# Программа для стеганографии методом LSB в BMP изображениях .

1.0

Создано системой Doxygen 1.8.17

# 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 Steganography

Класс для реализации стеганографии в BMP изображениях методом LSB.

#include <Steganography.h>

### Открытые члены

- Steganography (const string &key)
  Конструктор класса стеганографии, инициализирующий объект класса определенной степенью кодирования.
- int Concealment (const string &bmp_path, const string &hidden_path)
  Метод сокрытия информации в файле-контейнере
- int Extraction (const string &bmp_path, const string &hidden_path, const string &hiddensize)
  Метод извлечения скрываемой информации из файла-контейнера

### Закрытые члены

- unsigned char hmask ()
  Метод, формирующий битовую маску для байта файла, информацию из которого необходимо скрыть.
- unsigned char imask ()
  Метод, формирующий битовую маску для байта Bmp контейнера.

### Закрытые данные

- int degree
  Степень кодирования

### 4.1.1 Подробное описание

Класс для реализации стеганографии в BMP изображениях методом LSB.

Предупреждения

Реализация только для степени кодирования, равной 1/2/4/8.

### 4.1.2 Конструктор(ы)

#### 4.1.2.1 Steganography()

```
Steganography::Steganography (
                const string & key )
```

Конструктор класса стеганографии, инициализирующий объект класса определенной степенью кодирования.

Аргументы

| | |
|---|---|
| key | - степень кодирования, вводимая пользователем. |

Полученная степень кодирования в виде строки проверяется на корректность ввода, после чего с помощью функции stoi инициализируется приватным полем degree целочисленным типом int.

### 4.1.3 Методы

#### 4.1.3.1 Concealment()

```
int Steganography::Concealment (
                const string & bmp_path,
                const string & hidden_path )
```

Метод сокрытия информации в файле-контейнере

Аргументы

| | |
|---|---|
| bmp_path | - путь к файлу-контейнеру |
| hidden_path | - путь к скрываемому файлу |

После создания объектов-файлов осуществляется их проверка функциями класса исключений на корректность формата файла-контейнера и возможность открытия файлов. Переменная hiddensize хранит в себе размер скрываемого файла. Переменные Image,Hidden,ImageByte и HiddenByte моделируют байты используемых файлов. Далее запускается цикл, работающий следующим образом: пока не будет считано количество байт, равное объему скрываемого файла считываются байты скрываемого файла.

```
for (int i=0; i<hiddensize; i++) {
 hidden.read (reinterpret_cast<char*>(&Hidden),sizeof(unsigned char));
```

После считывания одного байта его размер в битах делится на степень кодирования, именно столько итераций внутреннего цикла приходится на 1 байт скрываемой информации. Во внутреннем цикл считывает байт контейнера, в котором обнуляются последние degree битов с помощью соответствующей маски. В байте скрываемой информации в свою очередь единичными битами остаются только

первые degree битов. Далее происходит сдвиг этих битов в конец байта скрываемой информации и сложение двух модифицированных байтов (файла-контейнера и скрываемого файла). Полученный файл записывается обратно в файл-контейнер на то же место, а в байте скрываемой информации делается сдвиг на степень кодирования, чтобы продолжать успешно «цеплять» биты и отправлять их в конец байтов контейнера.

```
for (int i=0; i<8; i+=degree) {
    bmp.read(reinterpret_cast <char*>(&Image), sizeof(char));
    bmp.seekg(-1,fstream::cur);
    ImageByte = Image&imask();
    HiddenByte = Hidden&hmask();
    HiddenByte »= (8-degree);
    ImageByte |= HiddenByte;
    bmp.write (reinterpret_cast <char*>(&ImageByte), sizeof(char));
    Hidden«=degree;
}
```

После завершения внешнего цикла файлы закрываются и функция завершает свою работу.

#### 4.1.3.2 Extraction()

```
int Steganography::Extraction (
                const string & bmp_path,
                const string & hidden_path,
                const string & hiddensize )
```

Метод извлечения скрываемой информации из файла-контейнера

Аргументы

| | |
|---|---|
| bmp_path | - путь к файлу-контейнеру |
| hidden_path | - путь к новому файлу, куда будет извлечена скрываемая информация |
| hiddensize | - размер скрываемой информации в байтах, задается пользователем |

После создания объектов-файлов метод проверяет файл-контейнер на формат BMP и возможность открытия. Также происходит проверка нового файла на возможность создания, а также проверяется количество байт, указанное пользователем на корректность ввода и на возможность извлечь данное количество байт из контейнера. Переменная Image моделирует байт контейнера без изменений, тогда как переменная ImageByte моделирует байт контейнера после применения к нему маски, противоположной маске для файла-контейнера при сокрытии. Переменная Symbol моделирует байт, который после заполнения будет записан в новый файл. Далее запускается цикл, в котором инициализируются байты, пока их количество не станет равно кол-ву байтов, заданному пользователем для извлечения

```
for (int i=0; i<DigitHiddenSize; i++) {
 unsigned char Symbol = 0;
```

Затем во внутреннем цикле, который проходит столько раз , чему равно отношение размера байт (8) к степени кодировки, считывается байт из файла-контейнера с скрываемой информацией. К нему применяется маска, которая обнуляет первые degree битов. Далее этот байт контейнера складывается с созданным в цикле байтом. Далее записанные байты сдвигаются влево на degree, чтобы созданный в цикле байт полностью заполнялся. После окончания внутреннего цикла заполненный байт записывается в новый файл.

```
for (int i=0; i<8; i+=degree) {
    bmp.read(reinterpret_cast <char*>(&Image),sizeof(char));
    ImageByte = Image&image_mask;
    Symbol«=degree;
    Symbol |= ImageByte;
}
 hidden.write (reinterpret_cast<char*>(&Symbol),sizeof(unsigned char));
```

После завершения внешнего цикла файлы закрываются и функция завершает свою работу.

#### 4.1.3.3 hmask()

unsigned char Steganography::hmask ( ) [private]

Метод, формирующий битовую маску для байта файла, информацию из которого необходимо скрыть.

Метод формирует маску, в которой оставляет единичными только первые degree битов. Маска используется при операции сокрытия.

Возвращает

Маска для файла со скрываемой информацией.

#### 4.1.3.4 imask()

unsigned char Steganography::imask ( ) [private]

Метод, формирующий битовую маску для байта Bmp контейнера.

Метод формирует маску, в которой обнуляет последние degree бит. Маска используется при операции сокрытия, а также противоположная ей маска используется при операции извлечения.

Возвращает

Маска для Bmp контейнера.

### 4.1.4 Данные класса

#### 4.1.4.1 degree

int Steganography::degree [private]

Степень кодирования

Степень кодирования определяет количество последних бит в байте изображения, которые будут использоваться как контейнер для бит скрываемой информации. Чем больше степень кодирования, тем сильнее отличается исходный-файл контейнер от файла-контейнера с скрываемой информацией.

Объявления и описания членов классов находятся в файлах:

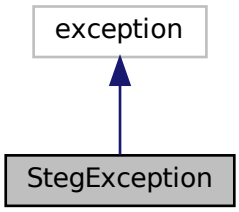
- Steganography.h
- Steganography.cpp

## 4.2 Класс StegException

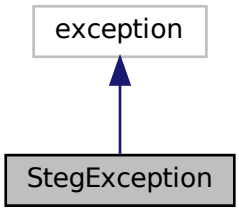
Собственный класс исключений

#include <Exception.h>

Граф наследования:StegException:




Граф связей класса StegException:




### Открытые члены

- StegException (const string &err, const string &desc)
  Конструктор инициализации ошибки строками.
- string what () noexcept
  Метод, возвращающий информацию об ошибке.

### Открытые статические члены

- static bool DegreeCheck (const string &key)
  Метод проверки степени кодирования.
- static bool FileCheck (const string &filepath)
  Метод проверки файла на существование/корректный ввод пути.

- static bool BMPCheck (const string &bmp_path)

  Метод проверки файла-контейнера на формат BMP.
- static bool ConFileSizeCheck (const string &bmp_path, const string &hidden_path, const int &key)

  Метод проверки на возможность скрыть данные из файла в контейнере при данной степени кодировки.
- static bool ExFileSizeCheck (const string &bmp_path, const string &hidden_path, const int &key, const string &hiddensize)

  Метод проверки на возможность извлечь из контейнера определенное количество байтов при данной степени кодировки.
- static bool NewFileCheck (const string &hidden_path)

  Метод проверки нового файла для записи в него извлеченных данных из контейнера на существование/корректный ввод пути.
- static bool FileComparsion (const string &hidden_path, const string &newfile_path)

  Метод для тестирования корректности работы функций сокрытия/извлечения.

### Закрытые данные

- string error

  поле ошибки, в котором указывается ее вид.
- string description

  поле ошибки, в котором указывается ее описание.

## 4.2.1 Подробное описание

Собственный класс исключений

## 4.2.2 Конструктор(ы)

### 4.2.2.1 StegException()

```
StegException::StegException (
                const string & err,
                const string & desc )
```

Конструктор инициализации ошибки строками.

Аргументы

| | |
|---|---|
| err | - вид ошибки. |
| desc | - описание ошибки. |

err присваивается приватному полю error,а desc присваивается приватному полю description.

### 4.2.3 Методы

#### 4.2.3.1 BMPCheck()

bool StegException::BMPCheck (
const string & bmp_path ) [static]

Метод проверки файла-контейнера на формат BMP.

Аргументы

| | |
|---|---|
| bmp_path | - путь к файлу-контейнеру. |

Функция проверят первые 2 байта файла на то,составляют ли они комбинацию BM.В случае, если нет, бросается исключение.

#### 4.2.3.2 ConFileSizeCheck()

bool StegException::ConFileSizeCheck (
const string & bmp_path,
const string & hidden_path,
const int & key ) [static]

Метод проверки на возможность скрыть данные из файла в контейнере при данной степени кодировки.

Аргументы

| | |
|---|---|
| bmp_path | - путь к файлу-контейнеру. |
| hidden_path | - путь к файлу, данные из которого будут скрываться. |
| key | - степень кодирования. |

Метод вычисляет объем файла-контейнера и файла, данные из которого будут скрываться и проверяет, может ли в контейнер быть помещен объем файла со скрываемой информацией при заданной степени кодировки. Если не может, выбрасывается исключение.

#### 4.2.3.3 DegreeCheck()

bool StegException::DegreeCheck (
const string & key ) [static]

Метод проверки степени кодирования.

Аргументы

| | |
|---|---|
| key | - степень кодирования, введенная пользователем для выполнения операций сокрытия/извлечения. |

Бросает исключение в случае если: строка с вводом степени пустая, степень содержит символы, не являющиеся цифрами, степень не является одним из перечисленных чисел:1/2/4/8

#### 4.2.3.4 ExFileSizeCheck()

```
bool StegException::ExFileSizeCheck (
            const string & bmp_path,
            const string & hidden_path,
            const int & key,
            const string & hiddensize )   [static]
```

Метод проверки на возможность извлечь из контейнера определенное количество байтов при данной степени кодировки.

Аргументы

| | |
|---|---|
| bmp_path | - путь к файлу-контейнеру. |
| hidden_path | - путь к файлу, данные из которого будут скрываться. |
| key | - степень кодирования. |
| hiddensize | - количество байтов для извлечения (вводится пользователем). |

Первоначально метод проверяет строку с введеным количеством байтов для извлечения на корректность ввода: если строка пустая или содержит символы не являющиеся цифрами выбрасывается исключение. Далее метод вычисляет объем файла-контейнера и максимальное количество байт,доступное для извлечения при данной степени кодировки. Если заданное пользователем число байт на извлечение больше чем вычисленное максимально допустимое число байт для извлечения, то бросается исключение.

#### 4.2.3.5 FileCheck()

```
bool StegException::FileCheck (
            const string & filepath )   [static]
```

Метод проверки файла на существование/корректный ввод пути.

Аргументы

| | |
|---|---|
| filepath | - путь к файлу. |

Бросает исключение в случае если: файл по заданному пути не открывается.

#### 4.2.3.6 FileComparsion()

```
bool StegException::FileComparsion (
            const string & hidden_path,
            const string & newfile_path )   [static]
```

Метод для тестирования корректности работы функций сокрытия/извлечения.

Аргументы

| | |
|---|---|
| hidden_path | - путь к файлу, данные из которого будут скрываться в контейнере. |
| newfile_path | - путь к новому файлу, куда записываются извлеченные данные из контейнера. |

В тестовом сценарии, где используется данная функция выполняется пара операций: сокрытие и извлечение. Далее содержимое файла, данные из которого скрывались, и содержимое файла, в который были записаны скрываемые данные из контейнера сравниваются на полное совпадение. В случае если если файлы не равны выбрасывается исключение, не дающее завершить тест успешно.

#### 4.2.3.7 NewFileCheck()

bool StegException::NewFileCheck (
const string & hidden_path ) [static]

Метод проверки нового файла для записи в него извлеченных данных из контейнера на существование/корректный ввод пути.

Аргументы

| | |
|---|---|
| hidden_path | - путь к новому файлу. |

Бросает исключение в случае если: файл по заданному пути не открывается (не создается).

Объявления и описания членов классов находятся в файлах:

- Exception.h
- Exception.cpp

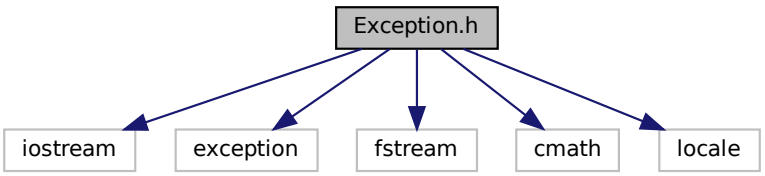
# Глава 5

# Файлы

## 5.1 Файл Exception.h

Описание класса StegException.

#include <iostream>
#include <exception>
#include <fstream>
#include <cmath>
#include <locale>
Граф включаемых заголовочных файлов для Exception.h:




### Классы

- class StegException

  Собственный класс исключений

### 5.1.1 Подробное описание

Описание класса StegException.

Автор

Самборский И.С.

Версия

1.0

Дата

30.05.2021

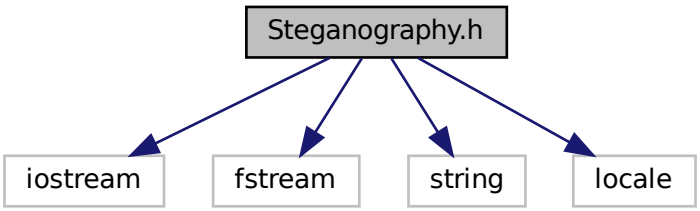
Авторство

ИБСТ ПГУ

## 5.2 Файл Steganography.h

Описание класса Steganography.

#include <iostream>
#include <fstream>
#include <string>
#include <locale>
Граф включаемых заголовочных файлов для Steganography.h:




### Классы

- class Steganography

  Класс для реализации стеганографии в BMP изображениях методом LSB.

### 5.2.1 Подробное описание

Описание класса Steganography.

Автор

Самборский И.С.

Версия

1.0

Дата

30.05.2021

Авторство

ИБСТ ПГУ

# Предметный указатель