**Отчёт**

# **Введение**

Развитие голосовых технологий стремительно меняет наше представление о взаимодействии с компьютерами. Голосовые помощники, способные распознать и выполнить команду, уже стали привычной частью нашей жизни. Однако потенциал голосового управления гораздо шире, чем просто поиск информации или управление умным домом. Одним из наиболее популярных языков программирования для разработки голосовых помощников является Python, благодаря его простоте, гибкости и широким возможностям

## **Актуальность:**

Разработка музыкального плеера с голосовым управлением на Python отвечает сразу нескольким актуальным трендам: растущей популярности голосовых интерфейсов, доступности мощных инструментов распознавания речи и стремлению к максимальной "интеллектуализации" пользовательских устройств. Создание такого плеера — это шаг навстречу будущему, где управление музыкой будет осуществляться так же естественно, как обычный разговор.

# **Цель:**

Целью данной работы является разработка музыкального плеера с голосовым управлением на языке программирования Python, который позволит пользователю управлять воспроизведением музыки с помощью голосовых команд.

### **Задачи**:

1. Изучение основных принципов работы голосовых помощников и способов взаимодействия с пользователем через голос.
2. Разработать базовый функционал музыкального плеера: воспроизведение, пауза, остановка, переключение треков
3. Разработка алгоритма работы голосового помощника, включая функции распознавания команд, выполнения действий и взаимодействия с пользователем.
4. Протестировать работу разработанного приложения, оценить качество распознавания речи и удобство использования голосового интерфейса.

# **Методы и этапы работы:**

1. Изучение материалов по распознаванию и синтезу речи на языке программирования Python.
2. Реализация функций распознавания речи, включая возможность обработки и интерпретации команд пользователя. Создание функции для синтеза речи.
3. Анализ функций популярных голосовых помощников и последующее изучение библиотек для эти функций на языке программирования Python.
4. Создание полезных функций для работы с голосовым ассистентом, интегрируя их с функциям по распознаванию и синтезу речи.
5. Тестирование и отладка голосового помощника для обеспечения его качественной работы и соответствия требованиям пользователя.

# **Использованные программы**:

* Проект языке программирования Python.
* Код писался в среде разработки PyCharm.

# **Итоги**:

Данный музыкальный плеер, разработанный с использованием Python, включает в себе технологии распознавания речи, парсинга веб-страниц и автоматизации браузера.

Технологии:

• Распознавание речи: speech\_recognition – для преобразования голосовых команд пользователя в текст.

• Нечёткое сравнение строк: fuzzywuzzy – для обработки неточностей и вариаций в голосовых командах, обеспечивая гибкость и естественность взаимодействия.

• Парсинг веб-страниц: BeautifulSoup – для извлечения данных о треках и плейлистах с сайта YouTube Music.

• Автоматизация браузера: Selenium – для управления браузером, имитации действий пользователя, таких как поиск, выбор трека и управление воспроизведением на сайте YouTube Music.

Функциональность:

1. Голосовой поиск треков:

• Пользователь произносит название трека или имя исполнителя.

• Система распознает речь и преобразует её в текст.

• С помощью Selenium плеер открывает сайт YouTube Music и вводит поисковый запрос.

• BeautifulSoup анализирует результаты поиска, извлекая самый лучший результат.

• Данный результат Selenium открывает в YouTube Music и производит запуск трека.

2. Воспроизведение треков из плейлиста пользователя на YouTube Music:

• Пользователь запускает воспроизведение плейлиста, просто назвав его имя, которое уже связано с соответствующей ссылкой в программе.

• Система распознаёт запрос и ищет плейлист на YouTube Music.

• BeautifulSoup собирает информацию о треках в плейлисте.

• Плеер запускает воспроизведение треков из выбранного плейлиста.

Дополнительные возможности:

• Управление воспроизведением: голосовые команды для паузы, воспроизведения, переключения треков, выбора трека, регулировки громкости*.*

# **Инструкция для пользователя:**

Описание процесса работы музыкального плеера в виде инструкции для пользователя:

1. Начало работы. При старте приложения пользователю предлагается выбрать режим работы: поиск музыки на YouTube Music или прослушивание своего плейлиста.

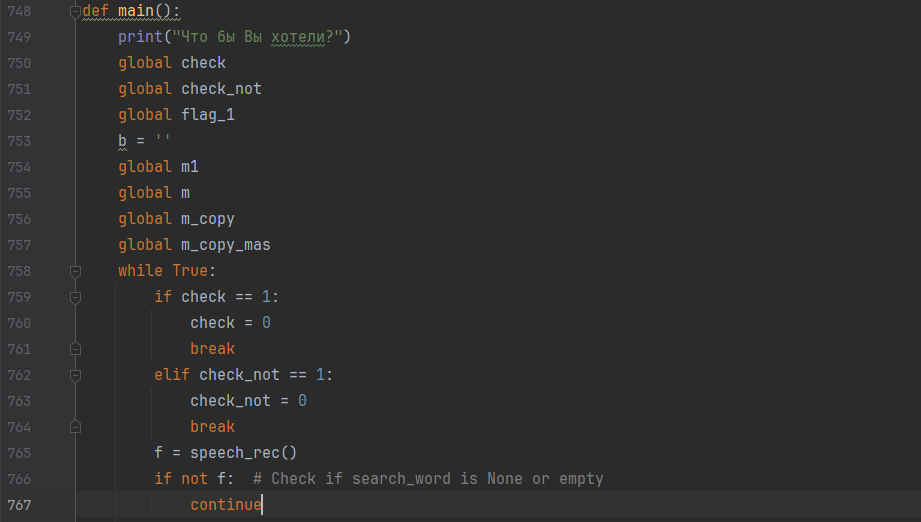


Рис. 1. Пример части функции, отвечающей за выбор режима.

1. Диктовка команды. Вы можете диктовать свою команду помощнику вслух. Например, вы можете сказать "Включи Лесник Король и Шут" или "Включи мой плейлист".

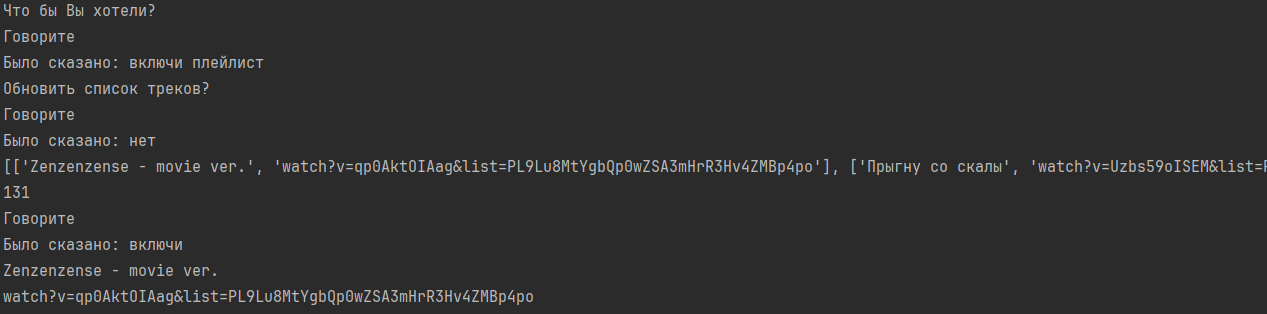


Рис. 2. Пример выбора режима воспроизведения треков из плейлиста

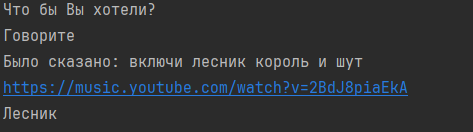


Рис. 3. Пример выбора режима поиска трека

1. Поиска трека. В этом режиме пользователь может найти любой трек на YouTube Music. Найденные треки добавляются в очередь воспроизведения, которой можно управлять с помощью голосовых команд (например, "следующий", "пауза", "включи…").



Рис. 4. Пример части функции, отвечающей за режим поиска трека

1. Прослушивание плейлиста. Чтобы воспользоваться этим режимом, пользователь сначала добавляет ссылку на свой плейлист YouTube Music в код программы. Плеер сохраняет ссылки на все треки из плейлиста в отдельный файл. После этого пользователь может управлять воспроизведением треков с помощью голосовых команд.



Рис. 5. Пример части функции, отвечающей за режим воспроизведения плейлиста

1. Смена режима. Пользователь может свободно переключаться между режимами поиска и прослушивания плейлистов с помощью голосовых команд.

# **Выводы.**

1. Был разработан музыкальный плеер на языке программирования Python, который умеет откликаться на голосовые команды, ищет музыку на YouTube Music, управляет воспроизведением и также работает с пользовательскими плейлистами.
2. Разработка голосового музыкального плеера позволила приобрести ценный опыт работы с технологиями распознавания речи и обработки естественного языка на Python. Полученные знания и навыки могут быть применены для создания более сложных голосовых ассистентов и других систем искусственного интеллекта.

# **Перспективы.**

* Разработать приложение для музыкального плеера, которое будет работать как на ПК, так и на смартфонах
* Реализовать функционал для работы со всеми плейлистами пользователя на YouTube Music, не ограничиваясь добавлением ссылок вручную.
* Добавить возможность интеграции с популярными музыкальными сервисами (Яндекс Музыка, Spotify, VK Музыка и т.д.)
* Найти более эффективные решения для оптимизации производительности и потребления ресурсов
* Рассмотреть возможность доработки программы до коммерческого уровня.