

Курсовая работа

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 AuthError	3
4.1.1 Подробное описание	4
4.2 Структура Calculator	4
4.3 Класс Interface	5
4.3.1 Подробное описание	5
4.3.2 Методы	5
4.4 Класс Listener	6
4.4.1 Подробное описание	7
4.4.2 Конструктор(ы)	7
4.5 Класс Logger	7
4.5.1 Подробное описание	8
4.5.2 Конструктор(ы)	8
4.5.3 Методы	8
4.6 Класс MathError	9
4.6.1 Подробное описание	10
4.7 Класс ServerError	10
4.7.1 Подробное описание	11
4.8 Класс Worker	11
4.8.1 Подробное описание	11
4.8.2 Конструктор(ы)	11
4.8.3 Методы	12
5 Файлы	13
5.1 Файл interface.h	13
5.1.1 Подробное описание	14
5.2 Файл listener.h	14
5.2.1 Подробное описание	15
5.3 Файл logger.h	15
5.3.1 Подробное описание	16
5.4 Файл server_errors.h	16
5.4.1 Подробное описание	17
5.5 Файл worker.h	18
5.5.1 Подробное описание	18
Предметный указатель	19

1 Иерархический список классов

1.1 Иерархия классов

Иерархия классов.

Calculator	4
Interface	5
Listener	6
Logger	7
runtime_error	
ServerError	10
AuthError	3
MathError	9
Worker	11

2 Алфавитный указатель классов

2.1 Классы

Классы с их кратким описанием.

AuthError	
Класс исключений при аутентификации клиента	3
Calculator	4
Interface	
Класс для обработки командной строки	5
Listener	
Класс для соединения с клиентом	6
Logger	
Класс записывает сообщения и время их получения в файл журнала	7
MathError	
Класс исключений при выполнении математических операций сервера	9
ServerError	
Общий класс исключений сервера	10
Worker	
Класс проводит вычисления с векторами	11

3 Список файлов

3.1 Файлы

Полный список документированных файлов.

interface.h	Заголовочный файл определения класса Interface	13
listener.h	Заголовочный файл определения класса Listener	14
logger.h	Заголовочный файл определения класса Logger	15
server_errors.h	Заголовочный файл определения классов исключений	16
worker.h	Заголовочный файл определения класса Worker	18

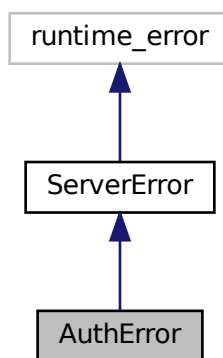
4 Классы

4.1 Класс AuthError

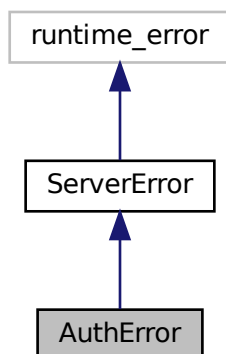
Класс исключений при аутентификации клиента.

```
#include <server_errors.h>
```

Граф наследования:AuthError:



Граф связей класса AuthError:



Открытые члены

- AuthError (const char *msg)

4.1.1 Подробное описание

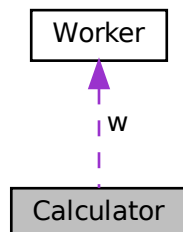
Класс исключений при аутентификации клиента.

Объявления и описания членов класса находятся в файле:

- [server_errors.h](#)

4.2 Структура Calculator

Граф связей класса Calculator:



Открытые атрибуты

- `Worker w {0, map<string,string>{}}`

Объявления и описания членов структуры находятся в файле:

- `tests.cpp`

4.3 Класс Interface

Класс для обработки командной строки.

```
#include <interface.h>
```

Открытые члены

- `Interface ()`
Конструктор класса Создаёт описания параметров.
- `bool start_server (int argc, char *argv[])`
Запустить сервер.

Закрытые данные

- `po::options_description desc`
Описание параметров
- `po::variables_map vm`
Массив для значений параметров
- `string base_path`
Имя файла с базой клиентов
- `string log_path`
Имя файла с журналом
- `int port`
Порт сервера

4.3.1 Подробное описание

Класс для обработки командной строки.

4.3.2 Методы

4.3.2.1 `start_server()` `bool Interface::start_server (`
 `int argc,`
 `char * argv[])`

Запустить сервер.

Аргументы

in	argc	Количество параметров ком.строки.
in	argv	Параметры ком.строки.

Возвращает

true Нет ошибок в параметрах.
false В параметрах обнаружены ошибки.

Объявления и описания членов классов находятся в файлах:

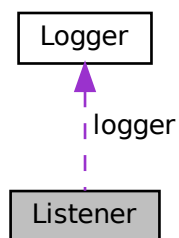
- [interface.h](#)
- [interface.cpp](#)

4.4 Класс Listener

Класс для соединения с клиентом.

```
#include <listener.h>
```

Граф связей класса Listener:



Открытые члены

- [Listener](#) (int port, int queue_length, const map< string, string > &client_base, [Logger](#) log)
Конструктор.
- void [start_listen](#) ()
Установить сервер и запустить прослушивание.

Закрытые данные

- int [server_socket](#)
Сокет
- map< string, string > [base](#)
Массив с данными клиентов <id:pass>
- [Logger](#) [logger](#)
Копия объекта записи в журнал.

4.4.1 Подробное описание

Класс для соединения с клиентом.

4.4.2 Конструктор(ы)

4.4.2.1 `Listener()` `Listener::Listener (`
 `int port,`
 `int queue_length,`
 `const map< string, string > & client_base,`
 `Logger log)`

Конструктор.

Аргументы

in	port	Порт.
in	queue_length	Длина очереди клиентов.
in	client_base	Массив с данными клиентов <id:pass>.
in	log	Объект записи в журнал.

Исключения

<code>ServerError</code>	Ошибка создания сервера. what = "Ошибка создания сокета" "Ошибка настройки сокета" "Ошибка в 'bind' сокета" "Не удалось запустить 'listen'"
--------------------------	--

Объявления и описания членов классов находятся в файлах:

- `listener.h`
- `listener.cpp`

4.5 Класс Logger

Класс записывает сообщения и время их получения в файл журнала.

```
#include <logger.h>
```

Открытые члены

- `Logger` (`const string &log_path`)
 Конструктор с именем файла.
- `Logger` (`Logger &other`)
 Конструктор копирования.
- `bool log` (`const string &message`)
 Записать в журнал.

Закрытые данные

- `string path`
Имя файла журнала

4.5.1 Подробное описание

Класс записывает сообщения и время их получения в файл журнала.

4.5.2 Конструктор(ы)

4.5.2.1 `Logger()` [1/2] `Logger::Logger (`
`const string & log_path)` [inline]

Конструктор с именем файла.

Аргументы

in	log_path	Имя файла журнала.
----	----------	--------------------

4.5.2.2 `Logger()` [2/2] `Logger::Logger (`
`Logger & other)` [inline]

Конструктор копирования.

Аргументы

in	other	Другой объект класса Logger .
----	-------	---

4.5.3 Методы

4.5.3.1 `log()` `bool Logger::log (`
`const string & message)`

Записать в журнал.

Аргументы

in	message	Сообщение для записи.
----	---------	-----------------------

Возвращает

true Сообщение успешно записано в файл журнала.
false Сообщение не удалось записать в файл журнала.

Объявления и описания членов классов находятся в файлах:

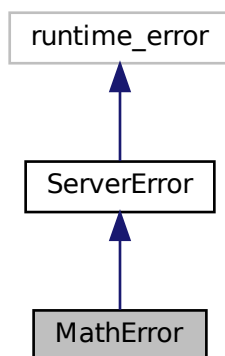
- [logger.h](#)
- [logger.cpp](#)

4.6 Класс MathError

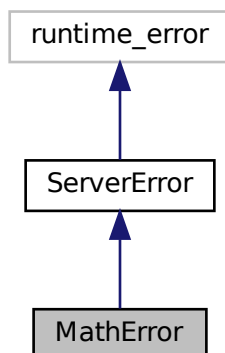
Класс исключений при выполнении математических операций сервера.

```
#include <server_errors.h>
```

Граф наследования:MathError:



Граф связей класса MathError:



Открытые члены

- `MathError (const char *msg)`

4.6.1 Подробное описание

Класс исключений при выполнении математических операций сервера.

Объявления и описания членов класса находятся в файле:

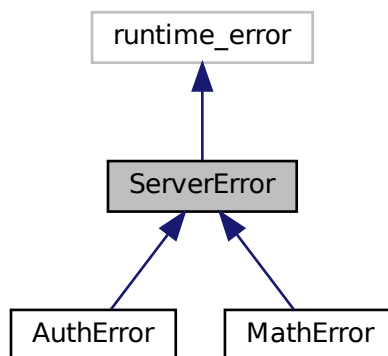
- [server_errors.h](#)

4.7 Класс ServerError

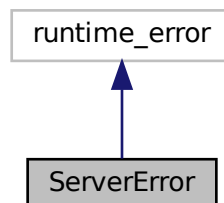
Общий класс исключений сервера.

```
#include <server_errors.h>
```

Граф наследования:ServerError:



Граф связей класса ServerError:



Открытые члены

- `ServerError (const char *msg)`

4.7.1 Подробное описание

Общий класс исключений сервера.

Объявления и описания членов класса находятся в файле:

- [server_errors.h](#)

4.8 Класс Worker

Класс проводит вычисления с векторами

```
#include <worker.h>
```

Открытые члены

- `Worker (int sock, const map< string, string > &client_base)`
Конструктор.
- `void authenticate ()`
Аутентифицировать клиента.
- `void math ()`
Обработать числовые данные клиента.

Закрытые члены

- `string recv_message ()`
Получить сообщение от клиента.
- `int32_t calculate (const vector< int32_t > &v)`
Вычислить среднее арифм. по значениям из вектора.

Закрытые данные

- `int work_socket`
Сокет
- `map< string, string > base`
Массив с данными клиентов <id:pass>

4.8.1 Подробное описание

Класс проводит вычисления с векторами

4.8.2 Конструктор(ы)

4.8.2.1 `Worker()` `Worker::Worker (`
 `int sock,`
 `const map< string, string > & client_base) [inline]`

Конструктор.

Аргументы

in	sock	Сокет.
in	client_base	Массив с данными клиентов <id:pass>.

4.8.3 Методы

4.8.3.1 authenticate() void Worker::authenticate ()

Аутентифицировать клиента.

Исключения

AuthError	Ошибка при аутентификации клиента. what = "Неизвестный id" "Ошибка отправки salt" "Ошибка аутентификации" "Ошибка отправки ОК"
---------------------------	---

4.8.3.2 calculate() int32_t Worker::calculate (
const vector< int32_t > & v) [private]

Вычислить среднее арифм. по значениям из вектора.

Аргументы

in	v	Вектор чисел.
----	---	---------------

Возвращает

int32_t Среднее арифм. значение.

4.8.3.3 math() void Worker::math ()

Обработать числовые данные клиента.

Исключения

MathError	Ошибка при работе с данными клиента. what = "Ошибка получения количества векторов" "Ошибка получения размера вектора" "Ошибка получения вектора" "Неверный размер полученного вектора" "Ошибка отправки результата"
---------------------------	---

4.8.3.4 `recv_message()` `string Worker::recv_message () [private]`

Получить сообщение от клиента.

Исключения

ServerError	Ошибки при получении данных. what = "Ошибка получения строки" "Clear bufer error"
-----------------------------	--

Возвращает

`string` Полученное сообщение.

Объявления и описания членов классов находятся в файлах:

- [worker.h](#)
- `worker.cpp`

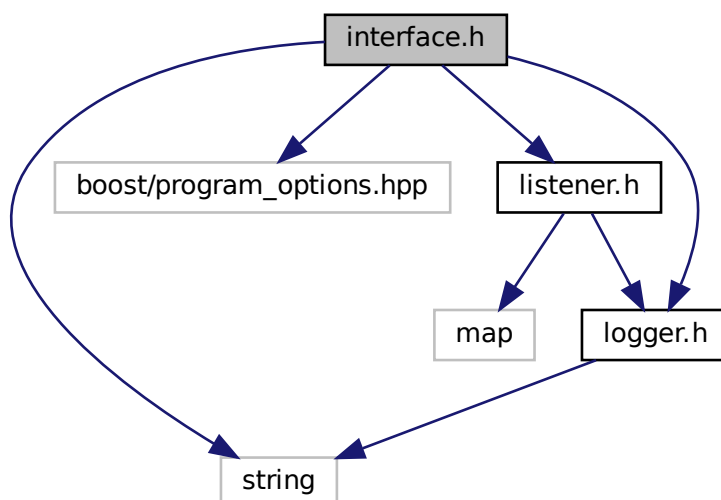
5 Файлы

5.1 Файл `interface.h`

Заголовочный файл определения класса [Interface](#).

```
#include <string>
#include <boost/program_options.hpp>
#include "listener.h"
#include "logger.h"
```

Граф включаемых заголовочных файлов для `interface.h`:



Классы

- class [Interface](#)

Класс для обработки командной строки.

5.1.1 Подробное описание

Заголовочный файл определения класса [Interface](#).

Автор

Kalgudaeva A.E.

Версия

1.0

Дата

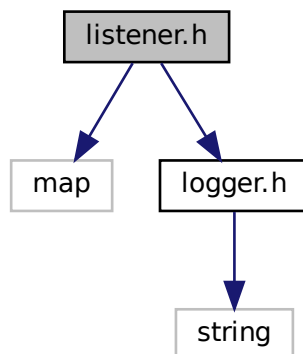
2023-12-27

5.2 Файл listener.h

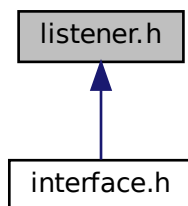
Заголовочный файл определения класса [Listener](#).

```
#include <map>
#include "logger.h"
```

Граф включаемых заголовочных файлов для listener.h:



Граф файлов, в которые включается этот файл:



Классы

- class [Listener](#)

Класс для соединения с клиентом.

5.2.1 Подробное описание

Заголовочный файл определения класса [Listener](#).

Автор

Kalgudaeva A.E.

Версия

1.0

Дата

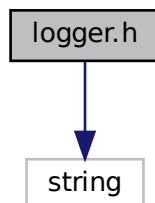
2023-12-27

5.3 Файл logger.h

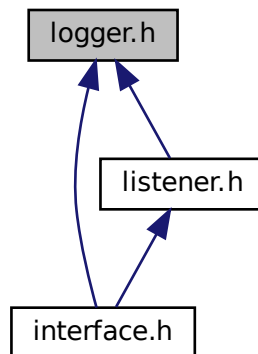
Заголовочный файл определения класса [Logger](#).

```
#include <string>
```

Граф включаемых заголовочных файлов для logger.h:



Граф файлов, в которые включается этот файл:



Классы

- class [Logger](#)

Класс записывает сообщения и время их получения в файл журнала.

5.3.1 Подробное описание

Заголовочный файл определения класса [Logger](#).

Автор

Kalgudaeva A.E.

Версия

1.0

Дата

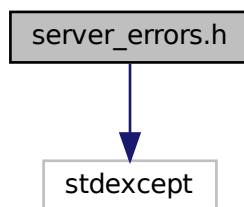
2023-12-27

5.4 Файл `server_errors.h`

Заголовочный файл определения классов исключений

```
#include <stdexcept>
```

Граф включаемых заголовочных файлов для server_errors.h:



Классы

- class [ServerError](#)
Общий класс исключений сервера.
- class [AuthError](#)
Класс исключений при аутентификации клиента.
- class [MathError](#)
Класс исключений при выполнении математических операций сервера.

5.4.1 Подробное описание

Заголовочный файл определения классов исключений

Автор

Kalgudaeva A.E.

Версия

1.0

Дата

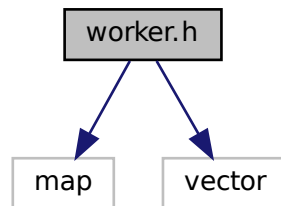
2023-12-27

5.5 Файл worker.h

Заголовочный файл определения класса [Worker](#).

```
#include <map>
#include <vector>
```

Граф включаемых заголовочных файлов для worker.h:



Классы

- class [Worker](#)

Класс проводит вычисления с векторами

5.5.1 Подробное описание

Заголовочный файл определения класса [Worker](#).

Автор

Kalgudaeva A.E.

Версия

1.0

Дата

2023-12-27

Предметный указатель

- authenticate
 - Worker, [12](#)
- AuthError, [3](#)
- calculate
 - Worker, [12](#)
- Calculator, [4](#)
- Interface, [5](#)
 - start_server, [5](#)
- interface.h, [13](#)
- Listener, [6](#)
 - Listener, [7](#)
- listener.h, [14](#)
- log
 - Logger, [8](#)
- Logger, [7](#)
 - log, [8](#)
 - Logger, [8](#)
- logger.h, [15](#)
- math
 - Worker, [12](#)
- MathError, [9](#)
- recv_message
 - Worker, [13](#)
- server_errors.h, [16](#)
- ServerError, [10](#)
- start_server
 - Interface, [5](#)
- Worker, [11](#)
 - authenticate, [12](#)
 - calculate, [12](#)
 - math, [12](#)
 - recv_message, [13](#)
 - Worker, [11](#)
- worker.h, [18](#)