

NIKITA KARGIN

Создано системой Doxygen 1.9.1

1 Иерархический список классов	1
1.1 Иерархия классов	1
2 Алфавитный указатель классов	3
2.1 Классы	3
3 Список файлов	5
3.1 Файлы	5
4 Классы	7
4.1 Класс Calculator	7
4.1.1 Подробное описание	7
4.1.2 Методы	7
4.1.2.1 math()	7
4.2 Класс ClientBase	8
4.2.1 Подробное описание	8
4.2.2 Методы	8
4.2.2.1 find()	8
4.2.2.2 read()	9
4.3 Класс Communicator	9
4.3.1 Подробное описание	10
4.3.2 Методы	10
4.3.2.1 getClientSocket()	10
4.3.2.2 Initialize()	10
4.3.2.3 WaitClient()	10
4.4 Класс Logs	11
4.4.1 Подробное описание	11
4.4.2 Конструктор(ы)	11
4.4.2.1 Logs()	11
4.4.3 Методы	12
4.4.3.1 Error()	12
4.4.3.2 Info()	12
4.4.3.3 Warning()	12
4.5 Класс ServerError	13
4.5.1 Подробное описание	14
4.5.2 Конструктор(ы)	14
4.5.2.1 ServerError()	14
4.6 Класс UserInterface	14
4.6.1 Подробное описание	15
4.6.2 Конструктор(ы)	15
4.6.2.1 UserInterface()	15
5 Файлы	17
5.1 Файл Calculator.h	17
5.1.1 Подробное описание	17

5.2 Файл ClientBase.h	18
5.2.1 Подробное описание	19
5.3 Файл Communicator.h	19
5.3.1 Подробное описание	20
5.4 Файл Logs.h	20
5.4.1 Подробное описание	21
5.5 Файл ServerError.h	22
5.5.1 Подробное описание	22
5.6 Файл UserInterface.h	23
5.6.1 Подробное описание	23
Предметный указатель	25

Глава 1

Иерархический список классов

1.1 Иерархия классов

Иерархия классов.

Calculator	7
ClientBase	8
Comunicator	9
Logs	11
std::runtime_error	
ServerError	13
UserInterface	14

Глава 2

Алфавитный указатель классов

2.1 Классы

Классы с их кратким описанием.

Calculator		
Класс Calculator	7
ClientBase		
Класс ClientBase	8
Communicator		
Класс Communicator	9
Logs		
Класс Logs	11
ServerError		
Класс ServerError	13
UserInterface		
Класс UsersInterface	14

Глава 3

Список файлов

3.1 Файлы

Полный список документированных файлов.

Calculator.h	
Заголовочный файл для модуля Calculator	17
ClientBase.h	
Заголовочный файл для модуля ClientBase	18
Communicator.h	
Заголовочный файл для модуля Communicator	19
Logs.h	
Заголовочный файл для модуля Logs	20
ServerError.h	
Заголовочный файл для модуля ServerError	22
UserInterface.h	
Заголовочный файл для модуля UsersInterface	23

Глава 4

Классы

4.1 Класс Calculator

Класс [Calculator](#).

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

Открытые статические члены

- static void [math](#) (int work_sock)
Статический метод для выполнения математических операций

4.1.1 Подробное описание

Класс [Calculator](#).

Класс [Calculator](#) считает математические операции.

Предупреждения

Реализация только для типа данных uint16_t

4.1.2 Методы

4.1.2.1 math()

```
static void Calculator::math (  
    int work_sock ) [static]
```

Статический метод для выполнения математических операций

Аргументы

work_sock	рабочий сокет для обработки операций
-----------	--------------------------------------

Объявления и описания членов класса находятся в файле:

- [Calculator.h](#)

4.2 Класс ClientBase

Класс [ClientBase](#).

```
#include <ClientBase.h>
```

Открытые статические члены

- static std::map< std::string, std::string > [read](#) (std::string pathToFile)
Считывает информацию о клиенте из файла и возвращает ее в виде карты.
- static std::string [find](#) (std::string &name, std::map< std::string, std::string > &all)
Находит сведения о клиенте на заданной карте, используя заданное имя клиента.

4.2.1 Подробное описание

Класс [ClientBase](#).

Класс [ClientBase](#) считывает базу данных с клиентами.

Предупреждения

Реализация только для типа данных uint16_t

4.2.2 Методы

4.2.2.1 find()

```
static std::string ClientBase::find (
    std::string & name,
    std::map< std::string, std::string > & all ) [static]
```

Находит сведения о клиенте на заданной карте, используя заданное имя клиента.

Находит информацию о клиенте на данной карте, выполняя поиск по имени клиента, указанному в заданном параметре. Имя клиента передается по ссылке, чтобы его можно было изменить на фактическое найденное имя, если оно есть.

Аргументы

name	Имя искомого клиента.
all	Карта с информацией о клиенте для поиска.

Возвращает

Сведения о клиенте, если совпадение найдено, или пустая строка, если совпадение не найдено.

4.2.2.2 read()

```
static std::map<std::string, std::string> ClientBase::read (
    std::string pathToFile ) [static]
```

Считывает информацию о клиенте из файла и возвращает ее в виде карты.

Считывает информацию о клиенте из файла, указанного по заданному пути, и возвращает ее в виде карты, где ключами являются имена клиентов, а значениями — сведения о клиенте.

Аргументы

pathToFile	Путь к файлу, содержащему информацию о клиенте.
------------	---

Возвращает

Карта с информацией о клиенте, где ключи — это имена клиентов, а значения — сведения о клиенте.

Объявления и описания членов класса находятся в файле:

- [ClientBase.h](#)

4.3 Класс Communicator

Класс [Communicator](#).

```
#include <Communicator.h>
```

Открытые члены

- void [Initialize](#) (int port)
Инициализирует коммуникатор
- int [WaitClient](#) (std::map< std::string, std::string > &all_names)
Ожидает клиента и получает все имена
- int [getClientSocket](#) ()
Возвращает сокет клиента

4.3.1 Подробное описание

Класс `Communicator`.

Класс `Communicator` отвечает за связь между клиентом и сервером.

Предупреждения

Реализация только для типа данных `uint16_t`

4.3.2 Методы

4.3.2.1 `getClientSocket()`

```
int Communicator::getClientSocket ( ) [inline]
```

Возвращает сокет клиента

Возвращает

сокет клиента

4.3.2.2 `Initialize()`

```
void Communicator::Initialize (  
    int port )
```

Инициализирует коммуникатор

Аргументы

<code>port</code>	- порт для подключения
-------------------	------------------------

4.3.2.3 `WaitClient()`

```
int Communicator::WaitClient (  
    std::map< std::string, std::string > & all_names )
```

Ожидает клиента и получает все имена

Аргументы

all_names	- словарь со всеми именами
-----------	----------------------------

Возвращает

возвращает код результата

Объявления и описания членов класса находятся в файле:

- [Communicator.h](#)

4.4 Класс Logs

Класс [Logs](#).

```
#include <Logs.h>
```

Открытые члены

- [Logs](#) ()
Конструктор по умолчанию для класса [Logs](#).
- [Logs](#) (std::string pathToFile)
Конструктор для класса [Logs](#).
- void [Info](#) (std::string str)
Регистрирует данное информационное сообщение.
- void [Warning](#) (std::string str)
Регистрирует данное предупреждающее сообщение.
- void [Error](#) (std::string str)
Регистрирует данное сообщение об ошибке

4.4.1 Подробное описание

Класс [Logs](#).

Класс [Logs](#) отвечает за регистрацию информации и ошибок сервера.

Предупреждения

Реализация только для типа данных uint16_t

4.4.2 Конструктор(ы)

4.4.2.1 Logs()

```
Logs::Logs (  
    std::string pathToFile )
```

Конструктор для класса [Logs](#).

Аргументы

pathToFile	Путь к файлу журнала.
------------	-----------------------

4.4.3 Методы

4.4.3.1 Error()

```
void Logs::Error (  
    std::string str )
```

Регистрирует данное сообщение об ошибке

Аргументы

str	Сообщение об ошибке, которое необходимо записать в журнал.
-----	--

4.4.3.2 Info()

```
void Logs::Info (  
    std::string str )
```

Регистрирует данное информационное сообщение.

Аргументы

str	Информационное сообщение, которое необходимо записать в журнал.
-----	---

4.4.3.3 Warning()

```
void Logs::Warning (  
    std::string str )
```

Регистрирует данное предупреждающее сообщение.

Аргументы

str	Предупреждающее сообщение, которое необходимо записать в журнал.
-----	--

Объявления и описания членов класса находятся в файле:

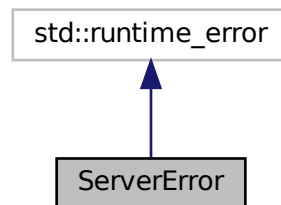
- [Logs.h](#)

4.5 Класс `ServerError`

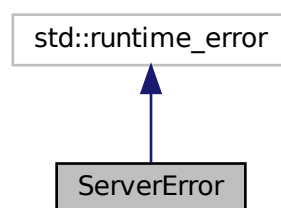
Класс [ServerError](#).

```
#include <ServerError.h>
```

Граф наследования: `ServerError`:



Граф связей класса `ServerError`:



Открытые члены

- [ServerError](#) (`const char *msg`)
Конструктор класса [ServerError](#).

4.5.1 Подробное описание

Класс [ServerError](#).

Класс [ServerError](#) отвечает за обработку ошибок сервера.

Предупреждения

Реализация только для типа данных `uint16_t`

4.5.2 Конструктор(ы)

4.5.2.1 ServerError()

```
ServerError::ServerError (
    const char * msg ) [inline]
```

Конструктор класса [ServerError](#).

Аргументы

msg	- сообщение об ошибке
-----	-----------------------

Объявления и описания членов класса находятся в файле:

- [ServerError.h](#)

4.6 Класс UserInterface

Класс `UsersInterface`.

```
#include <UserInterface.h>
```

Открытые члены

- [UserInterface](#) ()
Конструктор по умолчанию для класса [UserInterface](#).
- [UserInterface](#) (int argc, char **argv)
Конструктор для класса [UserInterface](#).
- void [operator](#)() ()
Перегруженный оператор вызова функции. Эта функция является точкой входа в класс [UserInterface](#).

4.6.1 Подробное описание

Класс `UsersInterface`.

Класс `UsersInterface` отвечает за инициализацию и запуск сервера.

Предупреждения

Реализация только для типа данных `uint16_t`

4.6.2 Конструктор(ы)

4.6.2.1 `UserInterface()`

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

Конструктор для класса [UserInterface](#).

Аргументы

<code>argc</code>	Количество аргументов командной строки.
<code>argv</code>	Аргументы командной строки.

Объявления и описания членов класса находятся в файле:

- [UserInterface.h](#)

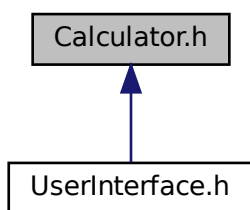
Глава 5

Файлы

5.1 Файл Calculator.h

Заголовочный файл для модуля [Calculator](#).

Граф файлов, в которые включается этот файл:



Классы

- class [Calculator](#)
Класс [Calculator](#).

5.1.1 Подробное описание

Заголовочный файл для модуля [Calculator](#).

Автор

Каргин Н.С.

Версия

1.0

Дата

28.12.2023

Авторство

ИБСТ ПГУ

Предупреждения

Курсовая работа студента

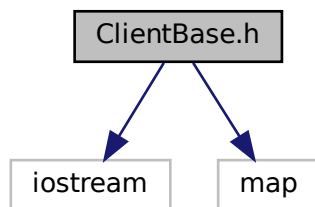
5.2 Файл ClientBase.h

Заголовочный файл для модуля [ClientBase](#).

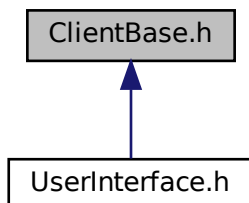
```
#include <iostream>
```

```
#include <map>
```

Граф включаемых заголовочных файлов для ClientBase.h:



Граф файлов, в которые включается этот файл:



Классы

- class [ClientBase](#)
Класс [ClientBase](#).

5.2.1 Подробное описание

Заголовочный файл для модуля [ClientBase](#).

Автор

Каргин Н.С.

Версия

1.0

Дата

28.12.2023

Авторство

ИБСТ ПГУ

Предупреждения

Курсовая работа студента

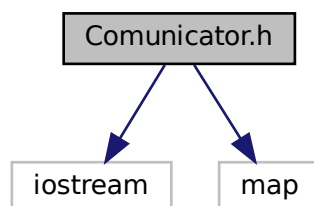
5.3 Файл Communicator.h

Заголовочный файл для модуля [Communicator](#).

```
#include <iostream>
```

```
#include <map>
```

Граф включаемых заголовочных файлов для Communicator.h:



Классы

- class [Communicator](#)
Класс [Communicator](#).

5.3.1 Подробное описание

Заголовочный файл для модуля [Communicator](#).

Автор

Каргин Н.С.

Версия

1.0

Дата

28.12.2023

Авторство

ИБСТ ПГУ

Предупреждения

Курсовая работа студента

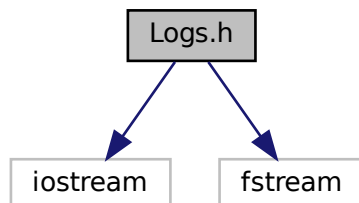
5.4 Файл Logs.h

Заголовочный файл для модуля [Logs](#).

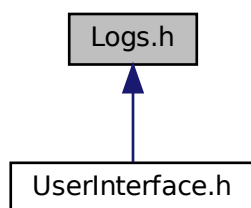
```
#include <iostream>
```

```
#include <fstream>
```

Граф включаемых заголовочных файлов для Logs.h:



Граф файлов, в которые включается этот файл:



Классы

- class [Logs](#)
Класс [Logs](#).

5.4.1 Подробное описание

Заголовочный файл для модуля [Logs](#).

Автор

Каргин Н.С.

Версия

1.0

Дата

20.12.2023

Авторство

ИБСТ ПГУ

Предупреждения

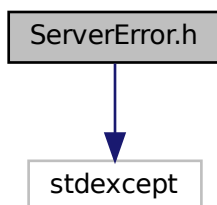
Курсовая работа студента

5.5 Файл ServerError.h

Заголовочный файл для модуля [ServerError](#).

```
#include <stdexcept>
```

Граф включаемых заголовочных файлов для ServerError.h:



Классы

- class [ServerError](#)
Класс [ServerError](#).

5.5.1 Подробное описание

Заголовочный файл для модуля [ServerError](#).

Автор

Каргин Н.С.

Версия

1.0

Дата

20.12.2023

Авторство

ИБСТ ПГУ

Предупреждения

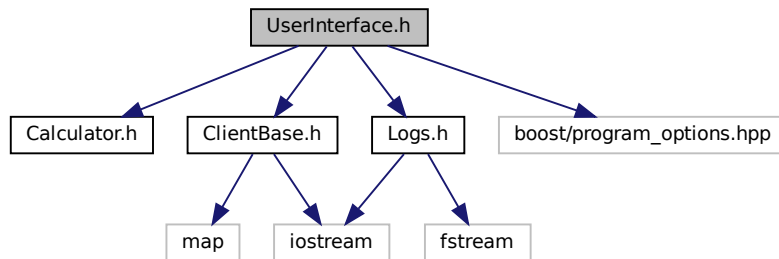
Курсовая работа студента

5.6 Файл UserInterface.h

Заголовочный файл для модуля UsersInterface.

```
#include "Calculator.h"  
#include "ClientBase.h"  
#include "Logs.h"  
#include <boost/program_options.hpp>
```

Граф включаемых заголовочных файлов для UserInterface.h:



Классы

- class [UserInterface](#)
Класс UsersInterface.

5.6.1 Подробное описание

Заголовочный файл для модуля UsersInterface.

Автор

Каргин Н.С.

Версия

1.0

Дата

20.12.2023

Авторство

ИБСТ ПГУ

Предупреждения

Курсовая работа студента

Предметный указатель

- Calculator, [7](#)
 - math, [7](#)
- Calculator.h, [17](#)
- ClientBase, [8](#)
 - find, [8](#)
 - read, [9](#)
- ClientBase.h, [18](#)
- Communicator, [9](#)
 - getClientSocket, [10](#)
 - Initialize, [10](#)
 - WaitClient, [10](#)
- Communicator.h, [19](#)
- Error
 - Logs, [12](#)
- find
 - ClientBase, [8](#)
- getClientSocket
 - Communicator, [10](#)
- Info
 - Logs, [12](#)
- Initialize
 - Communicator, [10](#)
- Logs, [11](#)
 - Error, [12](#)
 - Info, [12](#)
 - Logs, [11](#)
 - Warning, [12](#)
- Logs.h, [20](#)
- math
 - Calculator, [7](#)
- read
 - ClientBase, [9](#)
- ServerError, [13](#)
 - ServerError, [14](#)
- ServerError.h, [22](#)
- UserInterface, [14](#)
 - UserInterface, [15](#)
- UserInterface.h, [23](#)
- WaitClient
 - Communicator, [10](#)
- Warning
 - Logs, [12](#)