

client25

Создано системой Doxygen 1.9.1



---

1 Алфавитный указатель классов	1
1.1 Классы	1
2 Классы	3
2.1 Класс console	3
2.1.1 Конструктор(ы)	4
2.1.1.1 console()	4
2.2 Класс Filer	4
2.2.1 Конструктор(ы)	5
2.2.1.1 Filer()	5
2.3 Структура VectData	5
2.4 Класс Web	5
2.4.1 Подробное описание	6
2.4.2 Конструктор(ы)	6
2.4.2.1 Web()	6
2.4.3 Методы	6
2.4.3.1 Receiving()	6
2.4.3.2 Sending()	7
Предметный указатель	9



# Глава 1

## Алфавитный указатель классов

### 1.1 Классы

Классы с их кратким описанием.

<a href="#">console</a>	3
<a href="#">Filer</a>	4
<a href="#">VectData</a>	5
<a href="#">Web</a>	
Класс, обеспечивающий авторизацию и взаимодействие между клиентом и сервером	5



## Глава 2

# Классы

### 2.1 Класс console

#### Открытые члены

- `console ()=delete`  
запрет конструктора без параметров
- `console (int argc, char **argv)`  
Конструктор, внутри которого считываются параметры командной строки
- `string getIP () const`  
Адрес, на котором работает сервер
- `unsigned short int getPort () const`  
Порт, на котором работает сервер
- `string getAuth () const`  
Возвращает путь к файлу для авторизации
- `string getOriginal () const`  
Возвращает путь к файлу с исходными данными
- `string getResult () const`  
Возвращает путь к файлу с результатами
- `bool isIp ()`
- `bool isOriginal ()`
- `bool isResult ()`

#### Открытые атрибуты

- `unsigned short int port =33333`  
Порт, на котором работает сервер
- `string auth ="vclient.conf"`  
Путь к файлу с данными для авторизации
- `string ip`  
Адрес, на котором работает сервер
- `string original`  
Путь к файлу с исходными данными
- `string Result`  
Путь к файлу с результатами

### 2.1.1 Конструктор(ы)

#### 2.1.1.1 console()

```
console::console (
    int argc,
    char ** argv )
```

Конструктор, внутри которого считываются параметры командной строки

Параметры командной строки: -i адрес сервера, обязательный -o Путь к файлу с исходными данными, обязательный -r Путь к файлу с результатами, обязательный -a путь к файлу с данными для авторизации, необязательный -p Порт, на котором работает сервер, необязательный -h вызов подсказки При ошибках в параметрах вызывается справка и программа завершает работу

Аргументы

in	int	argc
in	char	**argv

Объявления и описания членов классов находятся в файлах:

- console.h
- console.cpp

## 2.2 Класс Filer

Открытые члены

- [Filer](#) ()=delete  
запрет конструктора без параметров
- [Filer](#) (const [console](#) &a)  
Конструктор, внутри которого инициализируются параметры из командной строки
- [~Filer](#) ()  
деструктор
- void [Write](#) (uint16\_t piece)  
Запись результатов в файл
- uint32\_t [getVectorSize](#) (unsigned int numVec)  
Размер одного вектора
- uint16\_t \* [getVector](#) (unsigned int numVec)  
Номер вектора
- uint32\_t [getAmount](#) ()  
Логин для авторизации
- string [getLogin](#) () const  
Количество векторов
- string [getPswd](#) () const  
Пароль для авторизации



### 2.2.1 Конструктор(ы)

#### 2.2.1.1 Filer()

```
Filer::Filer (
    const console & a )
```

Конструктор, внутри которого инициализируются параметры из командной строки

Аргументы

in	ifstream	original - Открытие файла с исходными данными для чтения
in	ofstream	Result -Открытие файла с результатами для записи
in	authFileName,путь	к файлу, на котором находятся

Объявления и описания членов классов находятся в файлах:

- operator.h
- operator.cpp

## 2.3 Структура VectData

Открытые атрибуты

- uint32\_t VecSize  
размер вектора
- vector< uint16\_t > \* vec  
вектор со значениями

Объявления и описания членов структуры находятся в файле:

- operator.h

## 2.4 Класс Web

Класс, обеспечивающий авторизацию и взаимодействие между клиентом и сервером

```
#include <connect.h>
```

## Открытые члены

- `Web ()=delete`  
запрет конструктора без параметров
- `Web (const console &a, const Filer &op)`  
Конструктор
- `void Install ()`  
Установка соединения между клиентом и сервером
- `void Auth ()`  
Авторизация клиента на сервере
- `int Receiving (int Socket)`  
Приём данных
- `void Sending (int Socket, char *buf, size_t size)`  
Отправка данных

### 2.4.1 Подробное описание

Класс, обеспечивающий авторизацию и взаимодействие между клиентом и сервером

### 2.4.2 Конструктор(ы)

#### 2.4.2.1 Web()

```
Web::Web (
    const console & a,
    const Filer & op )
```

#### Конструктор

Устанавливает порт IP и авторизационные данные, инициализирует основной сокет и структуры sockaddr\_in

### 2.4.3 Методы

#### 2.4.3.1 Receiving()

```
int Web::Receiving (
    int Socket )
```

#### Приём данных

Аргументы

in	socket,сокет	
----	--------------	--

#### 2.4.3.2 Sending()

```
void Web::Sending (
    int Socket,
    char * buf,
    size_t size )
```

Отправка данных

Аргументы

in	socket,сокет	
in	buf,буфер	с данными
in	size,количество	отправляемых байт

Объявления и описания членов классов находятся в файлах:

- connect.h
- connect.cpp



# Предметный указатель

- console, [3](#)
  - console, [4](#)
- Filer, [4](#)
  - Filer, [5](#)
- Receiving
  - Web, [6](#)
- Sending
  - Web, [7](#)
- VectData, [5](#)
- Web, [5](#)
  - Receiving, [6](#)
  - Sending, [7](#)
  - Web, [6](#)