# Ц**ель работы**

Изучение классификации видов тестирования программных систем, а также анализ применимых видов тестирования для разных программных проектов и выявление дефектов качества для каждого вида тестирования.

# Описание программного проекта

Программный проект разработан на Python 3.8.

Основные функции:

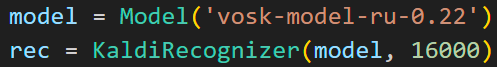
1. **Запуск и настройка аудиопотока:**
   * Инициализация аудиопотока с помощью библиотеки PyAudio.
   * Настройка параметров аудиопотока, таких как формат, количество каналов, частота дискретизации и размер буфера.
2. **Распознавание речи в реальном времени:**
   * Использование модели Vosk для распознавания русской речи.
   * Обработка входящего аудиопотока и распознавание речи в реальном времени.
   * Вывод распознанного текста в консоль.
3. **Обработка команд:**
   * Реализация цикла прослушивания команд.
   * Завершение работы программы при получении команды "пока".

Используемые библиотеки:

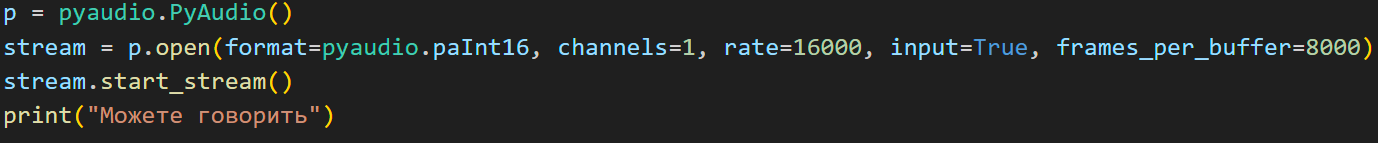
1. **Vosk API:**
   * Библиотека для офлайн-распознавания речи, поддерживающая множество языков, включая русский.
2. **PyAudio:**
   * Библиотека для работы с аудиопотоками, позволяющая записывать и воспроизводить звук.

Основные компоненты проекта:

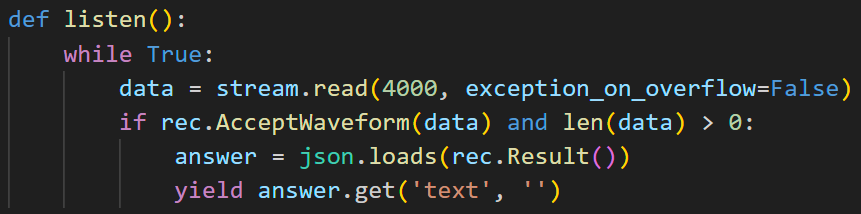
1. **Инициализация модели распознавания речи:**



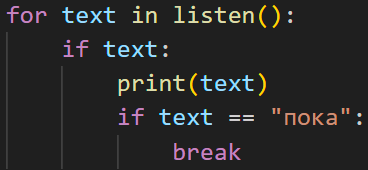
1. **Настройка аудиопотока:**



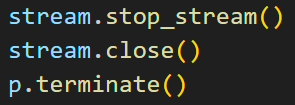
1. **Функция прослушивания и распознавания речи:**



1. **Основной цикл обработки команд:**



1. **Завершение работы:**



Функционал программы:

* Программа работает в реальном времени, распознавая и обрабатывая голосовые команды пользователя.
* Используется офлайн-модель распознавания речи, что позволяет использовать программу без доступа к интернету.
* Программа завершает работу по голосовой команде "пока", что обеспечивает удобство использования.

# Определение видов, типов, методов, уровней тестирования. Дефекты.

1. Уровни тестирования:

* **Компонентное (модульное) тестирование:**

Тестирование функции listen() для правильного распознавания и обработки аудиоданных.

* **Интеграционное тестирование:**

Тестирование взаимодействия между PyAudio и Vosk, а также корректного взаимодействия между аудиопотоком, распознавателем речи и обработчиком команд.

* **Системное тестирование:**

Проверка всей системы распознавания речи от начала до конца, включая инициализацию, распознавание речи и завершение по команде "пока".

1. Типы тестирования:

**Функциональное тестирование:**

Тестирование, что система правильно распознает команду "пока" и завершает работу.

**Нефункциональное тестирование:**

* Нагрузочное тестирование системы при обработке большого количества голосовых команд подряд;
* Стресс-тестирование при резком увеличении шума в окружающей среде.

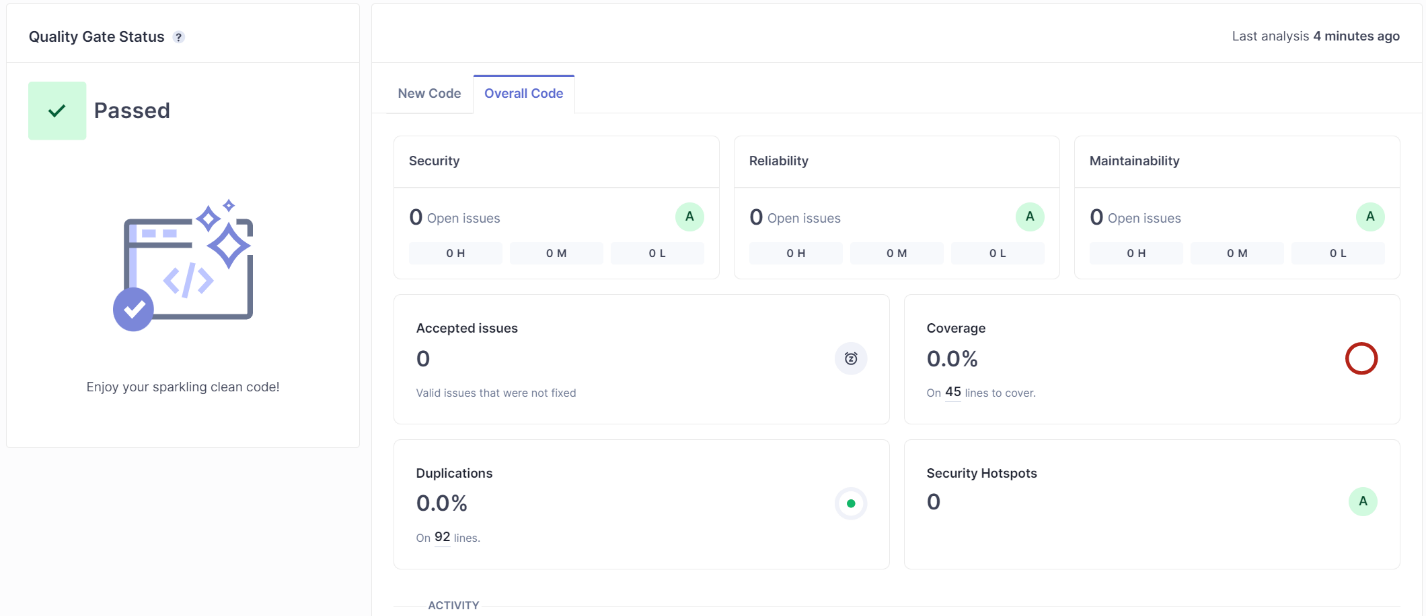
1. Методы тестирования:
2. **Статические методы:**

* **Рецензирование:** Ручная проверка кода и документации.

Рецензирование кода для обнаружения ошибок логики или нарушений стиля кодирования.

* **Статический анализ:** Автоматизированная проверка кода с помощью инструментов.

Использование инструментов, таких как pylint, для анализа кода на предмет ошибок и предупреждений. В данном случае используем SonarQube.



1. **Динамические методы:**

* **Методы белого ящика:** Тестирование с доступом к исходному коду.

Тестирование функций распознавания речи с проверкой внутренних состояний и переменных.

* **Методы черного ящика:** Тестирование без знания внутренней структуры системы.

Тестирование системы пользователем, вводя различные голосовые команды и проверяя корректность ответа.

* **Методы серого ящика:** Комбинация методов белого и черного ящика.

Тестирование взаимодействия между аудиопотоком и моделью Vosk с доступом к логам и отладочной информации.

1. Виды тестирования:

* **Функциональное тестирование:**

Проверка корректного распознавания команд и выполнения соответствующих действий.

**Дефект**: Неправильное распознавание «слов-команд».

* **Навигационное тестирование:**

Проверка корректного реагирования системы на команды навигации, такие как "пока".

**Дефект:** Неправильное распознавание слова «пока». Ошибка при замене контрольного слова выключения.

* **Инсталляционное тестирование:**

Проверка правильной установки программы на различных операционных системах и ее корректной работы после установки. Проверка на наличие Python компилятора.

**Дефект:** Отсутствие навыков программирования.

* **Нагрузочное тестирование:**

Проверка производительности системы при обработке большого объема данных. Работа с другими запущенными программами и подобное.

**Дефект**: Нехватка ресурсов.

* **Ресурсные тесты:**

**Проверка работоспособности функции listen() в течении длительного времени.**

**Дефекты: Нехватка ресурсов.**

* **Юзабилити-тестирование:**

Оценка удобства использования программы пользователями с различными уровнями технической подготовки. В данном случае оценивается, как экспертное знание.

**Дефекты**: Простой пользователь, который не знаком с компиляторами не может запустить код.

* **Локализация:**

**Тестирование на распознавание слов, не входящих в установленный языковой модуль.**

**Дефект: Неправильное распознавание слов**

* **Документация:**

**Проверка на выполнение требований функционала программного проекта и его элементов.**

**Дефекты: Несоответствие функционала программного проекта заданным требованиям.**

# Выводы по работе

В видах тестирования прописаны возможные дефекты.

Виды и методы тестирования определяются исходя из установленных требований при разработке программного проекта. Лучший метод для меня – это метод серого ящика, так как совмещает в себе тестирование разработчиком и целевой аудиторией программного проекта.

# Список использованных источников

1. Учебное пособие "Основы управления качеством программных средств" Т. В. Афанасьева А. Н. Афанасьев 2017г;