Kursa4

Создано системой Doxygen 1.9.4

Глава 1

Иерархический список классов

1.1 Иерархия классов

Иерархия классов.

Глава 2

Алфавитный указатель классов

2.1 Классы

Классы с их кратким описанием.

Глава 3

Список файлов

3.1 Файлы

Полный список документированных файлов.

Глава 4

Классы

4.1 Класс Calculator

Класс для выполнения вычислений с вектором чисел типа uint16_t.

#include <Calculate.h>

Открытые члены

- Calculator (std::vector< uint16_t > input_data)
  Конструктор, инициализирующий объект Calculator с входными данными.
- uint16_t send_res ()
  Возвращает результат вычислений.

4.1.1 Подробное описание

Класс для выполнения вычислений с вектором чисел типа uint16_t.

4.1.2 Конструктор(ы)

4.1.2.1 Calculator()

Calculator::Calculator (
        std::vector< uint16_t > input_data)

Конструктор, инициализирующий объект Calculator с входными данными.

Аргументы

input_data	Вектор чисел типа uint16_t для обработки.

4.1.3 Методы

4.1.3.1 send_res()

uint16_t Calculator::send_res ( )

Возвращает результат вычислений.

Возвращает

Число типа uint16_t, представляющее результат.

Объявления и описания членов классов находятся в файлах:

- Calculate.h
- Calculate.cpp

4.2 Класс Client_Communicate

Класс для управления сетевым взаимодействием клиента.

#include <Connection_of_client.h>

Открытые члены

- int connection (int port, std::map< std::string, std::string > database, Logger *l1)
  Устанавливает соединение с заданным портом.

Открытые статические члены

- static std::string sha224 (const std::string &input_str)
  Вычисляет SHA-224 хеш строки.

4.2.1 Подробное описание

Класс для управления сетевым взаимодействием клиента.

4.2.2 Методы

4.2.2.1 connection()

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

Устанавливает соединение с заданным портом.

Аргументы

port	Номер порта для подключения.
database	Карта данных (ключ-значение) для использования в коммуникации.
l1	Указатель на объект логгера для отслеживания событий.

Возвращает

Код состояния операции.

4.2.2.2 sha224()

```
std::string Client_Communicate::sha224 (
            const std::string & input_str )  [static]
```

Вычисляет SHA-224 хеш строки.

Аргументы

input_str	Входная строка для хеширования.

Возвращает

Хешированная строка в виде SHA-224.

Объявления и описания членов классов находятся в файлах:

- Connection_of_client.h
- Connection_of_client.cpp

4.3 Класс Connector_to_base

Класс для взаимодействия с базой данных.

#include <DatabaseConnector.h>

Открытые члены

- int connect_to_base (std::string base_file="/etc/vcalc.conf")

  Устанавливает соединение с базой данных, читая из файла конфигурации.
- std::map< std::string, std::string > get_data ()

  Возвращает данные базы.

4.3.1 Подробное описание

Класс для взаимодействия с базой данных.

4.3.2 Методы

4.3.2.1 connect_to_base()

```
int Connector_to_base::connect_to_base (
                std::string base_file = "/etc/vcalc.conf" )
```

Устанавливает соединение с базой данных, читая из файла конфигурации.

Аргументы

base_file	Путь к файлу конфигурации базы данных. По умолчанию "/etc/vcalc.conf".

Возвращает

Код состояния операции соединения.

4.3.2.2 get_data()

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

Возвращает данные базы.

Возвращает

Карту данных базы в формате ключ-значение.

Объявления и описания членов классов находятся в файлах:

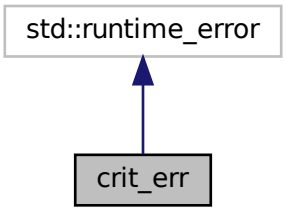
- DatabaseConnector.h
- DatabaseConnector.cpp

4.4 Класс crit_err

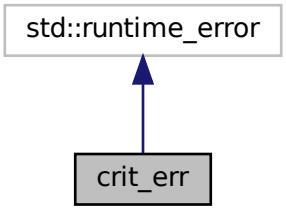
Класс для представления критических ошибок. Наследует от std::runtime_error.

#include <Errors.h>

Граф наследования:crit_err:



Граф связей класса crit_err:



Открытые члены

- crit_err (const std::string &s)
  Конструктор, инициализирующий критическую ошибку с сообщением.

4.4.1 Подробное описание

Класс для представления критических ошибок. Наследует от std::runtime_error.

4.4.2 Конструктор(ы)

4.4.2.1 crit_err()

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

Конструктор, инициализирующий критическую ошибку с сообщением.

Аргументы

s	Сообщение об ошибке.

Объявления и описания членов класса находятся в файле:

- Errors.h

4.5 Класс Interface

Класс для управления интерфейсом связи.

#include <Interface.h>

Открытые члены

- Interface ()
  Конструктор по умолчанию.
- int comm_proc (int argc, const char **argv)
  Обрабатывает командную строку для установления связи.

4.5.1 Подробное описание

Класс для управления интерфейсом связи.

4.5.2 Методы

4.5.2.1 comm_proc()

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

Обрабатывает командную строку для установления связи.

Аргументы

argc	Количество аргументов.
argv	Массив аргументов командной строки.

Возвращает

Код состояния операции.

Объявления и описания членов классов находятся в файлах:

- Interface.h
- Interface.cpp

4.6 Класс Logger

Класс для ведения логирования в файл.

#include <Logger.h>

Открытые члены

- int writelog (std::string s)
  Записывает сообщение в лог-файл.
- int set_path (std::string path_file)
  Устанавливает путь к лог-файлу.
- Logger ()
  Конструктор по умолчанию.
- Logger (std::string s)
  Конструктор с заданным путем к лог-файлу.

4.6.1 Подробное описание

Класс для ведения логирования в файл.

4.6.2 Конструктор(ы)

4.6.2.1 Logger()

```
Logger::Logger (
            std::string s )   [inline]
```

Конструктор с заданным путем к лог-файлу.

Аргументы

s	Путь к файлу лога.

4.6.3 Методы

4.6.3.1 set_path()

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

Устанавливает путь к лог-файлу.

Аргументы

path_file	Путь к файлу лога.

Возвращает

Код состояния операции установки пути.

4.6.3.2 writelog()

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

Записывает сообщение в лог-файл.

Аргументы

s	Сообщение для записи.

Возвращает

Код состояния операции записи.

Объявления и описания членов классов находятся в файлах:

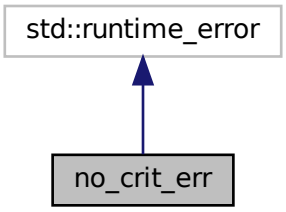
- Logger.h
- Logger.cpp

4.7 Класс no_crit_err

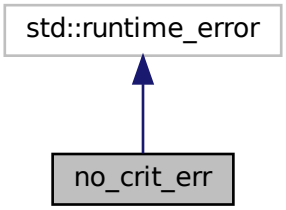
Класс для представления некритических ошибок. Наследует от std::runtime_error.

#include <Errors.h>

Граф наследования:no_crit_err:



Граф связей класса no_crit_err:



Открытые члены

- no_crit_err (const std::string s)
  Конструктор, инициализирующий некритическую ошибку с сообщением.

4.7.1 Подробное описание

Класс для представления некритических ошибок. Наследует от std::runtime_error.

4.7.2 Конструктор(ы)

4.7.2.1 no_crit_err()

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

Конструктор, инициализирующий некритическую ошибку с сообщением.

Аргументы

s	Сообщение об ошибке.

Объявления и описания членов класса находятся в файле:

- Errors.h

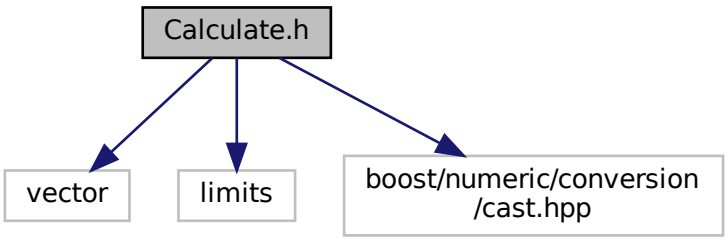
Глава 5

Файлы

5.1 Файл Calculate.h

Объявление класса Calculator для выполнения вычислений с вектором.

#include <vector>
#include <limits>
#include <boost/numeric/conversion/cast.hpp>
Граф включаемых заголовочных файлов для Calculate.h:



Классы

- class Calculator

  Класс для выполнения вычислений с вектором чисел типа uint16_t.

5.1.1 Подробное описание

Объявление класса Calculator для выполнения вычислений с вектором.

Автор

Селуков Е.В.

Версия

1.0

Дата

11.12.2024

Авторство

ИБСТ ПГУ

Предупреждения

Это учебный пример

5.2 Calculate.h

См. документацию.

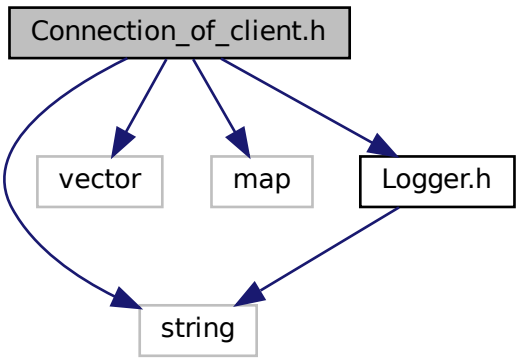
```
1
9 #pragma once
10 #include <vector>
11 #include <limits>
12 #include <boost/numeric/conversion/cast.hpp>
13
16 class Calculator {
17     uint16_t results;
18
19 public:
22     Calculator(std::vector<uint16_t> input_data);
23
26     uint16_t send_res();
27 };
```

5.3 Файл Connection_of_client.h

Объявление класса Client_Communicate для управления сетевым взаимодействием клиента.

#include <string>
#include <vector>
#include <map>

#include "Logger.h"
Граф включаемых заголовочных файлов для Connection_of_client.h:



Классы

- class Client_Communicate

  Класс для управления сетевым взаимодействием клиента.

5.3.1 Подробное описание

Объявление класса Client_Communicate для управления сетевым взаимодействием клиента.

Автор

Селуков Е.В.

Версия

1.0

Дата

11.12.2024

Авторство

ИБСТ ПГУ

Предупреждения

Это учебный пример

5.4 Connection_of_client.h

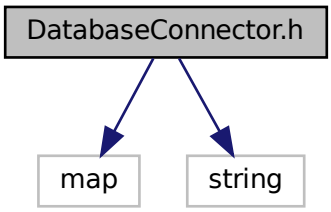
См. документацию.

```
1
9 #pragma once
10 #include <string>
11 #include <vector>
12 #include <map>
13
14 #include "Logger.h"
15
18 class Client_Communicate {
19 public:
25     int connection(int port, std::map<std::string, std::string> database, Logger* l1);
26
30     static std::string sha224(const std::string& input_str);
31 };
```

5.5 Файл DatabaseConnector.h

Объявление класса Connector_to_base для взаимодействия с базой данных.

#include <map>
#include <string>
Граф включаемых заголовочных файлов для DatabaseConnector.h:



Классы

- class Connector_to_base

  Класс для взаимодействия с базой данных.

5.5.1 Подробное описание

Объявление класса Connector_to_base для взаимодействия с базой данных.

Автор

Селуков Е.В.

Версия

1.0

Дата

11.12.2024

Авторство

ИБСТ ПГУ

Предупреждения

Это учебный пример

5.6 DatabaseConnector.h

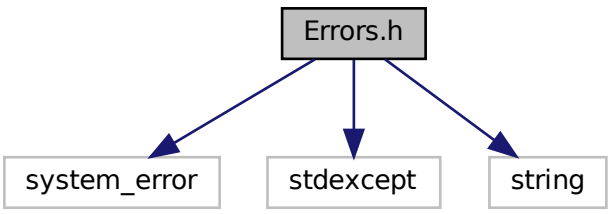
См. документацию.

```
1
9 #pragma once
10 #include <map>
11 #include <string>
12
15 class Connector_to_base {
16 private:
17     std::map<std::string, std::string> data_base;
18
19 public:
23     int connect_to_base(std::string base_file = "/etc/vcalc.conf");
24
27     std::map<std::string, std::string> get_data() {
28         return data_base;
29     }
30 };
```

5.7 Файл Errors.h

Объявление класса crit_err для представления критических ошибок.

#include <system_error>
#include <stdexcept>
#include <string>
Граф включаемых заголовочных файлов для Errors.h:



Классы

- class crit_err

  Класс для представления критических ошибок. Наследует от std::runtime_error.
- class no_crit_err

  Класс для представления некритических ошибок. Наследует от std::runtime_error.

5.7.1 Подробное описание

Объявление класса crit_err для представления критических ошибок.

Автор

Селуков Е.В.

Версия

1.0

Дата

11.12.2024

Авторство

ИБСТ ПГУ

Предупреждения

Это учебный пример

5.8 Errors.h

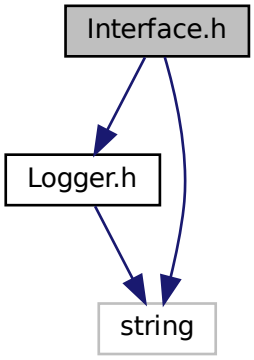
См. документацию.

```
1
9 #pragma once
10 #include <system_error>
11 #include <stdexcept>
12 #include <string>
13
17 class crit_err : public std::runtime_error {
18 public:
21     crit_err(const std::string& s) : std::runtime_error(s) {}
22 };
23
27 class no_crit_err : public std::runtime_error {
28 public:
31     no_crit_err(const std::string s) : std::runtime_error(s) {}
32 };
```

5.9 Файл Interface.h

Объявление класса Interface для управления интерфейсом связи.

#include "Logger.h"
#include <string>
Граф включаемых заголовочных файлов для Interface.h:



Классы

- class Interface

  Класс для управления интерфейсом связи.

5.9.1 Подробное описание

Объявление класса Interface для управления интерфейсом связи.

Автор

Селуков Е.В.

Версия

1.0

Дата

11.12.2024

Авторство

ИБСТ ПГУ

Предупреждения

Это учебный пример

5.10 Interface.h

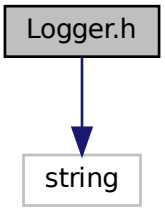
См. документацию.

```
1
9 #pragma once
10 #include "Logger.h"
11 #include <string>
14 class Interface {
15     int PORT;
16 public:
18     Interface() {}
23     int comm_proc(int argc, const char** argv);
24 };
```

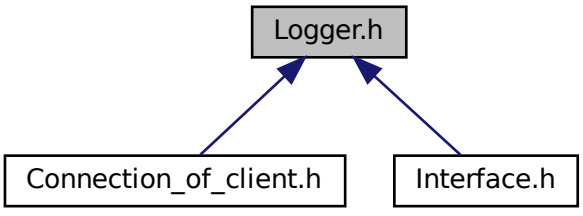
5.11 Файл Logger.h

Объявление класса Logger для ведения логирования в файл.

#include <string>
Граф включаемых заголовочных файлов для Logger.h:



Граф файлов, в которые включается этот файл:



Классы

- class Logger

  Класс для ведения логирования в файл.

5.11.1 Подробное описание

Объявление класса Logger для ведения логирования в файл.

Автор

Селуков Е.В.

Версия

1.0

Дата

11.12.2024

Авторство

ИБСТ ПГУ

Предупреждения

Это учебный пример

5.12 Logger.h

См. документацию.

```
1
9 #pragma once
10 #include <string>
13 class Logger {
17     static std::string getCurrentDateTime(std::string s);
18
19     std::string path_to_logfile;
20
21 public:
25     int writelog(std::string s);
26
30     int set_path(std::string path_file);
31
33     Logger() { path_to_logfile = " "; };
34
37     Logger(std::string s) { path_to_logfile = s; };
38 };
```

Предметный указатель