

Сервер Полехина 22ПТ1

1.0

Создано системой Doxygen 1.9.1

1 Иерархический список классов	2
1.1 Иерархия классов	2
2 Алфавитный указатель классов	2
2.1 Классы	2
3 Список файлов	2
3.1 Файлы	2
4 Классы	3
4.1 Класс Calculator	3
4.1.1 Подробное описание	3
4.1.2 Конструктор(ы)	3
4.2 Класс Client_Communicate	4
4.2.1 Методы	4
4.3 Класс Connector_to_base	5
4.3.1 Подробное описание	6
4.3.2 Методы	6
4.4 Класс crit_err	6
4.4.1 Подробное описание	7
4.4.2 Конструктор(ы)	7
4.5 Класс Interface	8
4.5.1 Подробное описание	8
4.5.2 Методы	8
4.6 Класс Logger	9
4.6.1 Подробное описание	9
4.6.2 Методы	9
4.7 Класс no_crit_err	11
4.7.1 Подробное описание	11
4.7.2 Конструктор(ы)	11
5 Файлы	12
5.1 Файл Calculator.h	12
5.1.1 Подробное описание	12
5.2 Файл Client_Communicate.h	13
5.2.1 Подробное описание	13
5.3 Файл Connector_to_base.h	13
5.3.1 Подробное описание	14
5.4 Файл Errors.h	14
5.4.1 Подробное описание	15
5.5 Файл Interface.h	15
5.5.1 Подробное описание	16
5.6 Файл Logger.h	16
5.6.1 Подробное описание	17

1 Иерархический список классов

1.1 Иерархия классов

Иерархия классов.

Calculator	3
Client_Communicate	4
Connector_to_base	5
Interface	8
Logger	9
std::runtime_error	
crit_err	6
no_crit_err	11

2 Алфавитный указатель классов

2.1 Классы

Классы с их кратким описанием.

Calculator	
Класс для операции умножение вектора	3
Client_Communicate	4
Connector_to_base	
Класс для подключения к базе данных	5
crit_err	
Класс критических ошибок	6
Interface	
Класс для разбора параметров командной строки	8
Logger	
Класс для записи логов	9
no_crit_err	
Класс не критических ошибок	11

3 Список файлов

3.1 Файлы

Полный список документированных файлов.

Calculator.h	Заголовочный файл для модуля вычислений	12
Client_Communicate.h	Заголовочный файл для модуля сетевого взаимодействия	13
Connector_to_base.h	Заголовочный файл для модуля подключения к базе данных	13
Errors.h	Заголовочный файл для модуля ошибок	14
Interface.h	Заголовочный файл для модуля разбора ПКС	15
Logger.h	Заголовочный файл для модуля записи логов	16

4 Классы

4.1 Класс Calculator

Класс для операции умножение вектора

```
#include <Calculator.h>
```

Открытые члены

- [Calculator](#) (std::vector< int16_t >input_data)
Вычисление
- int16_t [send_res](#) ()
Получение результата вычислений

Закрытые данные

- int16_t results

4.1.1 Подробное описание

Класс для операции умножение вектора

Вектор указывается в параметрах конструктора. Для получения результат вычислений используется метод `send_res`.

4.1.2 Конструктор(ы)

4.1.2.1 `Calculator()` `Calculator::Calculator (`
`std::vector< int16_t > input_data)`

Вычисление

Аргументы

in	input_data	Вектор данных. Не должен быть пустой. Тип данных int16_t Исключения не возбуждаются
----	------------	---

Предупреждения

При переполнении будет возвращено максимальное значение int16_t или минимальное значение int16_t

Объявления и описания членов классов находятся в файлах:

- [Calculator.h](#)
- Calculator.cpp

4.2 Класс Client_Communicate

Открытые члены

- int [connection](#) (int port, std::map< std::string, std::string > database, [Logger](#) *l1)
Функция обработки соединения с клиентом

Открытые статические члены

- static std::string [md5](#) (std::string input_str)
Функция хеширования.Алгоритм md5.
- static std::string [generate_salt](#) ()
Функция генерации случайного числа SALT.

4.2.1 Методы

4.2.1.1 `connection()` int Client_Communicate::connection (
int port,
std::map< std::string, std::string > database,
[Logger](#) * l1)

Функция обработки соединения с клиентом

Аргументы

in	port	Порт для работы сервера
in	database	Массив map с базой данных, где ключ - логин
in	l1	Ссылка на класс Logger для записи логов

4.2.1.2 generate_salt() std::string Client_Communicate::generate_salt () [static]

Функция генерации случайного числа SALT.

Возвращает

Случайное число SALT

4.2.1.3 md5() std::string Client_Communicate::md5 (
 std::string input_str) [static]

Функция хеширования.Алгоритм md5.

Аргументы

in	input_str	Строка для хеширования
----	-----------	------------------------

Возвращает

Зашифрованная строка

Объявления и описания членов классов находятся в файлах:

- [Client_Communicate.h](#)
- Client_Communicate.cpp

4.3 Класс Connector_to_base

Класс для подключения к базе данных

```
#include <Connector_to_base.h>
```

Открытые члены

- int [connect_to_base](#) (std::string base_file="/home/stud/base/base.txt")
Функция установки соединения с базой данных
- std::map< std::string, std::string > [get_data](#) ()
Функция получения базы данных в виде массива map.

Закрытые данные

- std::map< std::string, std::string > data_base

4.3.1 Подробное описание

Класс для подключения к базе данных

Чтение базы данных происходит в методе `connect_to_base`. Для получения базы данных в виде массива `map` используется метод `get_data`.

4.3.2 Методы

4.3.2.1 `connect_to_base()` `int Connector_to_base::connect_to_base (`
`std::string base_file = "/home/stud/base/base.txt")`

Функция установки соединения с базой данных

Аргументы

<code>in</code>	<code>base_file</code>	Путь к файлу с базой данных. Значение по умолчанию - "/home/stud/base/base.txt"
-----------------	------------------------	--

Исключения

<code>crit_err</code> ,если	файл с базой данных не прочитался
-----------------------------	-----------------------------------

4.3.2.2 `get_data()` `std::map<std::string,std::string> Connector_to_base::get_data ()` `[inline]`

Функция получения базы данных в виде массива `map`.

Возвращает

`data_base`

Объявления и описания членов классов находятся в файлах:

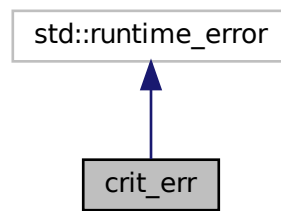
- [Connector_to_base.h](#)
- [Connector_to_base.cpp](#)

4.4 Класс `crit_err`

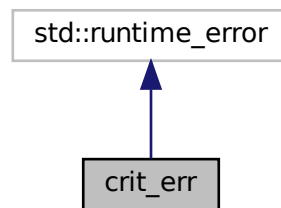
Класс критических ошибок

```
#include <Errors.h>
```

Граф наследования:`crit_err`:



Граф связей класса `crit_err`:



Открытые члены

- `crit_err` (`const std::string &s`)
Конструктор ошибки

4.4.1 Подробное описание

Класс критических ошибок

В конструкторе указывается строка с текстом ошибки

4.4.2 Конструктор(ы)

4.4.2.1 `crit_err()` `crit_err::crit_err (`
`const std::string & s) [inline]`

Конструктор ошибки

Аргументы

in	s	Текст ошибки
----	---	--------------

Объявления и описания членов класса находятся в файле:

- [Errors.h](#)

4.5 Класс Interface

Класс для разбора параметров командной строки

```
#include <Interface.h>
```

Открытые члены

- int [comm_proc](#) (int argc, const char **argv)
Функция для разбора ПКС

Закрытые данные

- int [PORT](#)
Порт работы сервера

4.5.1 Подробное описание

Класс для разбора параметров командной строки

Разбор ПКС происходит в методе `comm_proc`

4.5.2 Методы

4.5.2.1 `comm_proc()` int Interface::comm_proc (
int argc,
const char ** argv)

Функция для разбора ПКС

Аргументы

in	argc	Количество параметров в командной строке
in	argv	Массив с параметрами командной строки

Исключения

boost::program_options::error	
-------------------------------	--

Объявления и описания членов классов находятся в файлах:

- [Interface.h](#)
- [Interface.cpp](#)

4.6 Класс Logger

Класс для записи логов

```
#include <Logger.h>
```

Открытые члены

- `int writelog (std::string s)`
Функция для записи логов
- `int set_path (std::string path_file)`
Функция для установки пути записи логов
- `Logger (std::string s)`

Закрытые статические члены

- `static std::string getCurrentDateTime (std::string s)`
Вспомогательная функция для получения текущего времени

Закрытые данные

- `std::string path_to_logfile`
Атрибут класса, хранящий путь к файлу для записи логов

4.6.1 Подробное описание

Класс для записи логов

Запись логов происходит в методе `writelog`. Для установки пути записи логов используется метод `set_path`

4.6.2 Методы

4.6.2.1 `getCurrentDateTime()` `string Logger::getCurrentDateTime (std::string s) [static], [private]`

Вспомогательная функция для получения текущего времени

Предупреждения

Возвращает

```
4.6.2.2  set_path()  int Logger::set_path (
                                std::string path_file )
```

Аргументы

Исключения

4.6.2.3 writelog() int Logger::writelog (std::string s)

Аргументы

Исключения

Объявления и описания членов классов находятся в файлах:

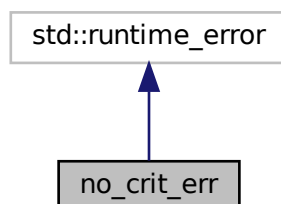
- Создано системой Doxygen

4.7 Класс no_crit_err

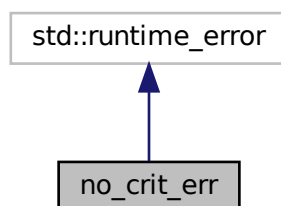
Класс не критических ошибок

```
#include <Errors.h>
```

Граф наследования: no_crit_err:



Граф связей класса no_crit_err:



Открытые члены

- `no_crit_err` (`const std::string s`)
Конструктор ошибки

4.7.1 Подробное описание

Класс не критических ошибок

В конструкторе указывается строка с текстом ошибки

4.7.2 Конструктор(ы)

4.7.2.1 `no_crit_err()` `no_crit_err::no_crit_err (`
`const std::string s) [inline]`

Конструктор ошибки

Аргументы

in	s	Текст ошибки
----	---	--------------

Объявления и описания членов класса находятся в файле:

- [Errors.h](#)

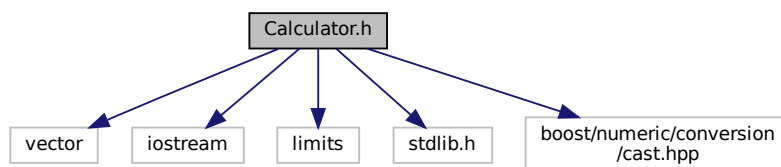
5 Файлы

5.1 Файл Calculator.h

Заголовочный файл для модуля вычислений

```
#include <vector>
#include <iostream>
#include <limits>
#include <stdlib.h>
#include <boost/numeric/conversion/cast.hpp>
```

Граф включаемых заголовочных файлов для Calculator.h:



Классы

- class [Calculator](#)

Класс для операции умножение вектора

5.1.1 Подробное описание

Заголовочный файл для модуля вычислений

Автор

Полехин Д.А.

Версия

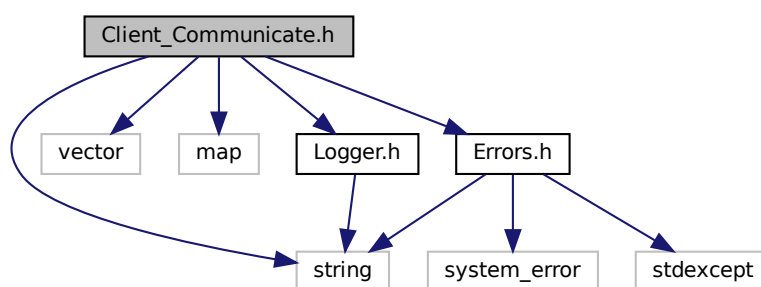
1.0

5.2 Файл Client_Communicate.h

Заголовочный файл для модуля сетевого взаимодействия

```
#include <string>
#include <vector>
#include <map>
#include "Logger.h"
#include "Errors.h"
```

Граф включаемых заголовочных файлов для Client_Communicate.h:



Классы

- class [Client_Communicate](#)

5.2.1 Подробное описание

Заголовочный файл для модуля сетевого взаимодействия

Автор

Полехин Д.А.

Версия

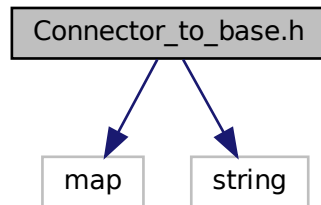
1.0

5.3 Файл Connector_to_base.h

Заголовочный файл для модуля подключения к базе данных

```
#include <map>
#include <string>
```

Граф включаемых заголовочных файлов для Connector_to_base.h:



Классы

- class [Connector_to_base](#)

Класс для подключения к базе данных

5.3.1 Подробное описание

Заголовочный файл для модуля подключения к базе данных

Автор

Полехин Д.А.

Версия

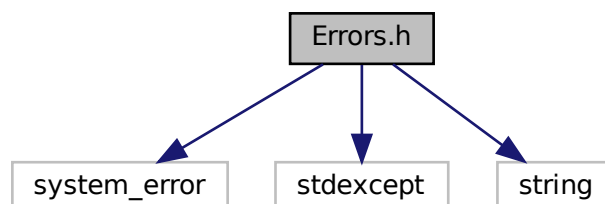
1.0

5.4 Файл Errors.h

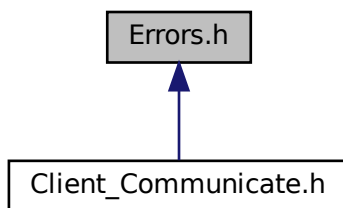
Заголовочный файл для модуля ошибок

```
#include <system_error>
#include <stdexcept>
#include <string>
```

Граф включаемых заголовочных файлов для Errors.h:



Граф файлов, в которые включается этот файл:



Классы

- class `crit_err`
Класс критических ошибок
- class `no_crit_err`
Класс не критических ошибок

5.4.1 Подробное описание

Заголовочный файл для модуля ошибок

Автор

Полехин Д.А.

Версия

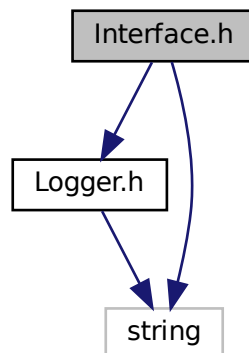
1.0

5.5 Файл Interface.h

Заголовочный файл для модуля разбора ПКС

```
#include "Logger.h"  
#include <string>
```


Граф включаемых заголовочных файлов для Interface.h:



Классы

- class [Interface](#)

Класс для разбора параметров командной строки

5.5.1 Подробное описание

Заголовочный файл для модуля разбора ПКС

Автор

Полексин Д.А.

Версия

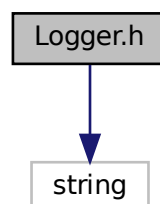
1.0

5.6 Файл Logger.h

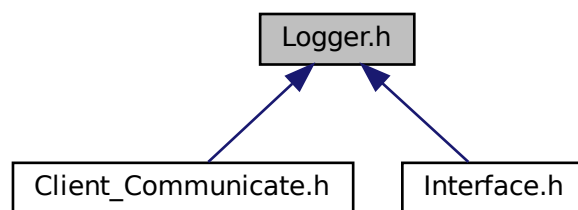
Заголовочный файл для модуля записи логов

```
#include <string>
```

Граф включаемых заголовочных файлов для Logger.h:



Граф файлов, в которые включается этот файл:



Классы

- class [Logger](#)
Класс для записи логов

5.6.1 Подробное описание

Заголовочный файл для модуля записи логов

Автор

Полехин Д.А.

Версия

1.0

Предметный указатель

- Calculator, [3](#)
 - Calculator, [3](#)
- Calculator.h, [12](#)
- Client_Communicate, [4](#)
 - connection, [4](#)
 - generate_salt, [5](#)
 - md5, [5](#)
- Client_Communicate.h, [13](#)
- comm_proc
 - Interface, [8](#)
- connect_to_base
 - Connector_to_base, [6](#)
- connection
 - Client_Communicate, [4](#)
- Connector_to_base, [5](#)
 - connect_to_base, [6](#)
 - get_data, [6](#)
- Connector_to_base.h, [13](#)
- crit_err, [6](#)
 - crit_err, [7](#)
- Errors.h, [14](#)
- generate_salt
 - Client_Communicate, [5](#)
- get_data
 - Connector_to_base, [6](#)
- getCurrentDateTime
 - Logger, [9](#)
- Interface, [8](#)
 - comm_proc, [8](#)
- Interface.h, [15](#)
- Logger, [9](#)
 - getCurrentDateTime, [9](#)
 - set_path, [10](#)
 - writelog, [10](#)
- Logger.h, [16](#)
- md5
 - Client_Communicate, [5](#)
- no_crit_err, [11](#)
 - no_crit_err, [11](#)
- set_path
 - Logger, [10](#)
- writelog
 - Logger, [10](#)