

Jarvis

v.0.0.1

Создано системой Doxygen 1.8.13

Оглавление

Глава 1

Алфавитный указатель пространств имен

1.1 Пространства имен

Полный список документированных пространств имен.

Arduino	Для работы с Arduino	??
Commands	Namespace для хранения всего, что связано с выполнением команд	??
CommandType	Хранение констант - основных типов команд	??
connection	Для сетевых подключений	??
Connection	Для подключения к Arduino	??
Jarvis	Основной namespace для всего проекта	??

Глава 2

Иерархический список классов

2.1 Иерархия классов

Иерархия классов.

Jarvis::Commands::Command	??
Jarvis	??
noncopyable	
Jarvis::Jarvis	??
Jarvis::Parser	??
Jarvis::Sentence	??
Jarvis::Arduino::Connection::SerialPort	??
Jarvis::connection::Transport	??
Jarvis::connection::SpeechKit	??
Jarvis::Voice	??

Глава 3

Алфавитный указатель классов

3.1 Классы

Классы с их кратким описанием.

Jarvis::Commands::Command	
Класс, отвечающий за выполнение команд	??
Jarvis::Jarvis	??
Jarvis	
Класс, описывающий аналог ИИ	??
Jarvis::Parser	
Класс для парсинга файлов различных типов	??
Jarvis::Sentence	
Класс для работы с предложениями к Jarvis	??
Jarvis::Arduino::Connection::SerialPort	
Класс для подключения по порту	??
Jarvis::connection::SpeechKit	
Класс для "общения" с сервером YandexSpeechKitCloud	??
Jarvis::connection::Transport	
Абстрактный класс для "общения" с сервером	??
Jarvis::Voice	
Описывает объект голос	??

Глава 4

Пространства имен

4.1 Пространство имен Arduino

Для работы с [Arduino](#).

4.1.1 Подробное описание

Для работы с [Arduino](#).

4.2 Пространство имен Commands

namespace для хранения всего, что связано с выполнением команд

4.2.1 Подробное описание

namespace для хранения всего, что связано с выполнением команд

4.3 Пространство имен CommandType

Хранение констант - основных типов команд

4.3.1 Подробное описание

Хранение констант - основных типов команд

4.4 Пространство имен Connection

Для подключения к [Arduino](#).

4.4.1 Подробное описание

Для подключения к [Arduino](#).

4.5 Пространство имен Connection

Для подключения к [Arduino](#).

4.5.1 Подробное описание

Для подключения к [Arduino](#).

4.6 Пространство имен Jarvis

Основной namespace для всего проекта

Классы

- class [Jarvis](#)
- class [Parser](#)
 - Класс для парсинга файлов различных типов
- class [Sentence](#)
 - Класс для работы с предложениями к [Jarvis](#).
- class [Voice](#)
 - Описывает объект голос

Функции

- Parser::Tree [jsonParse](#) (Parser::stringTarget &stream)
- Parser::Tree [xmlParse](#) (Parser::stringTarget &stream)
- bool [operator==](#) (const [Sentence](#) &lhs, const [Sentence](#) &rhs)
- bool [operator!=](#) (const [Sentence](#) &lhs, const [Sentence](#) &rhs)
- bool [operator>](#) (const [Sentence](#) &lhs, const [Sentence](#) &rhs)
- bool [operator<](#) (const [Sentence](#) &lhs, const [Sentence](#) &rhs)
- bool [operator>=](#) (const [Sentence](#) &lhs, const [Sentence](#) &rhs)
- bool [operator<=](#) (const [Sentence](#) &lhs, const [Sentence](#) &rhs)

4.6.1 Подробное описание

Основной namespace для всего проекта

4.6.2 Функции

4.6.2.1 jsonParse()

```
Parser::Tree Jarvis::jsonParse (
    Parser::stringTarget & stream )
```

Возвращает

Дерево, распарсенного файла

4.6.2.2 operator!=(())

```
bool Jarvis::operator!=(
    const Sentence & lhs,
    const Sentence & rhs )
```

Аргументы

lhs	Левый операнд
rhs	Правый операнд

Возвращает

bool Результат сравнения

4.6.2.3 operator<()

```
bool Jarvis::operator< (
    const Sentence & lhs,
    const Sentence & rhs )
```

Аргументы

lhs	Левый операнд
rhs	Правый операнд

Возвращает

bool Результат сравнения

4.6.2.4 operator<=()

```
bool Jarvis::operator<= (
    const Sentence & lhs,
    const Sentence & rhs )
```

Аргументы

lhs	Левый операнд
rhs	Правый операнд

Возвращает

bool Результат сравнения

4.6.2.5 operator==()

```
bool Jarvis::operator==(
    const Sentence & lhs,
    const Sentence & rhs )
```

Аргументы

lhs	Левый операнд
rhs	Правый операнд

Возвращает

bool Результат сравнения

4.6.2.6 operator>()

```
bool Jarvis::operator> (
    const Sentence & lhs,
    const Sentence & rhs )
```

Аргументы

lhs	Левый операнд
rhs	Правый операнд

Возвращает

bool Результат сравнения

4.6.2.7 operator>=()

```
bool Jarvis::operator>= (
    const Sentence & lhs,
    const Sentence & rhs )
```

Аргументы

lhs	Левый операнд
rhs	Правый операнд

Возвращает

bool Результат сравнения

4.6.2.8 xmlParse()

```
Parser::Tree Jarvis::xmlParse (
    Parser::stringTarget & stream )
```

Возвращает

Дерево, распарсенного файла

Глава 5

Классы

5.1 Класс Jarvis::Commands::Command

Класс, отвечающий за выполнение команд

```
#include <Command.hpp>
```

Открытые статические члены

- static void **execute** (const CommandType::commandType &type, const scriptName &script, const argumentsList arguments={})
Подготовка и отправка данных

5.1.1 Подробное описание

Класс, отвечающий за выполнение команд

5.1.2 Методы

5.1.2.1 execute()

```
static void Jarvis::Commands::Command::execute (  
    const CommandType::commandType & type,  
    const scriptName & script,  
    const argumentsList arguments = {} ) [static]
```

Подготовка и отправка данных

Аргументы

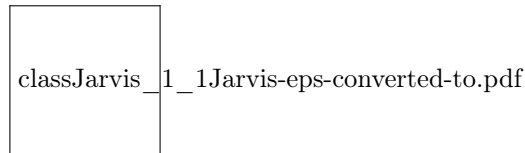
type	Тип скрипта
script	Название скрипта
arguments	Список аргументов скрипта (по умолчанию пустой) Статический метод сбора данных и запуска команды

Объявления и описания членов класса находятся в файле:

- /Users/vladislavpereskokov/Desktop/Jarvis/Voice/include/Command.hpp

5.2 Класс Jarvis::Jarvis

Граф наследования:Jarvis::Jarvis:



Открытые члены

- `Jarvis ()=default`
Конструктор по-умолчанию
- `Jarvis (Voice &voice)`
Конструктор
- `~Jarvis ()=default`
Деструктор
- `Jarvis (const Jarvis ©)=delete`
- `Jarvis (Jarvis &©)=delete`
- `Jarvis & operator= (const Jarvis ©)=delete`
- `Jarvis & operator= (Jarvis &©)=delete`
- `~JarvisDestroyer ()`
Деструктор
- `void initialize (Jarvis *jarvis)`
Инициализация системы
- `JarvisDestroyer (const JarvisDestroyer ©)=delete`
- `JarvisDestroyer (JarvisDestroyer &©)=delete`
- `JarvisDestroyer & operator= (const JarvisDestroyer ©)=delete`
- `JarvisDestroyer & operator= (JarvisDestroyer &©)=delete`

Открытые статические члены

- `static Jarvis & instance ()`
Инициализация системы
- `static void say (Voice &voice, const sayPhrase &phrase)`
Воспроизведение фразы
- `static stringVoice toArduino ()`
Статический метод для запуска скрипта
- `static stringVoice sendToYandexSpeechKit ()`
Статический метод для "общения" с сервером Яндекса
- `static void sendToSerialPort (SerialPort &serial, const std::string &command)`
Статический метод для "общения" с портом Arduino.
- `static map getMapFromYandexSpeechKit (const pathToConfigs &path)`
Статический метод для "общения" получения данных с сервера Yandex.

Открытые атрибуты

- private structs: struct JarvisDestroyer : boost::noncopyable { public methods: JarvisDestroyer() = default
- private params: Jarvis * _jarvis
- Voice _voice

Статические открытые данные

- static JarvisDestroyer _destroyer

Друзья

- struct JarvisDestroyer

5.2.1 Конструктор(ы)

5.2.1.1 Jarvis()

Jarvis::Jarvis::Jarvis (
Voice & voice)

Конструктор

Аргументы

voice	Голос Jarvis'a
-------	----------------

5.2.2 Методы

5.2.2.1 getMapFromYandexSpeechKit()

static map Jarvis::Jarvis::getMapFromYandexSpeechKit (
const pathToConfigs & path) [static]

Статический метод для "общения" получения данных с сервера Yandex.

Аргументы

path	Путь до конфигурационного файла
------	---------------------------------

Возвращает

map Отображение вида std::string -> std::string

5.2.2.2 initialize()

```
void Jarvis::Jarvis::initialize (
    Jarvis * jarvis )
```

Инициализация системы

Аргументы

jarvis	Объект Jarvis Метод для инициализации системы
--------	---

5.2.2.3 instance()

```
static Jarvis& Jarvis::Jarvis::instance ( ) [static]
```

Инициализация системы

Возвращает

Jarvis возвращает тип единственного объекта Jarvis Метод для инициализации системы

5.2.2.4 say()

```
static void Jarvis::Jarvis::say (
    Voice & voice,
    const sayPhrase & phrase ) [static]
```

Воспроизведение фразы

Аргументы

voice	Голос Jarvis
phrase	Фраза, которую Jarvis должен воспроизвести голосом Метод для воспроизведения фразы, которую должен сказать Jarvis

5.2.2.5 sendToSerialPort()

```
static void Jarvis::Jarvis::sendToSerialPort (
    SerialPort & serial,
    const std::string & command ) [static]
```

Статический метод для "общения" с портом [Arduino](#).

Аргументы

serial	Порт, по которому будет отправлен сигнал
command	Команда, которая будет отправлена

5.2.2.6 sendToYandexSpeechKit()

```
static stringVoice Jarvis::Jarvis::sendToYandexSpeechKit ( ) [static]
```

Статический метод для "общения" с сервером Яндекса

Возвращает

stringVoice Строка, преобразованная из голоса

5.2.2.7 toArduino()

```
static stringVoice Jarvis::Jarvis::toArduino ( ) [static]
```

Статический метод для запуска скрипта

Возвращает

stringVoice Строка, преобразованная из голоса

5.2.3 Данные класса

5.2.3.1 _destroyer

```
JarvisDestroyer Jarvis::Jarvis::_destroyer [static]
```

Статический объект, отвечающий за разрушение [Jarvis'a](#)

5.2.3.2 `_voice`

Voice Jarvis::Jarvis::_voice

Голос [Jarvis](#)

5.2.3.3 `params`

private Jarvis::Jarvis::params

Объект [Jarvis](#)

Статический объект [Jarvis](#)

Объявления и описания членов класса находятся в файле:

- `/Users/vladislavpereskokov/Desktop/Jarvis/Voice/include/Jarvis.hpp`

5.3 Класс Jarvis

Класс, описывающий аналог ИИ

```
#include <Jarvis.hpp>
```

5.3.1 Подробное описание

Класс, описывающий аналог ИИ

Объявления и описания членов класса находятся в файле:

- `/Users/vladislavpereskokov/Desktop/Jarvis/Voice/include/Jarvis.hpp`

5.4 Класс Jarvis::Parser

Класс для парсинга файлов различных типов

```
#include <Parser.hpp>
```

Открытые члены

- `Parser ()`=default
Конструктор по-умолчанию
- `Parser (const Path &path)`
Конструктор
- `Parser (const Parser ©)`=default
Конструктор копирования
- `Parser & operator= (const Parser ©)`=default
Оператор копирования
- `~Parser ()`=default
Деструктор
- `void loadTarget (const Path &path)`
Метод для подгрузки файла
- `Tree jsonParse ()`
Парсер json файла
- `Tree xmlParse ()`
Парсер xml файла

Открытые атрибуты

- `private params: Path _target`
- `Tree _tree`

Друзья

- `Tree jsonParse (stringTarget &stream)`
Парсер json файла
- `Tree xmlParse (stringTarget &stream)`
Парсер xml файла

5.4.1 Подробное описание

Класс для парсинга файлов различных типов

5.4.2 Конструктор(ы)

5.4.2.1 Parser()

```
Jarvis::Parser::Parser (  
    const Path & path ) [explicit]
```

Конструктор

Аргументы

path	Путь до файла
------	---------------

5.4.3 Методы

5.4.3.1 jsonParse()

```
Tree Jarvis::Parser::jsonParse ( )
```

Парсер json файла

Возвращает

Дерево, распарсенного файла

5.4.3.2 loadTarget()

```
void Jarvis::Parser::loadTarget (
    const Path & path )
```

Метод для подгрузки файла

Аргументы

path	Путь до файла
------	---------------

5.4.3.3 xmlParse()

```
Tree Jarvis::Parser::xmlParse ( )
```

Парсер xml файла

Возвращает

Дерево, распарсенного файла

5.4.4 Документация по друзьям класса и функциям, относящимся к классу

5.4.4.1 jsonParse

```
Tree jsonParse (
    Parser::stringTarget & stream ) [friend]
```

Парсер json файла

Возвращает

Дерево, распарсенного файла

5.4.4.2 xmlParse

```
Tree xmlParse (
    Parser::stringTarget & stream ) [friend]
```

Парсер xml файла

Возвращает

Дерево, распарсенного файла

5.4.5 Данные класса

5.4.5.1 _tree

```
Tree Jarvis::Parser::_tree
```

Дерево

5.4.5.2 params

```
private Jarvis::Parser::params
```

путь до файла

Объявления и описания членов класса находятся в файле:

- /Users/vladislavpereskokov/Desktop/Jarvis/Voice/include/Parser.hpp

5.5 Класс Jarvis::Sentence

Класс для работы с предложениями к [Jarvis](#).

```
#include <Sentence.hpp>
```

Открытые члены

- [Sentence](#) (const phrase &phrase)
Конструктор
- [Sentence](#) (const [Sentence](#) ©)
Конструктор копирования
- [Sentence](#) ([Sentence](#) &©)
Конструктор копирования
- [Sentence](#) & [operator=](#) (const [Sentence](#) ©)
Оператор копирования
- [Sentence](#) & [operator=](#) ([Sentence](#) &©)
Оператор копирования
- [~Sentence](#) ()=default
Деструктор
- void [setSentence](#) (const [Sentence](#) &sentence)
Метод добавления фразы
- void [setSentence](#) (const phrase &phrase)
Метод добавления фразы
- phrase [getSentence](#) () const
Метод добавления фразы

Открытые атрибуты

- private [params](#): phrase _phrase

Друзья

- bool [operator==](#) (const [Sentence](#) &lhs, const [Sentence](#) &rhs)
Оператор сравнения
- bool [operator!=](#) (const [Sentence](#) &lhs, const [Sentence](#) &rhs)
Оператор сравнения
- bool [operator>](#) (const [Sentence](#) &lhs, const [Sentence](#) &rhs)
Оператор больше
- bool [operator<](#) (const [Sentence](#) &lhs, const [Sentence](#) &rhs)
Оператор меньше
- bool [operator>=](#) (const [Sentence](#) &lhs, const [Sentence](#) &rhs)
Оператор больше или равно
- bool [operator<=](#) (const [Sentence](#) &lhs, const [Sentence](#) &rhs)
Оператор меньше или равно

5.5.1 Подробное описание

Класс для работы с предложениями к [Jarvis](#).

5.5.2 Конструктор(ы)

5.5.2.1 Sentence() [1/3]

```
Jarvis::Sentence::Sentence (
    const phrase & phrase )
```

Конструктор

Аргументы

phrase	Фраза
--------	-------

5.5.2.2 Sentence() [2/3]

```
Jarvis::Sentence::Sentence (
    const Sentence & copy )
```

Конструктор копирования

Аргументы

copy	Копируемый объект
------	-------------------

5.5.2.3 Sentence() [3/3]

```
Jarvis::Sentence::Sentence (
    Sentence && copy )
```

Конструктор копирования

Аргументы

copy	Копируемый объект
------	-------------------

5.5.3 Методы

5.5.3.1 getSentence()

```
phrase Jarvis::Sentence::getSentence ( ) const
```

Метод добавления фразы

Возвращает

Фраза (строка)

5.5.3.2 operator=() [1/2]

```
Sentence& Jarvis::Sentence::operator= (
    const Sentence & copy )
```

Оператор копирования

Аргументы

copy	Копируемый объект
------	-------------------

5.5.3.3 operator=() [2/2]

```
Sentence& Jarvis::Sentence::operator= (
    Sentence && copy )
```

Оператор копирования

Аргументы

copy	Копируемый объект
------	-------------------

5.5.3.4 setSentence() [1/2]

```
void Jarvis::Sentence::setSentence (
    const Sentence & sentence )
```

Метод добавления фразы

Аргументы

sentence	Фраза
----------	-------

5.5.3.5 setSentence() [2/2]

```
void Jarvis::Sentence::setSentence (
    const phrase & phrase )
```

Метод добавления фразы

Аргументы

phrase	Фраза (строка)
--------	----------------

5.5.4 Документация по друзьям класса и функциям, относящимся к классу

5.5.4.1 operator!=

```
bool operator!= (
    const Sentence & lhs,
    const Sentence & rhs ) [friend]
```

Оператор сравнения

Аргументы

lhs	Левый операнд
rhs	Правый операнд

Возвращает

bool Результат сравнения

5.5.4.2 operator<

```
bool operator< (
    const Sentence & lhs,
    const Sentence & rhs ) [friend]
```

Оператор меньше

Аргументы

lhs	Левый операнд
rhs	Правый операнд

Возвращает

bool Результат сравнения

5.5.4.3 operator<=

```
bool operator<= (
    const Sentence & lhs,
    const Sentence & rhs ) [friend]
```

Оператор меньше или равно

Аргументы

lhs	Левый операнд
rhs	Правый операнд

Возвращает

bool Результат сравнения

5.5.4.4 operator==

```
bool operator== (
    const Sentence & lhs,
    const Sentence & rhs ) [friend]
```

Оператор сравнения

Аргументы

lhs	Левый операнд
rhs	Правый операнд

Возвращает

bool Результат сравнения

5.5.4.5 operator>

```
bool operator> (
    const Sentence & lhs,
    const Sentence & rhs ) [friend]
```

Оператор больше

Аргументы

lhs	Левый операнд
rhs	Правый операнд

Возвращает

bool Результат сравнения

5.5.4.6 operator>=

```
bool operator>= (
    const Sentence & lhs,
    const Sentence & rhs ) [friend]
```

Оператор больше или равно

Аргументы

lhs	Левый операнд
rhs	Правый операнд

Возвращает

bool Результат сравнения

5.5.5 Данные класса

5.5.5.1 params

```
private Jarvis::Sentence::params
```

Строка

Объявления и описания членов класса находятся в файле:

- /Users/vladislavpereskokov/Desktop/Jarvis/Voice/include/Sentence.hpp

5.6 Класс Jarvis::Arduino::Connection::SerialPort

Класс для подключения по порту

```
#include <SerialPort.hpp>
```

Открытые члены

- `SerialPort` (const portName &name, const portRate rate)
Конструктор
- `~SerialPort` ()=default
Деструктор
- void `write` (const message &message)
Метод для отправки сообщения
- message `read` ()
Метод для получения сообщения
- `SerialPort` (const `SerialPort` ©)=delete
- `SerialPort` (const `SerialPort` &©)=delete
- `SerialPort` & operator= (const `SerialPort` ©)=delete
- void `setRate` (port &port, const portRate rate)
Заполнение информацией

Открытые атрибуты

- private `params`: serviceIO _service
- port `_port`

5.6.1 Подробное описание

Класс для подключения по порту

5.6.2 Конструктор(ы)

5.6.2.1 SerialPort()

```
Jarvis::Arduino::Connection::SerialPort::SerialPort (
    const portName & name,
    const portRate rate )
```

Конструктор

Аргументы

name	Имя порта для подключения
rate	Сокрость передачи данных

5.6.3 Методы

5.6.3.1 read()

```
message Jarvis::Arduino::Connection::SerialPort::read ( )
```

Метод для получения сообщения

Возвращает

message Сообщение

5.6.3.2 setRate()

```
void Jarvis::Arduino::Connection::SerialPort::setRate (
    port & port,
    const portRate rate )
```

Заполнение информацией

Аргументы

port	Порт для "общения"
rate	Скорость для передачи данных

5.6.3.3 write()

```
void Jarvis::Arduino::Connection::SerialPort::write (
    const message & message )
```

Метод для отправки сообщения

Аргументы

message	Сообщение
---------	-----------

5.6.4 Данные класса

5.6.4.1 _port

```
port Jarvis::Arduino::Connection::SerialPort::_port
```

порт

5.6.4.2 params

```
private Jarvis::Arduino::Connection::SerialPort::params
```

I/O сервис

Объявления и описания членов класса находятся в файле:

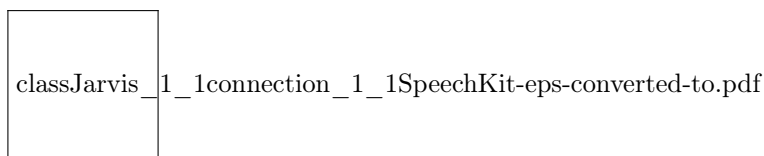
- /Users/vladislavpereskokov/Desktop/Jarvis/Voice/include/SerialPort.hpp

5.7 Класс Jarvis::connection::SpeechKit

Класс для "общения" с сервером YandexSpeechKitCloud.

```
#include <SpeechKit.hpp>
```

Граф наследования: Jarvis::connection::SpeechKit:



Открытые члены

- [SpeechKit](#) (const Transport::jPath &path, const Transport::OptionsList::yandexOptions yandexOptions)
Конструктор
- [~SpeechKit](#) ()=default
Деструктор
- bool [send](#) () override
Отправка данных
- void [setOptions](#) () override
Задание параметров отправки
- [SpeechKit](#) (const [SpeechKit](#) ©)=delete
- [SpeechKit](#) & operator= (const [SpeechKit](#) ©)=delete

Дополнительные унаследованные члены

5.7.1 Подробное описание

Класс для "общения" с сервером YandexSpeechKitCloud.

5.7.2 Конструктор(ы)

5.7.2.1 SpeechKit()

```
Jarvis::connection::SpeechKit::SpeechKit (
    const Transport::jPath & path,
    const Transport::OptionsList::yandexOptions yandexOptions )
```

Конструктор

Аргументы

path	Путь до конфигурационного файла
yandexOptions	Список нужных ключей для конфигурационного файла

5.7.3 Методы

5.7.3.1 send()

bool Jarvis::connection::SpeechKit::send () [override], [virtual]

Отправка данных

Возвращает

bool Подтверждение отправки

Замещает [Jarvis::connection::Transport](#).

Объявления и описания членов класса находятся в файле:

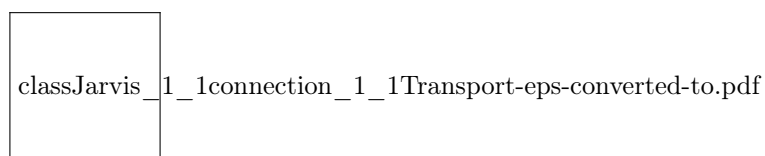
- /Users/vladislavpereskokov/Desktop/Jarvis/Voice/include/SpeechKit.hpp

5.8 Класс Jarvis::connection::Transport

Абстрактный класс для "общения" с сервером

```
#include <Transport.hpp>
```

Граф наследования:Jarvis::connection::Transport:



Открытые типы

- using [property](#) = [OptionsList](#)

Определяет тип для параметров отправки

Открытые члены

- `OptionsList` (const jPath &path, const yandexOptions yandexOptions)
Конструктор
- optionValue `getOption` (const option &option)
Метод получения значения параметра
- `~OptionsList` ()=default
Деструктор
- `OptionsList` ()=delete
- `OptionsList` (const OptionsList ©)=delete
- `OptionsList` & operator= (const `OptionsList` ©)=delete
- void `fillList` (const jList &list)
Метод заполнения списка параметров
- url `makeUrl` (const jList &list, url &url)
Метод составления URL'a из ключей и их значений для post запроса
- bool `findYandexOption` (const yandexOption &option)
Метод поиска ключевых параметров
- `Transport` (const jPath &path, const OptionsList::yandexOptions yandexOptions)
Конструктор
- virtual `~Transport` ()
Деструктор
- virtual bool `send` ()=0
Отправка данных
- responseString `recv` () const
Получение данных
- bool `connect` ()
Соединение
- bool `isConnect` () const
Проверка соединения
- bool `checkConnection` (const socket *curl) const
Проверка соединения
- virtual void `setOptions` ()=0
Задание параметров отправки
- `Transport` (const `Transport` ©)=delete
- `Transport` & operator= (const `Transport` ©)=delete

Открытые атрибуты

- protected `params`: optionList _optList
- private `params`: yandexOptions _yandexOptions
- `property` _properties
- headerList * _headers
- response _response

5.8.1 Подробное описание

Абстрактный класс для "общения" с сервером

5.8.2 Определения типов

5.8.2.1 property

using Jarvis::connection::Transport::property = [OptionsList](#)

Определяет тип для параметров отправки

OptionsList property

5.8.3 Конструктор(ы)

5.8.3.1 Transport()

```
Jarvis::connection::Transport::Transport (
    const jPath & path,
    const OptionsList::yandexOptions yandexOptions ) [explicit]
```

Конструктор

Аргументы

path	Путь до конфигурационного файла
yandexOptions	Список нужных ключей для конфигурационного файла

5.8.4 Методы

5.8.4.1 checkConnection()

```
bool Jarvis::connection::Transport::checkConnection (
    const socket * curl ) const
```

Проверка соединения

Аргументы

curl	Сокет для проверки
------	--------------------

Возвращает

bool Подтверждение соединения

5.8.4.2 connect()

```
bool Jarvis::connection::Transport::connect ( )
```

Соединение

Возвращает

bool Подтверждение соединения

5.8.4.3 fillList()

```
void Jarvis::connection::Transport::fillList (
    const jList & list )
```

Метод заполнения списка параметров

Аргументы

list	Отображение из распарсенного дерева
------	-------------------------------------

5.8.4.4 findYandexOption()

```
bool Jarvis::connection::Transport::findYandexOption (
    const yandexOption & option )
```

Метод поиска ключевых параметров

Аргументы

option	Искомый параметр
--------	------------------

Возвращает

bool Подтверждение - найдено ли

5.8.4.5 getOption()

```
optionValue Jarvis::connection::Transport::getOption (
    const option & option )
```

Метод получения значения параметра

Аргументы

option	Искомый параметр
--------	------------------

Возвращает

optionValue Значение заданного параметра

5.8.4.6 isConnected()

```
bool Jarvis::connection::Transport::isConnect ( ) const
```

Проверка соединения

Возвращает

bool Подтверждение соединения

5.8.4.7 makeUrl()

```
url Jarvis::connection::Transport::makeUrl (
    const jList & list,
    url & url )
```

Метод составления URL'a из ключей и их значений для post запроса

Аргументы

list	Отображение из распарсенного дерева
url	Сырой url

Возвращает

url URL для отправки

5.8.4.8 OptionsList()

```
Jarvis::connection::Transport::OptionsList (
    const jPath & path,
    const yandexOptions yandexOptions ) [explicit]
```

Конструктор

Аргументы

path	Путь до конфигурационного файла
yandexOptions	Список с необходимыми ключами

5.8.4.9 recv()

```
responseString Jarvis::connection::Transport::recv ( ) const
```

Получение данных

Возвращает

responseString Полученное сообщение

5.8.4.10 send()

```
virtual bool Jarvis::connection::Transport::send ( ) [pure virtual]
```

Отправка данных

Возвращает

bool Подтверждение отправки

Замещается в [Jarvis::connection::SpeechKit](#).

5.8.5 Данные класса

5.8.5.1 _headers

```
headerList* Jarvis::connection::Transport::_headers
```

Заголовки

5.8.5.2 `_properties`

`property Jarvis::connection::Transport::_properties`

Параметры соединения

5.8.5.3 `_response`

`response Jarvis::connection::Transport::_response`

Ответ с сервера

5.8.5.4 `params` [1/2]

`protected Jarvis::connection::Transport::params`

Список параметров

Сокет

5.8.5.5 `params` [2/2]

`private Jarvis::connection::Transport::params`

Список ключевых параметров

Объявления и описания членов класса находятся в файле:

- `/Users/vladislavpereskokov/Desktop/Jarvis/Voice/include/Transport.hpp`

5.9 Класс Jarvis::Voice

Описывает объект голос

```
#include <Voice.hpp>
```

Открытые члены

- `Voice` (`const path &path`, `const name &name`)
Конструктор
- `~Voice` ()=default
Деструктор
- `Voice` (`const Voice ©`)=default
Конструктор копирования
- `Voice` (`Voice &©`)=default
Конструктор копирования
- `Voice & operator=` (`const Voice ©`)=default
Конструктор копирования
- `Voice & operator=` (`Voice &©`)=default
Конструктор копирования
- `bool say` ()
Конструктор копирования
- `void setSentence` (`const Sentence &sentence`)
Метод изменения фразы
- `Sentence getSentence` () const
Метод получения фразы
- `bool execute` ()
Метод запуска воспроизведения голоса
- `bool findCommand` (`const connection::Map &waves`, `const Sentence &sentence`)
Метод нахождения нужной команды

Открытые атрибуты

- `private params`: `name _name`
- `map _waves`
- `Sentence _sentence`

5.9.1 Подробное описание

Описывает объект голос

5.9.2 Конструктор(ы)

5.9.2.1 `Voice()` [1/3]

```
Jarvis::Voice::Voice (
    const path & path,
    const name & name )
```

Конструктор

Аргументы

path	Путь до конфигурационного файла
name	Имя голоса

5.9.2.2 Voice() [2/3]

```
Jarvis::Voice::Voice (  
    const Voice & copy ) [default]
```

Конструктор копирования

Аргументы

copy	Копируемый объект
------	-------------------

5.9.2.3 Voice() [3/3]

```
Jarvis::Voice::Voice (  
    Voice && copy ) [default]
```

Конструктор копирования

Аргументы

copy	Копируемый объект
------	-------------------

5.9.3 Методы

5.9.3.1 execute()

```
bool Jarvis::Voice::execute ( )
```

Метод запуска воспроизведения голоса

Возвращает

bool Проверка - запустилось ли

5.9.3.2 findCommand()

```
bool Jarvis::Voice::findCommand (
    const connection::Map & waves,
    const Sentence & sentence )
```

Метод нахождения нужной команды

Аргументы

waves	Сэмплы
sentence	Фраза

Возвращает

bool Найдено ли

5.9.3.3 getSentence()

```
Sentence Jarvis::Voice::getSentence ( ) const
```

Метод получения фразы

Возвращает

sentence Предложение

5.9.3.4 operator=() [1/2]

```
Voice& Jarvis::Voice::operator= (
    const Voice & copy ) [default]
```

Конструктор копирования

Аргументы

copy	Копируемый объект
------	-------------------

5.9.3.5 operator=() [2/2]

```
Voice& Jarvis::Voice::operator= (
    Voice && copy ) [default]
```

Конструктор копирования

Аргументы

сору	Копируемый объект
------	-------------------

5.9.3.6 say()

```
bool Jarvis::Voice::say ( )
```

Конструктор копирования

Возвращает

bool Подтверждение - была ли сказана фраза

5.9.3.7 setSentence()

```
void Jarvis::Voice::setSentence (
    const Sentence & sentence )
```

Метод изменения фразы

Аргументы

sentence	Предложение
----------	-------------

5.9.4 Данные класса

5.9.4.1 _sentence

[Sentence](#) Jarvis::Voice::_sentence

Фраза

5.9.4.2 _waves

map Jarvis::Voice::_waves

Необходимый сэмпл

5.9.4.3 params

private Jarvis::Voice::params

Имя голоса

Объявления и описания членов класса находятся в файле:

- /Users/vladislavpereskokov/Desktop/Jarvis/Voice/include/Voice.hpp