для добавления новых стажеров.

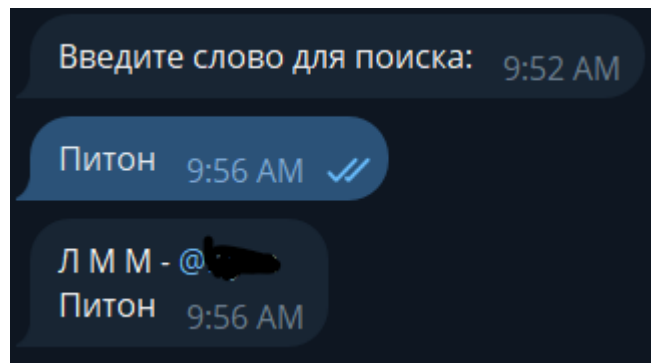
Ниже приведены примеры реализации некоторых функций.



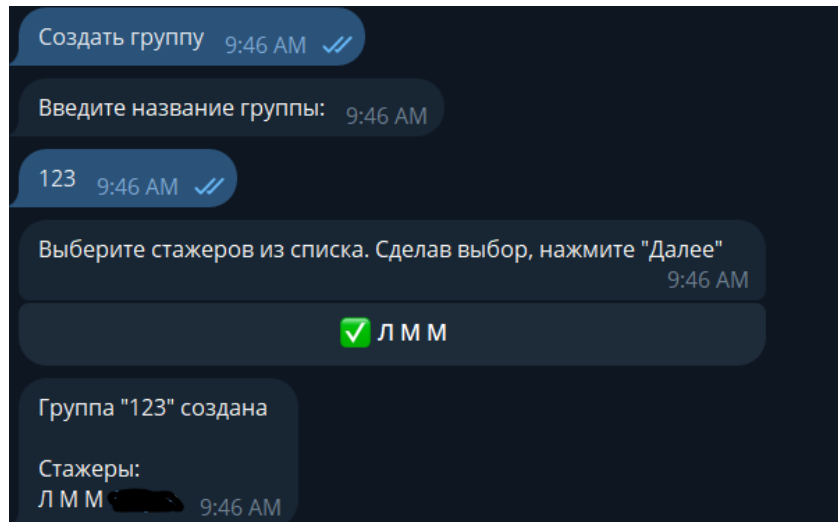
*Рисунок 4 – Выбор команды*



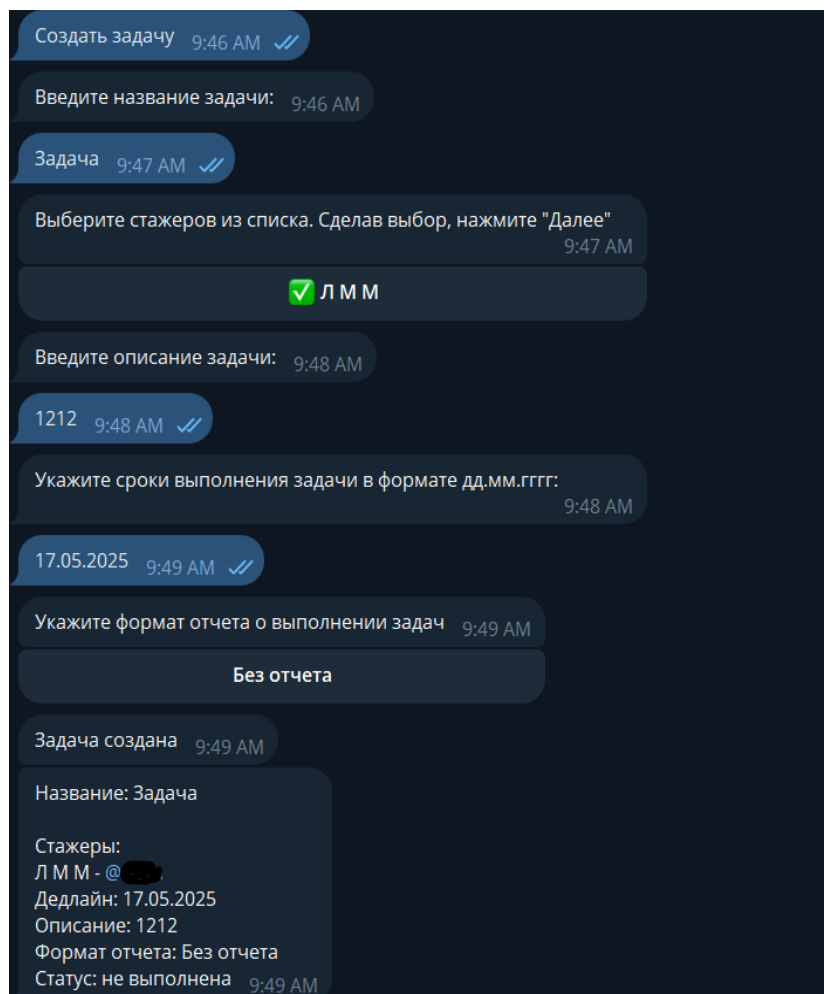
*Рисунок 5 – Выбор команды*



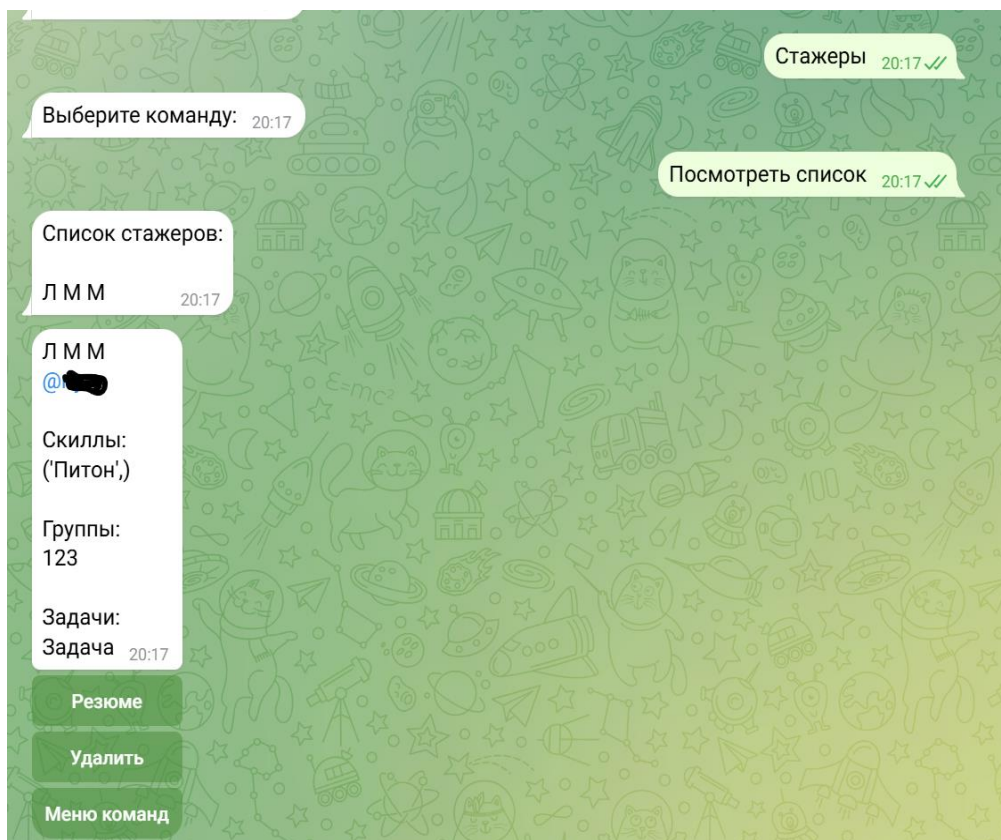
*Рисунок 6 – Поиск стажеров по скиллам*



*Рисунок 7 – Создание группы*



*Рисунок 8 – Создание задачи*



*Рисунок 9 – Просмотр списка стажеров и информации о стажере*

#### Функционал суперадминистратора:

Суперадминистратор может просматривать список всех пользователей и удалять их, просматривать информацию о пользователях и о ключах администраторов, выдавать ключи для регистрации новых администраторов, обновлять одновременно ключи всех администраторов в целях безопасности. Суперадминистратор также может редактировать свой профиль.

#### Функционал стажера:

Стажер имеет возможность просматривать список своих задач, групп, в которые он добавлен, и более подробную информацию о группах и задачах. После выполнения задач стажер должен отчитываться о выполнении задачи в указанном формате, стажер получает уведомление о принятии или отклонении результатов выполнения задачи. Стажер также может редактировать свой профиль.

Ниже приведены примеры реализации некоторых функций.

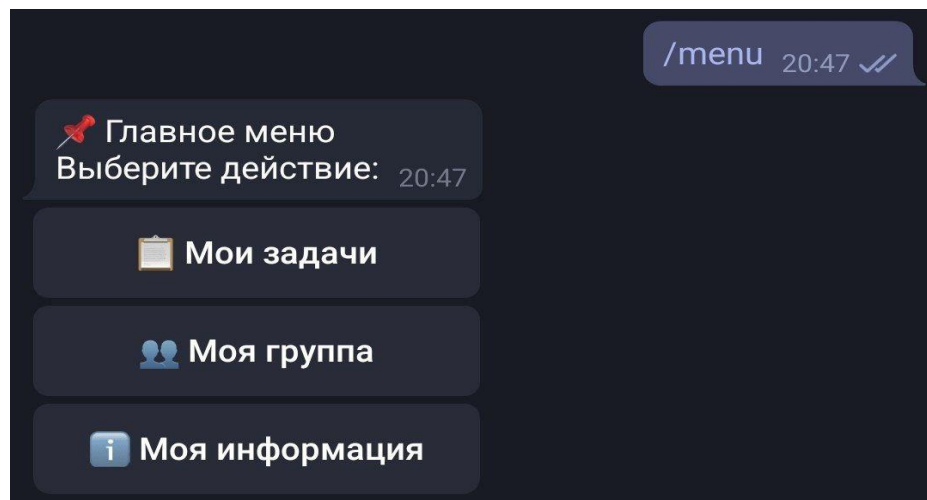


Рисунок 10 – Меню

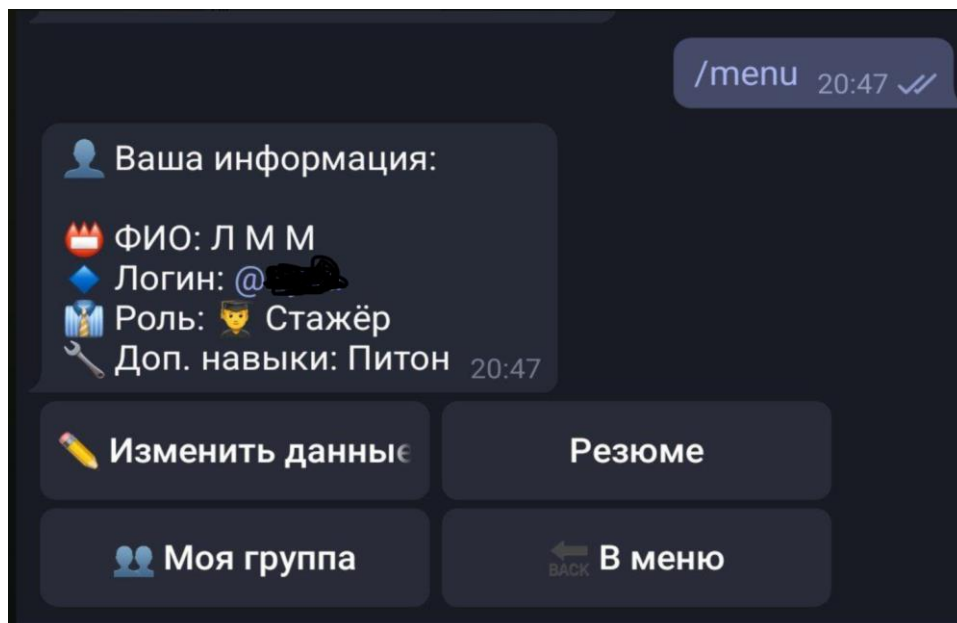


Рисунок 11 – Просмотр профиля

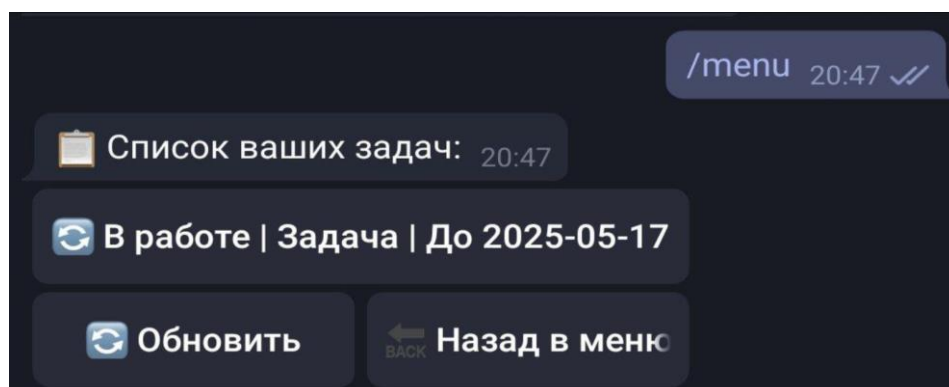
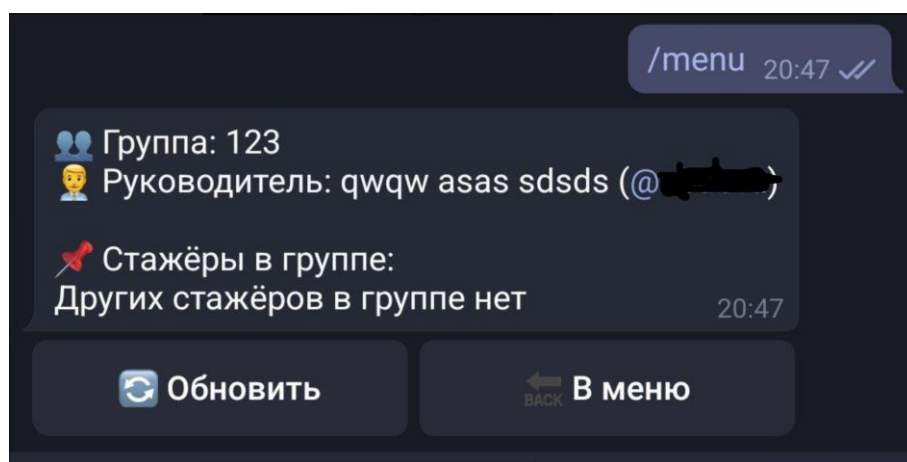


Рисунок 12 – Просмотр списка задач



*Рисунок 13 – Просмотр информации о группе*

### **3 ОПИСАНИЕ ОШИБОК**

1. Отсутствие проверки полей ввода (бот остается работоспособным, но выдает сервер выдает ошибку);
2. Возможность проведения SQL-инъекции вследствие отсутствия проверки ввода;
3. Отсутствие ограничения количества запросов, посылаемых на сервер, что может замедлить или прекратить работу Docker-контейнера, развернутого на сервере Linux.

## **4 ТЕХНИЧЕСКОЕ ЗАДАНИЕ И ДОКУМЕНТАЦИЯ ПП ДРУГОЙ КОМАНДЫ**

### **Введение**

Программа "Конвертер величин" представляет собой desktop-приложение для быстрого и удобного преобразования различных физических величин между единицами измерения. Приложение предназначено для использования в образовательных целях, в быту, а также в профессиональной деятельности, где требуется оперативный пересчет значений между различными системами измерений.

Области применения:

- Учебный процесс в школах и вузах
- Инженерные расчеты
- Научно-исследовательская деятельность
- Повседневное использование для бытовых расчетов



## **Основания для разработки**

Разработка программы осуществляется в рамках учебного проекта для демонстрации практических навыков программирования на Python с использованием библиотеки tkinter для создания графического интерфейса.

Исходные документы:

- Задание на практическую работу по программированию
- Рекомендации по разработке пользовательских интерфейсов

## **Назначение разработки**

Программа предназначена для автоматизации процесса конвертации физических величин, что позволит:

- Сократить время расчета по сравнению с ручными вычислениями
- Исключить ошибки при пересчете значений между системами измерений
- Обеспечить наглядное представление результатов преобразования
- Предоставить удобный интерфейс для работы с различными категориями величин

## **Требования к программе**

### **Функциональные требования**

- Конвертация величин в категориях: длина, масса, температура, время, объем, площадь
- Возможность выбора конкретных единиц измерения внутри каждой категории
- Функция обмена выбранных единиц измерения
- Отображение результатов преобразования с точностью до 8 знаков после запятой
- Обработка ошибок ввода и некорректных данных
- Валидация вводимых значений (проверка на отрицательные значения для определенных категорий)
- Очистка всех полей ввода по запросу пользователя

### **Требования к надежности**

- Обработка исключительных ситуаций без аварийного завершения программы
- Корректное восстановление после ошибок ввода
- Защита от некорректных операций (деление на ноль, нечисловые значения)
- Стабильная работа при непрерывном использовании

### **Условия эксплуатации**

- Операционная система: Windows 7/8/10/11, Linux, MacOS
- Минимальные требования: 512 МБ ОЗУ, 50 МБ свободного места на диске
- Обязательное наличие интерпретатора Python версии 3.6 и выше
- Наличие библиотеки tkinter в составе Python

### **Требования к совместимости**

- Совместимость с операционными системами семейства Windows, Linux, MacOS
- Поддержка кодировки UTF-8 для корректного отображения символов
- Независимость от дополнительного программного обеспечения (кроме Python)

## **Требования к интерфейсу**

### **Основные элементы интерфейса**

- Выпадающий список для выбора категории величин
- Поле ввода исходного значения с возможностью использования точки или запятой как разделителя
- Два выпадающих списка для выбора единиц измерения (исходная и целевая)
- Кнопка обмена единицами измерения ( $\leftrightarrow$ )
- Кнопка "Конвертировать" зеленого цвета с белым текстом
- Поле вывода результата с зеленым текстом на светлом фоне
- Область дополнительной информации о результате преобразования
- Кнопка "Очистить все" красного цвета с белым текстом

### **Особенности интерфейса**

- Все элементы управления должны быть подписаны на русском языке
- Размер окна фиксированный (500x650 пикселей)
- Шрифты: Arial различных размеров (9-16) и начертаний
- Цветовая схема: сочетание зеленых, серых и белых тонов
- Кнопка конвертации должна быть заметным акцентным элементом
- Поле результата должно визуально отличаться от поля ввода

### **Критерии приемки**

- Успешное выполнение 100% тестов
- Отсутствие аварийных завершений программы при любых сценариях использования
- Соответствие времени отклика интерфейса требованиям (не более 0.5 сек)
- Полное соответствие функциональности заявленным требованиям

## **Требования к документации**

### **Обязательная документация**

- Запрос пользователя с описанием требуемой системы
- Техническое описание программы
- Техническое задание
- Исходный код с соответствующими комментариями

## **Порядок контроля и приемки**

### **Методы тестирования**

- Функциональное тестирование всех элементов интерфейса
- Тестирование корректности преобразования величин
- Тестирование обработки ошибок и исключительных ситуаций
- Тестирование пользовательского интерфейса на удобство использования

### **Приемочные испытания**

- Проверка на соответствие техническому заданию
- Тестирование на различных операционных системах
- Проверка работы при различных разрешениях экрана
- Оценка удобства интерфейса целевой аудиторией

### **Этапы и сроки разработки**

1. Проектирование архитектуры приложения – 1 день
2. Разработка графического интерфейса – 2 дня
3. Реализация логики преобразования величин – 3 дня
4. Реализация обработки ошибок и исключений – 1 день
5. Тестирование и отладка программы – 2 дня
6. Написание документации – 1 день

Общий срок разработки: 10 рабочих дней



## **Дополнительная документация на программный продукт "Конвертер величин"**

### **1. Руководство пользователя**

#### **Обзор приложения**

"Конвертер величин" — это графическое приложение для конвертации различных физических единиц измерения. Оно поддерживает категории: Длина, Масса, Температура, Время, Объем и Площадь. Приложение позволяет вводить значение, выбирать исходную и целевую единицы, выполнять конвертацию и отображать результат.

#### **Установка и запуск**

- **Требования:** Python 3.x с установленной библиотекой Tkinter (обычно входит в стандартную поставку Python).
- **Запуск:** Сохраните код в файл `main.py` и выполните команду `python main.py` в терминале. Приложение откроется в отдельном окне.

#### **Интерфейс пользователя**

- **Категория:** Выберите категорию из выпадающего списка (например, "Длина").
- **Исходное значение:** Введите числовое значение в поле ввода. Поддерживаются целые и дробные числа (используйте точку или запятую как разделитель).
- **Из:** Выберите исходную единицу измерения.
- **В:** Выберите целевую единицу измерения.
- **Кнопка "↔":** Обменяет исходную и целевую единицы.
- **Кнопка "Конвертировать":** Выполнит расчет и отобразит результат.
- **Раздел результата:** Показывает конвертированное значение и дополнительную информацию (например, "10 метры = 1000 сантиметры").
- **Кнопка "Очистить все":** Сбросит все поля.

#### **Примеры использования**

1. **Конвертация длины:** Выберите "Длина", введите "1" в поле значения, "километры" в "Из", "метры" в "В". Нажмите "Конвертировать". Результат:

"1000".

2. **Конвертация температуры:** Выберите "Температура", введите "0", "°C" в "Из", "°F" в "В". Результат: "32".
3. **Обмен единиц:** Нажмите "↔" для быстрого переключения между "Из" и "В".
4. **Обработка ошибок:** Если значение не число, приложение покажет сообщение "Введите корректное число". Для отрицательных значений в "Масса" или "Длина" — "Значение не может быть отрицательным".

### **Советы**

- Используйте клавишу Enter для конвертации после ввода значения.
- Результат форматируется для удобства: с разным количеством знаков после запятой в зависимости от величины.
- Для температуры учитывается абсолютный ноль (в Кельвинах результат не может быть отрицательным).

## **2. Описание архитектуры системы**

### **Общая структура**

Приложение построено на основе объектно-ориентированного подхода. Основной класс UnitConverter инкапсулирует всю логику: инициализацию данных, настройку UI и обработку конвертации. Главная функция main() создает корневое окно Tkinter и экземпляр класса.

- **Модули и библиотеки:**
  - tkinter и ttk для графического интерфейса.
  - messagebox для вывода ошибок.
- **Данные:**
  - Словарь units хранит списки единиц по категориям.
  - Словарь conversion\_factors содержит коэффициенты конвертации (относительно базовой единицы, первой в списке).
- **Методы:**
  - \_\_init\_\_: Инициализирует окно, данные и UI.
  - setup\_ui: Настраивает элементы интерфейса (фреймы, метки,

комбобоксы, кнопки).

- `update_units`: Обновляет списки единиц при смене категории.
- `swap_units`: Обменивает единицы "Из" и "В".
- `clear_fields`: Очищает поля ввода и результата.
- `convert_temperature`: Специальная функция для температуры (конвертация через Цельсий).
- `convert`: Основная логика конвертации, включая проверки и форматирование.

### Архитектурные принципы

- **Модульность**: UI отделен от логики конвертации.
- **Событийно-ориентированный подход**: Используются привязки событий (e.g., `<<ComboboxSelected>>`, `<Return>`).
- **Обработка исключений**: Ловит `ValueError` (некорректное число), `ZeroDivisionError` и другие.
- **Базовая единица**: Для большинства категорий конвертация идет через базовую единицу (e.g., метры для длины).
- **Особенности**: Температура обрабатывается отдельно из-за нелинейных формул. Для объема реализована специальная логика (см. ошибки ниже).

### 3. Схемы и диаграммы

#### Диаграмма классов (UML-в текстовом формате)

UnitConverter
Атрибуты: - root: Tk - units: dict {категория: список единиц} - conversion_factors: dict {категория: коэффициенты} - category_var: StringVar - from_var: StringVar - to_var: StringVar - input_var: StringVar - result_var: StringVar - result_info_var: StringVar
Методы: + <code>__init__(root)</code> : Инициализация окна и UI + <code>setup_ui()</code> : Создание интерфейса + <code>update_units(event)</code> : Обновление единиц + <code>swap_units()</code> : Обмен единиц "Из" и "В" + <code>clear_fields()</code> : Очистка полей ввода/вывода + <code>convert_temperature(value, from, to)</code> : Конвертация температуры

+ convert(): Выполнение конвертации

## Схема интерфейса (ASCP-арт)

Конвертер величин

Категория:  
Длина

Исходное значение:

Из: метры В: километры

Конвертировать

Результат конвертации

Результат:

Очистить все

Рисунок 1 - Интерфейс

### Диаграмма потока данных для конвертации

1. Пользователь выбирает категорию → `update_units` обновляет комбобоксы.
2. Ввод значения + единицы → Нажатие "Конвертировать" → `convert`:
  - Проверки (значение, категория, единицы).
  - Если Температура: `convert_temperature`.
  - Иначе: `value * factor_from / factor_to`.
  - Форматирование → Отображение в `result_var` и `result_info_var`.

## 5 РЕЗУЛЬТАТЫ ТЕСТИРОВАНИЯ ПП ДРУГОЙ КОМАНДЫ

Конвертер величин

Категория:  
Время

Исходное значение:  
1

Из: недели В: секунды

Конвертировать

Результат конвертации

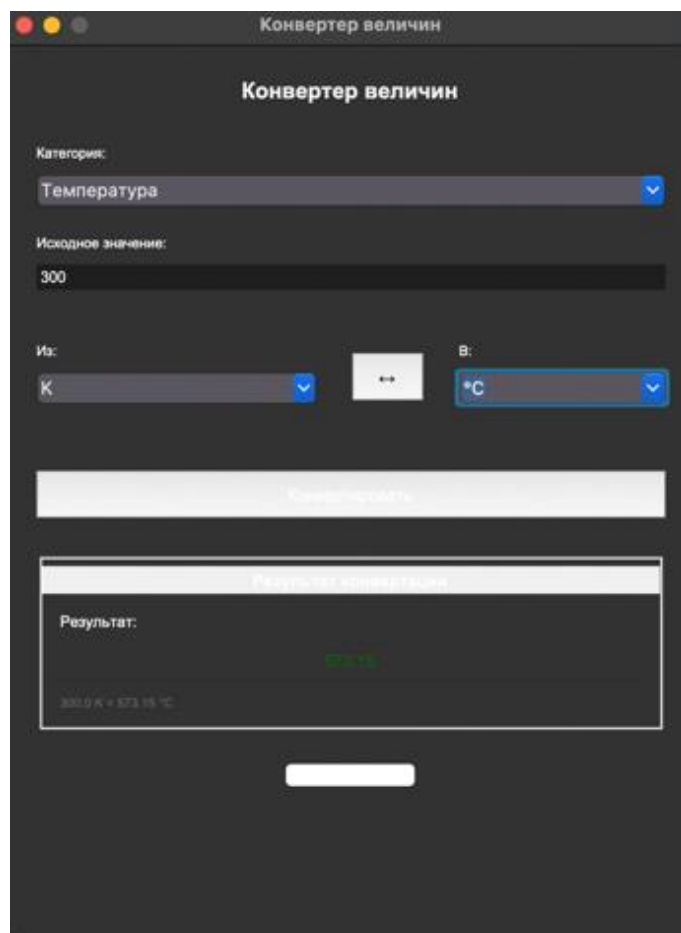
Результат:  
604800

1.0 недели = 604800 секунды

Рисунок 14 – Ошибка 1

1. Идентификатор: НР-001;
2. Название: Проверка корректности расчета по заданным параметрам;
3. Описание: Убедиться, что пользователь увидит корректный результат конвертации при вводе корректных значений;
4. Предварительные условия:
  - а) пользователь вводит корректные значения, соответствующие условиям для конвертации (неотрицательные).
5. Шаги выполнения:
  - а) в поле «Категории» выбрана интересующая категория для перевода;
  - б) в поле «Исходное значение» введено корректное число;
  - в) в поле «Из» выбрана единица измерения;
  - г) в поле «В» выбрана единица измерения;
  - д) нажать кнопку «Конвертировать».

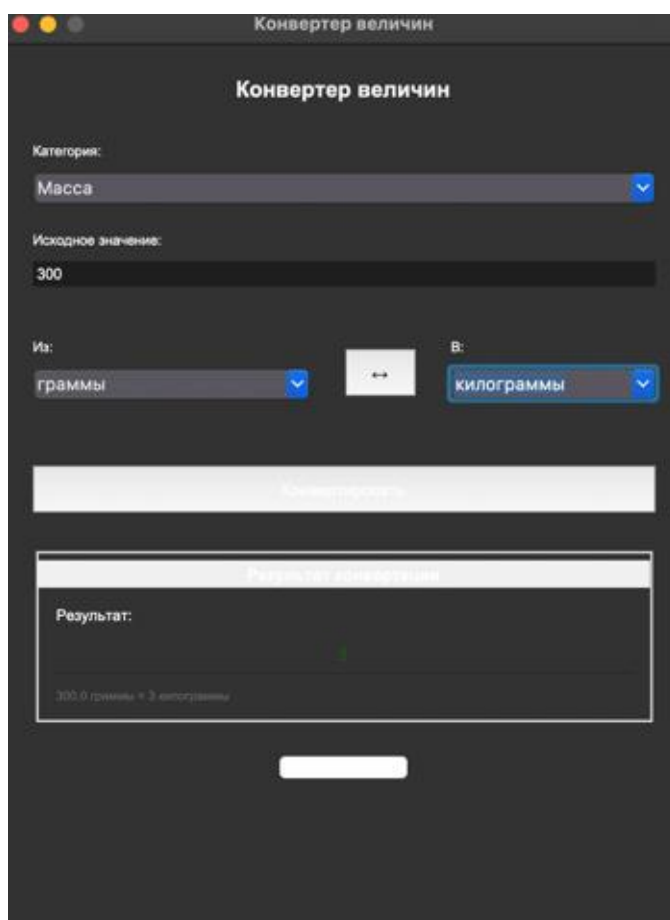
6. Ожидаемый результат: В поле «результат конвертации» выводиться корректный, правильный математически, перевод;
7. Фактический результат: В поле «результат конвертации» выводиться результат с ошибкой в расчетах;
8. Статус: Failed.



*Рисунок 15 – Ошибка 2*

1. Идентификатор: НР-002;
2. Название: Проверка корректности расчета по заданным параметрам;
3. Описание: Убедиться, что пользователь увидит корректный результат конвертации при вводе корректных значений;
4. Предварительные условия:
  - а) пользователь вводит корректные значения, соответствующие условиям для конвертации.
5. Шаги выполнения:
  - а) в поле «Категории» выбрана интересующая категория для перевода;

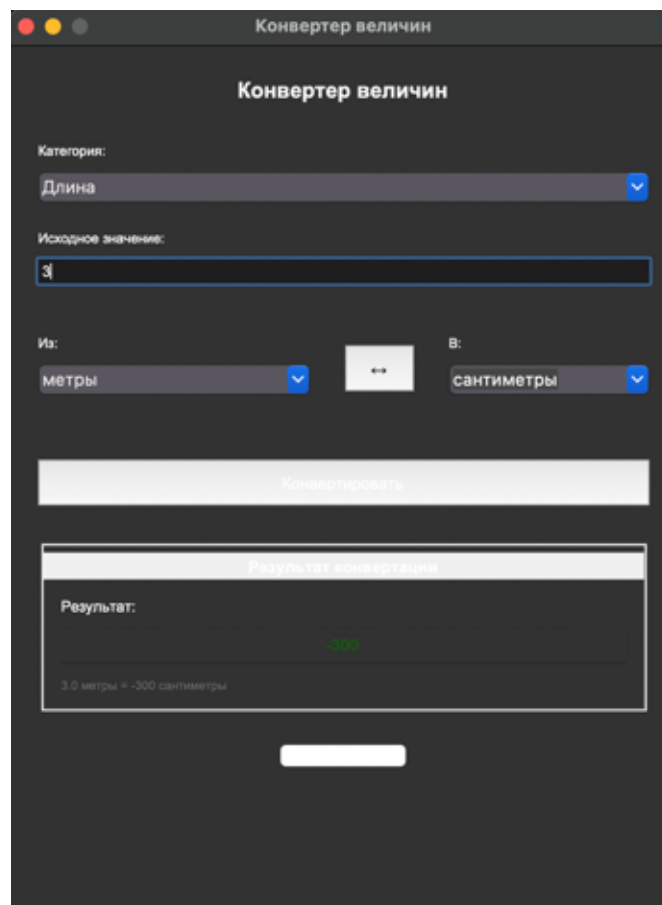
- б) в поле «Исходное значение» введено корректное число;
  - в) в поле «Из» выбрана единица измерения;
  - г) в поле «В» выбрана единица измерения;
  - д) нажать кнопку «Конвертировать».
6. Ожидаемый результат: В поле «результат конвертации» выводиться корректный, правильный математически, перевод;
7. Фактический результат: В поле «результат конвертации» выводиться результат с ошибкой в расчетах;
8. Статус. Failed.



*Рисунок 16 – Ошибка 3*

1. Идентификатор: НР-003;
2. Название: Проверка корректности расчета по заданным параметрам;
3. Описание: Убедиться, что пользователь увидит корректный результат конвертации при вводе корректных значений;
4. Предварительные условия:

- а) пользователь вводит корректные значения, соответствующие условиям для конвертации (неотрицательные).
5. Шаги выполнения:
- а) в поле «Категории» выбрана интересующая категория для перевода;
  - б) в поле «Исходное значение» введено корректное число;
  - в) в поле «Из» выбрана единица измерения;
  - г) в поле «В» выбрана единица измерения;
  - д) нажать кнопку «Конвертировать».
6. Ожидаемый результат: В поле «результат конвертации» выводиться корректный, правильный математически, перевод;
7. Фактический результат: В поле «результат конвертации» выводиться результат с ошибкой в расчетах;
8. Статус. Failed.



*Рисунок 17 – Ошибка 4*

1. Идентификатор: НР-004;



2. Название: Проверка корректности расчета по заданным параметрам;
3. Описание: Убедиться, что пользователь увидит корректный результат конвертации при вводе корректных значений;
4. Предварительные условия:
  - а) пользователь вводит корректные значения, соответствующие условиям для конвертации (неотрицательные).
5. Шаги выполнения:
  - а) в поле «Категории» выбрана интересующая категория для перевода;
  - б) в поле «Исходное значение» введено корректное число;
  - в) в поле «Из» выбрана единица измерения;
  - г) в поле «В» выбрана единица измерения;
  - д) нажать кнопку «Конвертировать».
6. Ожидаемый результат: В поле «результат конвертации» выводиться корректный, правильный математически, перевод;
7. Фактический результат: В поле «результат конвертации» выводиться результат с ошибкой в расчётах;
8. Статус. Failed.

## **6 АНАЛИЗ ДОКУМЕНТАЦИИ ДРУГОЙ КОМАНДЫ**

Документация, приложенная к проекту, соответствует требованиям к документации, описанным в техническом задании.

Руководство пользователя полностью описывает функционал приложения и порядок взаимодействия с ним.

## **7 ЗАКЛЮЧЕНИЕ**

При выполнении данной работы было написано техническое задание на разработку программы, разработана программа, создана документация для разработчика и пользователя, выявлены и описаны ошибки в программе. Для выполнения второй части практической работы был получен проект другой команды, проведено тестирование методом «черного ящика», выявленные ошибки описаны и проанализирована документация.