Руководство программиста

Телеграм бот «OverEdit»

Содержание

[1 Инсталляция программы: 4](#_Toc157660343)

[1.1 Пользовательский вариант 4](#_Toc157660344)

[1.2 Вариант поддержания и модернизации 4](#_Toc157660345)

[2 Проверка работоспособности чат-бота 5](#_Toc157660346)

[3 Поддержка программы 6](#_Toc157660347)

[4 Модернизация 7](#_Toc157660348)

# Инсталляция программы

## Пользовательский вариант

* Скачать файлы для работоспособности программы, которые находятся по ссылке <https://github.com/NataCher/OverEdit>.
* Для установки необходимо запустить файл main. В нем требуется указать токен telegram-bot для его работы.
* При отсутствии токена, создать его с помощью телеграмм-бота BotFather: <https://t.me/BotFather>.
* Проверить бота.

## Вариант поддержания и модернизации

* Убедитесь, что на вашем компьютере установлен Python, с библиотекой Telebot.
* Убедитесь, что на вашем смартфоне или персональном компьютере установлен telegram или вы можете войти в него в web версии.
* Склонируйте репозиторий программы с GitHub с помощью команды git clone https://github.com/NataCher/OverEdit.
* Перейдите в папку вашего проекта с помощью команды cd repository.
* Установите зависимости проекта, выполнив команду yarn install.
* Запустите программу для проверки успешности установки командой yarn start.
* Запустите телеграмм и перейдите в общение к боту по ссылке https://t.me/OverEdit\_NC\_bot.

# Проверка работоспособности чат-бота

Для проверки работоспособности требуется запустить общение с чат-ботом с помощью команды /start.

# Поддержка программы

Регулярно проверяйте обновления библиотеки telebot, Python и правил платформы Телеграмм по работе ботов.

Ведите документацию по программе, включая инструкции по устранению проблем и ответы на часто задаваемые вопросы, используя Markdown или другой удобный формат, и храните ее в репозитории на GitHub.

Мониторьте производительность программы и проводите оптимизацию кода, включая Python, и базы данных при необходимости.

Регулярно резервируйте данные программы, чтобы предотвратить потерю информации в случае сбоев системы, используя соответствующие инструменты и методы, и храните их в безопасном месте, отдельно от репозитория.

# Модернизация

Анализируйте обратную связь пользователей и их потребности, чтобы определить возможности для улучшения программы.

Создайте новую ветку в репозитории для разработки новых функций или улучшений с помощью команды git checkout -b feature-branch.

Разрабатывайте и реализуйте новые функции или улучшения, основываясь на исследованиях и требованиях пользователей, используя Python для разработки логики программы.

Тестируйте новые функции перед их выпуском, чтобы убедиться в их правильной работе и отсутствии ошибок, используя соответствующие инструменты для тестирования Python кода.

Обновляйте документацию программы, чтобы отразить новые функции и изменения, и убедитесь, что она соответствует актуальной версии программы, и отправьте изменения в репозиторий с помощью команды git push origin feature-branch.

Проводите регулярное обслуживание программы, включая оптимизацию кода, в том числе Python и telebot.