

CRC32-IEEE 802.3

1.0

Создано системой Doxygen 1.8.17



---

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 <code>calculatorCrc</code> . . . . .	7
4.1.1 Подробное описание . . . . .	7
4.1.2 Конструктор(ы) . . . . .	8
4.1.2.1 <code>calculatorCrc()</code> . . . . .	8
4.1.3 Методы . . . . .	8
4.1.3.1 <code>calculateCRC()</code> . . . . .	8
4.1.3.2 <code>FileRead()</code> . . . . .	9
4.2 Класс <code>Client</code> . . . . .	9
4.2.1 Подробное описание . . . . .	10
4.3 Класс <code>error</code> . . . . .	10
4.3.1 Подробное описание . . . . .	11
4.3.2 Конструктор(ы) . . . . .	11
4.3.2.1 <code>error()</code> [1/2] . . . . .	11
4.3.2.2 <code>error()</code> [2/2] . . . . .	11
5 Файлы	13
5.1 Файл <code>src32.h</code> . . . . .	13
5.1.1 Подробное описание . . . . .	14
5.2 Файл <code>help.h</code> . . . . .	14
5.2.1 Подробное описание . . . . .	15
Предметный указатель	17



# Глава 1

## Иерархический список классов

### 1.1 Иерархия классов

Иерархия классов.

calculatorCrc . . . . .	7
Client . . . . .	9
invalid_argument	
error . . . . .	10



## Глава 2

# Алфавитный указатель классов

### 2.1 Классы

Классы с их кратким описанием.

<code>calculatorCrc</code>	Описание класса <code>calculatorCRC</code> . . . . .	7
<code>Client</code>	Данный класс выполняет функцию выдачи справки пользователю о режимах работы программы . . . . .	9
<code>error</code>	Описание класса <code>error</code> . . . . .	10





## Глава 3

# Список файлов

### 3.1 Файлы

Полный список документированных файлов.

<a href="#">crc32.h</a>	Вычисление контрольной суммы CRC32-IEEE 802.3 . . . . .	<a href="#">13</a>
<a href="#">help.h</a>	. . . . .	<a href="#">14</a>



## Глава 4

# Классы

### 4.1 Класс calculatorCrc

Описание класса calculatorCRC.

```
#include <crc32.h>
```

Открытые члены

- `calculatorCrc ()=delete`  
Запрет объявления объекта класса без параметров
- `calculatorCrc (const unsigned long pol, string filename)`  
Конструктор, инициализирующий значение образующего полинома.
- `unsigned long FileRead (string filename, const unsigned long polynom)`  
Метод класса FileRead.
- `unsigned long calculateCRC (const char *message, unsigned long len, unsigned long polynom)`  
Метод класса calculateCRC.

Закрытые данные

- `unsigned long polynom`  
Атрибут, хранящий числовое значение образующего полинома.
- `string filename`  
Атрибут, хранящий полный путь к файлу, из которого будут читаться данные.

#### 4.1.1 Подробное описание

Описание класса calculatorCRC.

Класс реализует вычисление контрольной суммы CRC с использованием образующего полинома: 0x04C11DB7

## 4.1.2 Конструктор(ы)

### 4.1.2.1 calculatorCrc()

```
calculatorCrc::calculatorCrc (
    const unsigned long pol,
    string filename )
```

Конструктор, инициализирующий значение образующего полинома.

В данном конструкторе возбуждаются исключительные ситуации, связанные с возможными ошибками с файлами.

Аргументы

filename	- полный путь к файлу
polynom	- значение образующего полинома

## 4.1.3 Методы

### 4.1.3.1 calculateCRC()

```
unsigned long calculatorCrc::calculateCRC (
    const char * message,
    unsigned long len,
    unsigned long polynom )
```

Метод класса calculateCRC.

Рассчитывает значение CRC по следующему алгоритму:

- 1 - В регистр заносится начальное значение 0xFFFFFFFF (или 32 единичных бита);
- 2 - Выполняется цикл с числом повторений, равному количеству байт в сообщении;
- 3 - Текущее значение регистра складывается по модулю 2 (операция XOR) с каждым байтом сообщения;
- 4 - Запускается вложенный цикл, в котором проверяется каждый бит каждого байта сообщения;
- 5 - Если выбранный бит равен единице: регистр сдвигается влево на 1 бит, затем складывается по модулю 2 (операция XOR) со значением образующего полинома;
- 6 - Если выбранный бит не равен единице: регистр сдвигается влево на 1 бит без дополнительных действий;
- 7 - По окончании основного и вложенного циклов инвертируются биты содержимого регистра.

## Аргументы

message	- массив, состоящий из байтов сообщения.
len	- длина массива байтов сообщения.
polynom	- значение образующего полинома (0x04C11DB7).

## Возвращает

Метод возвращает вычисленное значение CRC.

## 4.1.3.2 FileRead()

```
unsigned long calculatorCrc::FileRead (
    string filename,
    const unsigned long polynom )
```

Метод класса FileRead.

Метод открывает файл, для которого будет вычислена контрольная сумма, определяет его размер, записывает данные в буфер и вызывает метод, вычисляющий значение контрольной суммы

## Аргументы

filename	- полный путь к файлу
polynom	- значение образующего полинома

## Возвращает

Возвращает результат работы функции "calculateCRC", вычисляющей значение CRC.

Объявления и описания членов классов находятся в файлах:

- [crc32.h](#)
- [crc32.cpp](#)

## 4.2 Класс Client

Данный класс выполняет функцию выдачи справки пользователю о режимах работы программы

```
#include <help.h>
```

## Открытые члены

- void [help](#) ()  
Метод, выводящий справочную информацию на экран

## Закрытые данные

- `const string helper = "Справочная информация:\n"/crc - вычисление контрольной суммы CRC32-IEEE 802.3\n"/exit - завершение работы программы\n"`Пример правильной записи полного пути к файлу: `/home/student/test1.txt"`

Атрибут, содержащий справочную информацию

### 4.2.1 Подробное описание

Данный класс выполняет функцию выдачи справки пользователю о режимах работы программы

Объявления и описания членов классов находятся в файлах:

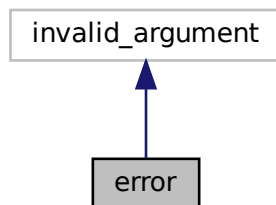
- `help.h`
- `help.cpp`

## 4.3 Класс error

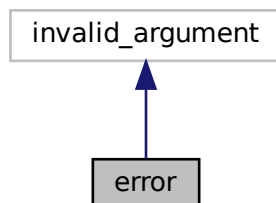
Описание класса `error`.

```
#include <crc32.h>
```

Граф наследования: `error`:



Граф связей класса `error`:



## Открытые члены

- [error](#) (const string &what\_arg)  
Конструктор принимает на вход строку и возбуждает исключение
- [error](#) (const char \*what\_arg)  
Конструктор принимает на вход си-строку и возбуждает исключение

### 4.3.1 Подробное описание

Описание класса error.

Данный класс предназначен для возбуждения и обработки ошибок, которые могут возникнуть при взаимодействии с программой.

### 4.3.2 Конструктор(ы)

#### 4.3.2.1 error() [1/2]

```
error::error (  
    const string & what_arg )    [inline], [explicit]
```

Конструктор принимает на вход строку и возбуждает исключение

Аргументы

what_arg	
----------	--

#### 4.3.2.2 error() [2/2]

```
error::error (  
    const char * what_arg )    [inline], [explicit]
```

Конструктор принимает на вход си-строку и возбуждает исключение

Аргументы

what_arg	
----------	--

Объявления и описания членов класса находятся в файле:

- [crc32.h](#)





## Глава 5

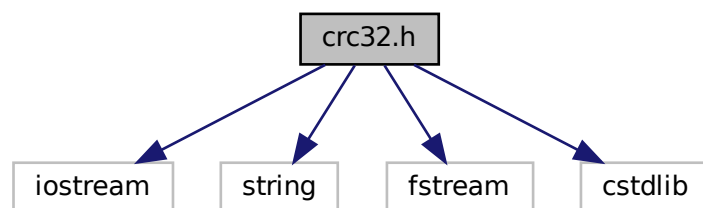
# Файлы

### 5.1 Файл crc32.h

Вычисление контрольной суммы CRC32-IEEE 802.3.

```
#include <iostream>
#include <string>
#include <fstream>
#include <cstdlib>
```

Граф включаемых заголовочных файлов для crc32.h:



## Классы

- class `calculatorCrc`  
Описание класса calculatorCRC.
- class `error`  
Описание класса error.

### 5.1.1 Подробное описание

Вычисление контрольной суммы CRC32-IEEE 802.3.

Автор

Ерисковский А.И

Дата

11/06/21

Версия

1.0

Авторство

ИБСТ ПГУ

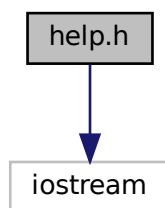
Предупреждения

Это учебный пример

## 5.2 Файл help.h

```
#include <iostream>
```

Граф включаемых заголовочных файлов для help.h:



### Классы

- class [Client](#)

Данный класс выполняет функцию выдачи справки пользователю о режимах работы программы

### 5.2.1 Подробное описание

Автор

Ерисковский А.И

Версия

1.0

Дата

11/06/21

Авторство

ИБСТ ПГУ

Предупреждения

Это учебный пример



# Предметный указатель

- calculateCRC
  - calculatorCrc, [8](#)
- calculatorCrc, [7](#)
  - calculateCRC, [8](#)
  - calculatorCrc, [8](#)
  - FileRead, [9](#)
- Client, [9](#)
- crc32.h, [13](#)
- error, [10](#)
  - error, [11](#)
- FileRead
  - calculatorCrc, [9](#)
- help.h, [14](#)