Удалов В.В. 23ПТ1

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

Выполняет вычисления на векторе целых чисел.

#include <Calculator.h>

Открытые члены

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

4.1.1 Подробное описание

Выполняет вычисления на векторе целых чисел.

Обрабатывает массив целых чисел и вычисляет произведение.

4.1.2 Конструктор(ы)

4.1.2.1 Calculator()

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

Конструктор, инициализирующий вычисления с входными данными.

Аргументы

input_data	Вектор целых чисел для вычисления.

4.1.3 Методы

4.1.3.1 send_res()

uint16_t Calculator::send_res ( )

Возвращает результат вычисления.

Возвращает

uint16_t Результат вычисления.

Объявления и описания членов классов находятся в файлах:

- Calculator.h
- Calculator.cpp

4.2 Класс Communicate

Обрабатывает обмен данными между сервером и клиентом.

#include <Communicate.h>

Открытые члены

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

Открытые статические члены

- static std::string sha256 (std::string input_str)
  Вычисляет SHA256-хэш строки.
- static std::string generate_salt ()
  Генерирует случайный соль.

4.2.1 Подробное описание

Обрабатывает обмен данными между сервером и клиентом.

4.2.2 Методы

4.2.2.1 connection()

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

Устанавливает соединение с клиентом.

Аргументы

port	Порт сервера.
database	База данных с учетными данными пользователей.
l1	Экземпляр Logger для ведения журнала.

Возвращает

int Код состояния.

4.2.2.2 generate_salt()

```
std::string Communicate::generate_salt ( )  [static]
```

Генерирует случайный соль.

Возвращает

std::string Сгенерированная соль в шестнадцатеричном формате.

4.2.2.3 sha256()

```
std::string Communicate::sha256 (
                std::string input_str )  [static]
```

Вычисляет SHA256-хэш строки.

Аргументы

input_str	Входная строка.

Возвращает

std::string SHA256-хэш входной строки.

Объявления и описания членов классов находятся в файлах:

- Communicate.h
- Communicate.cpp

4.3 Класс Connector

Управляет подключениями к базе данных.

#include <Connector.h>

Открытые члены

- int connect (std::string base_file="/home/stud/Server/test_files/auth.txt")
  Подключается к базе данных.
- std::map< std::string, std::string > get_data ()
  Получает данные из базы данных.

4.3.1 Подробное описание

Управляет подключениями к базе данных.

Загружает данные пользователей из файла и хранит их в словаре.

4.3.2 Методы

4.3.2.1 connect()

```
int Connector::connect (
            std::string base_file = "/home/stud/Server/test_files/auth.txt" )
```

Подключается к базе данных.

Аргументы

base_file	Путь к файлу базы данных.

Возвращает

int Код состояния.

4.3.2.2 get_data()

std::map< std::string, std::string > Connector::get_data ( )

Получает данные из базы данных.

Возвращает

std::map<std::string, std::string> Словарь записей в базе данных.

Объявления и описания членов классов находятся в файлах:

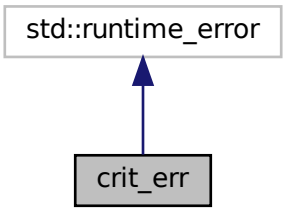
- Connector.h
- Connector.cpp

4.4 Класс crit_err

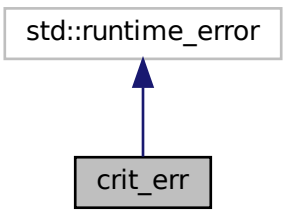
Представляет критическую ошибку.

#include <Errors.h>

Граф наследования:crit_err:



Граф связей класса crit_err:



Открытые члены

- crit_err (const std::string &s)
  Конструктор для crit_err.

4.4.1 Подробное описание

Представляет критическую ошибку.

4.4.2 Конструктор(ы)

4.4.2.1 crit_err()

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

Конструктор для crit_err.

Аргументы

s	Описание ошибки.

Объявления и описания членов класса находятся в файле:

- Errors.h

4.5 Класс Interface

Управляет процессом обмена сообщениями с клиентами.

#include <Interface.h>

Открытые члены

- Interface ()
  Конструктор.
- int comm_proc (int argc, const char **argv)
  Обрабатывает аргументы командной строки и инициализирует обмен сообщениями.

4.5.1 Подробное описание

Управляет процессом обмена сообщениями с клиентами.

Обрабатывает аргументы командной строки и настраивает логирование.

4.5.2 Методы

4.5.2.1 comm_proc()

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

Обрабатывает аргументы командной строки и инициализирует обмен сообщениями.

Аргументы

argc	Количество аргументов командной строки.
argv	Массив аргументов командной строки.

Возвращает

int Код состояния.

Объявления и описания членов классов находятся в файлах:

- Interface.h
- Interface.cpp

4.6 Класс Logger

Класс для обработки записи логов.

#include <Logger.h>

Открытые члены

- int writelog (std::string s)
  Записывает строку в лог.
- int set_path (std::string path_file)
  Устанавливает путь к файлу лога.
- Logger ()
  Конструктор по умолчанию.
- Logger (std::string s)
  Конструктор с указанием пути к файлу.

4.6.1 Подробное описание

Класс для обработки записи логов.

Позволяет записывать и управлять файлами логов.

4.6.2 Конструктор(ы)

4.6.2.1 Logger()

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

Конструктор с указанием пути к 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 файлу лога.

Аргументы

path_file	Путь к файлу лога.

Возвращает

int Код состояния.

4.6.3.2 writelog()

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

Записывает строку в лог.

Аргументы

s	Строка для записи.

Возвращает

int Код состояния.

Объявления и описания членов классов находятся в файлах:

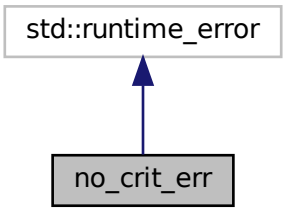
- Logger.h
- Logger.cpp

4.7 Класс no_crit_err

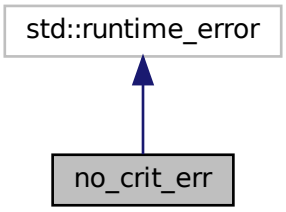
Представляет некритическую ошибку.

#include <Errors.h>

Граф наследования:no_crit_err:



Граф связей класса no_crit_err:



Открытые члены

- no_crit_err (const std::string s)
  Конструктор для no_crit_err.

4.7.1 Подробное описание

Представляет некритическую ошибку.

4.7.2 Конструктор(ы)

4.7.2.1 no_crit_err()

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

Конструктор для no_crit_err.

Аргументы

s	Описание ошибки.

Объявления и описания членов класса находятся в файле:

- Errors.h

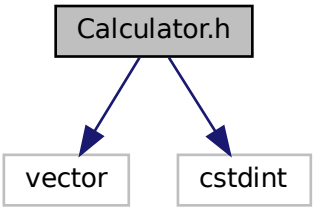
Глава 5

Файлы

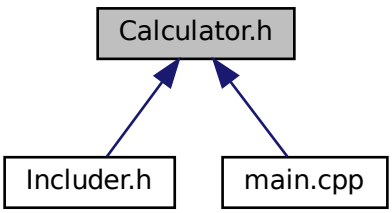
5.1 Файл Calculator.h

Объявление класса Calculator для векторных вычислений.

#include <vector>
#include <cstdint>
Граф включаемых заголовочных файлов для Calculator.h:



Граф файлов, в которые включается этот файл:



Классы

- class Calculator

  Выполняет вычисления на векторе целых чисел.

5.1.1 Подробное описание

Объявление класса Calculator для векторных вычислений.

Версия

1.0

Дата

20.11.2024

Предупреждения

Это учебный пример

5.2 Calculator.h

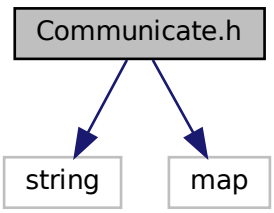
См. документацию.

```
1
8 #pragma once
9 #include <vector>
10 #include <cstdint>
11
15 class Calculator {
16     uint16_t results;
18 public:
23     Calculator(std::vector<uint16_t> input_data);
24
29     uint16_t send_res();
30 };
```

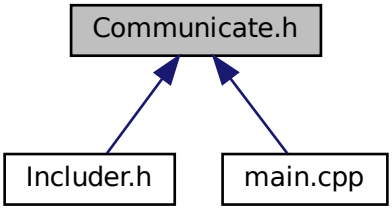
5.3 Файл Communicate.h

Объявление класса Communicate для серверного обмена данными.

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



Граф файлов, в которые включается этот файл:



Классы

- class Communicate

  Обрабатывает обмен данными между сервером и клиентом.

5.3.1 Подробное описание

Объявление класса Communicate для серверного обмена данными.

Версия

1.0

Дата

20.11.2024

Предупреждения

Это учебный пример

5.4 Communicate.h

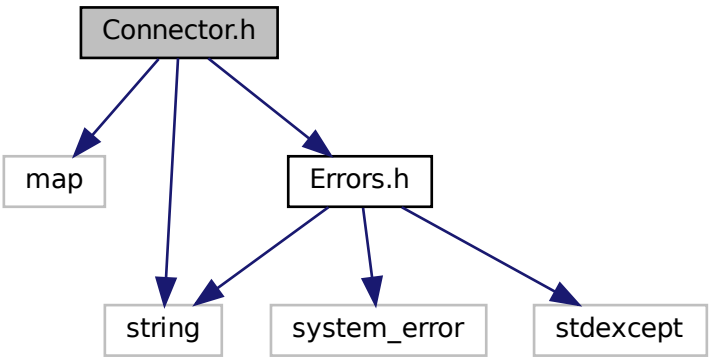
См. документацию.

```
1
8 #pragma once
9 #include <string>
10 #include <map>
11
12 class Logger;
13
16 class Communicate {
17 public:
25     int connection(int port, std::map<std::string, std::string> database, Logger* l1);
26
32     static std::string sha256(std::string input_str);
33
38     static std::string generate_salt();
39 };
```

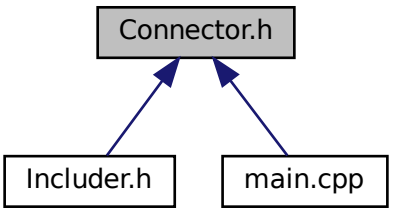
5.5 Файл Connector.h

Объявление класса Connector для управления подключениями к базе данных.

#include <map>
#include <string>
#include "Errors.h"
Граф включаемых заголовочных файлов для Connector.h:



Граф файлов, в которые включается этот файл:



Классы

- class Connector

  Управляет подключениями к базе данных.

5.5.1 Подробное описание

Объявление класса Connector для управления подключениями к базе данных.

Версия

1.0

Дата

20.11.2024

Предупреждения

Это учебный пример

5.6 Connector.h

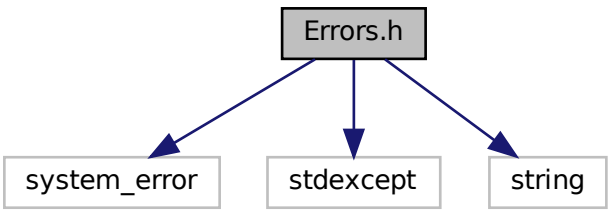
См. документацию.

```
1
8 #pragma once
9 #include <map>
10 #include <string>
11 #include "Errors.h"
12
16 class Connector {
17 private:
18     std::map<std::string, std::string> data_base;
20 public:
26     int connect(std::string base_file = "/home/stud/Server/test_files/auth.txt");
27
32     std::map<std::string, std::string> get_data();
33 };
```

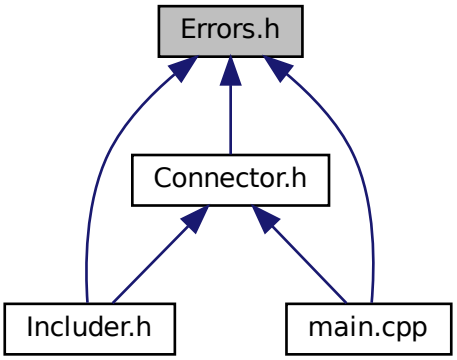
5.7 Файл Errors.h

Определяет классы исключений для обработки 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
  Представляет критическую ошибку.
- class no_crit_err
  Представляет некритическую ошибку.

5.7.1 Подробное описание

Определяет классы исключений для обработки ошибок.

Версия

1.0

Дата

20.11.2024

Предупреждения

Это учебный пример

5.8 Errors.h

См. документацию.

```
1
8 #pragma once
9 #include <system_error>
10 #include <stdexcept>
11 #include <string>
12
15 class crit_err: public std::runtime_error {
16 public:
21     crit_err(const std::string& s): std::runtime_error(s) {}
22 };
23
26 class no_crit_err: public std::runtime_error {
27 public:
32     no_crit_err(const std::string s): std::runtime_error(s) {}
33 };
```

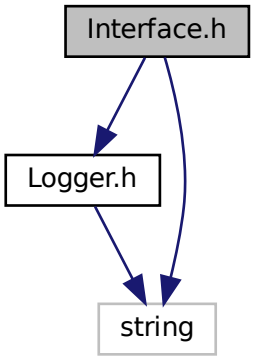
5.9 Includer.h

```
1 #include "Connector.h"
2 #include "Communicate.h"
3 #include "Calculator.h"
4 #include "Interface.h"
5 #include "Logger.h"
6 #include "Errors.h"
```

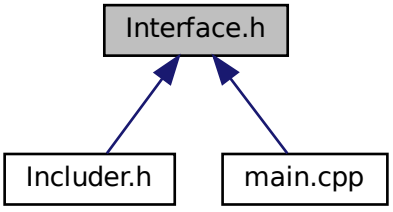
5.10 Файл Interface.h

Объявление класса Interface для обработки взаимодействий с пользователем.

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



Граф файлов, в которые включается этот файл:



Классы

- class Interface

  Управляет процессом обмена сообщениями с клиентами.

5.10.1 Подробное описание

Объявление класса Interface для обработки взаимодействий с пользователем.

Версия

1.0

Дата

20.11.2024

Предупреждения

Это учебный пример

5.11 Interface.h

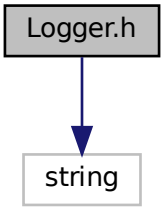
См. документацию.

```
1
8 #pragma once
9 #include "Logger.h"
10 #include <string>
11
15 class Interface {
16     int PORT;
18 public:
22     Interface() {}
23
30     int comm_proc(int argc, const char** argv);
31 };
```

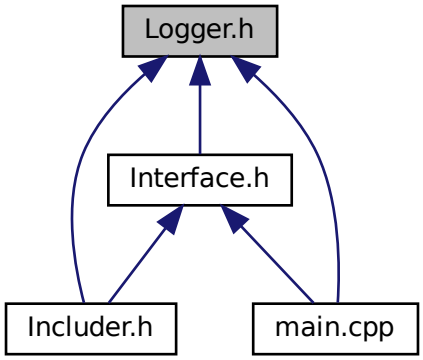
5.12 Файл Logger.h

Объявление класса Logger для ведения логов.

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



Граф файлов, в которые включается этот файл:



Классы

- class Logger

  Класс для обработки записи логов.

5.12.1 Подробное описание

Объявление класса Logger для ведения логов.

Автор

Удалов В.В.

Версия

1.0

Дата

20.11.2024

Авторство

ИБСТ ПГУ

Предупреждения

Это учебный пример

5.13 Logger.h

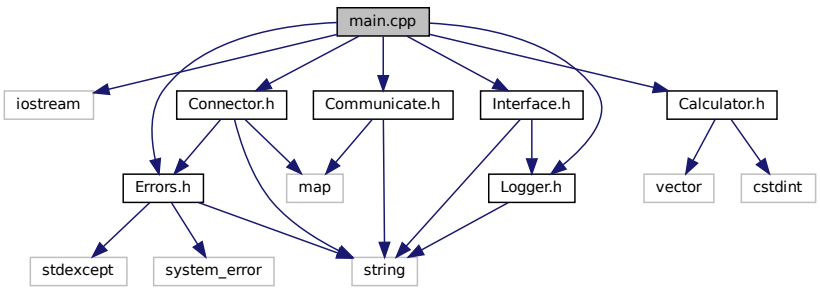
См. документацию.

```
1
10 #pragma once
11 #include <string>
12
16 class Logger {
22     static std::string getCurrentDateTime(std::string s);
23
24     std::string path_to_logfile;
26 public:
32     int writelog(std::string s);
33
39     int set_path(std::string path_file);
40
44     Logger();
45
50     Logger(std::string s);
51 };
```

5.14 Файл main.cpp

Главная точка входа для приложения.

#include <iostream>
#include "Connector.h"
#include "Interface.h"
#include "Communicate.h"
#include "Calculator.h"
#include "Errors.h"
#include "Logger.h"
Граф включаемых заголовочных файлов для main.cpp:



Функции

- int main (int argc, const char **argv)

  Главная функция для инициализации интерфейса и запуска процесса обмена сообщениями.

5.14.1 Подробное описание

Главная точка входа для приложения.

Автор

Удалов В.В.

Версия

1.0

Дата

20.11.2024

Авторство

ИБСТ ПГУ

Предупреждения

Это учебный пример

5.14.2 Функции

5.14.2.1 main()

```
int main (
                int argc,
                const char ** argv )
```

Главная функция для инициализации интерфейса и запуска процесса обмена сообщениями.

Аргументы

argc	Количество аргументов командной строки.
argv	Массив аргументов командной строки.

Возвращает

int Код завершения программы.

Предметный указатель