Шифр табличной маршрутной перестановки

1.0

Создано системой Doxygen 1.8.17

1 Иерархический список классов

1.1 Иерархия классов

Иерархия классов.

2 Алфавитный указатель классов

2.1 Классы

Классы с их кратким описанием.

3 Список файлов

3.1 Файлы

Полный список документированных файлов.

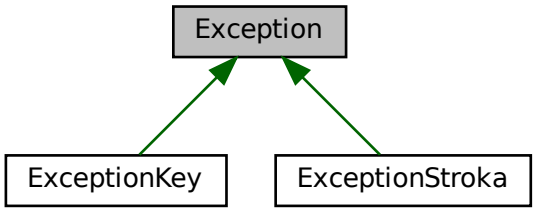
4 Классы

4.1 Класс Exception

Абстрактный класс для исключений

#include <Exception.h>

Граф наследования:Exception:



Открытые члены

- virtual string what ()=0

  Предназначен для вывода описания ошибки.
- virtual int code ()=0

  Чисто виртуальный метод. Предназначен для вывода кода ошибки.
- virtual string fix ()=0

  Чисто виртуальный метод. Предназначен для вывода информации об исправлении ошибки.

Защищенные данные

- string error

  атрибут, хранящий описание ошибки
- int num

  атрибут, хранящий информацию о коде ошибки
- string correction

  атрибут, хранящий информацию об исправлении ошибки

4.1.1 Подробное описание

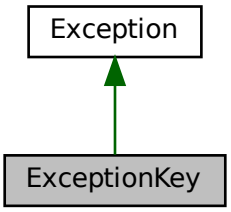
Абстрактный класс для исключений

Объявления и описания членов класса находятся в файле:

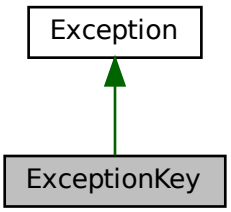
- Exception.h

4.2 Класс ExceptionKey

Граф наследования:ExceptionKey:



Граф связей класса ExceptionKey:



Открытые члены

- ExceptionKey ()=delete

  Запрещающий конструктор без параметров
- ExceptionKey (const int &num, const string &error, const string &correction)

  Конструктор с параметрами
- string what () override

  Предназначен для вывода описания ошибки.
- string fix () override

  Чисто виртуальный метод. Предназначен для вывода информации об исправлении ошибки.
- int code () override

  Чисто виртуальный метод. Предназначен для вывода кода ошибки.

Открытые статические члены

- static bool check_key (const wstring &data, const string str_key)

  Статический метод, проверяющий ключ при шифровании или расшифровании на наличие ошибок

Дополнительные унаследованные члены

4.2.1 Конструктор(ы)

4.2.1.1 ExceptionKey() ExceptionKey::ExceptionKey (
const int & num,
const string & error,
const string & correction)

Конструктор с параметрами

Аргументы

num	- целочисленное число, хранящее информацию о коде ошибки.
error	- строка, хранящая описание ошибки.
correction	- строка, хранящая информацию об исправлении ошибки.

4.2.2 Методы

4.2.2.1 check_key() bool ExceptionKey::check_key (
const wstring & data,
const string str_key) [static]

Статический метод, проверяющий ключ при шифровании или расшифровании на наличие ошибок

Сначала ключ проверяется на пустоту при помощи обычного условия. Если ключ является пустым, то с помощью ключевого слова "throw" возбуждается исключение типа "ExceptionKey". Далее ключ проверяется на допустимые символы при помощи обычного условия. То есть, если в ключе присутствуют символы каких-либо алфавитов или специальные символы, то вобуждается исключение с помощью ключевого слова "throw" типа "ExceptionKey.

Если две первые проверки завершились успехом, то ключ проверяется на нужный размер и на натуральность при помощи обычного условия. Если ключ является ненатуральным числом или не соотвествует нужному размеру, то вобуждается исключение с помощью ключевого слова "throw" типа "ExceptionKey.

Ключ является корректным только в том случае, если он является натуральным числом и не превышает размера строки для шифрования или расшифрования.

Аргументы

data	- std::wstring, строка, которую нужно зашифровать или расшифровать.
str_key	- std::string, ключ, который нужно проверить при шифровании или расшифровании.

Возвращает

значение "true", если проверки завершились успешно.

Исключения

ExceptionKey,если	• ключ оказался пустым; • в ключе присутствует недопустимые символы; • ключ не соответствует нужному размеру.

Объявления и описания членов классов находятся в файлах:

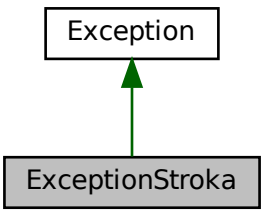
- ExceptionKey.h
- ExceptionKey.cpp

4.3 Класс ExceptionStroka

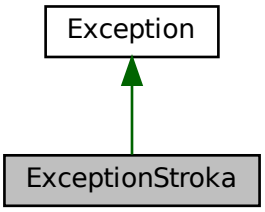
Класс для исключений строки при шифрования или расшифрования

#include <ExceptionStroka.h>

Граф наследования:ExceptionStroka:



Граф связей класса ExceptionStroka:



Открытые члены

- ExceptionStroka ()=delete

  Запрещающий конструктор без параметров
- ExceptionStroka (const int &num, const string &error, const string &correction)

  Конструктор с параметрами
- string what () override

  Предназначен для вывода описания ошибки.
- string fix () override

  Чисто виртуальный метод. Предназначен для вывода информации об исправлении ошибки.
- int code () override

  Чисто виртуальный метод. Предназначен для вывода кода ошибки.

Открытые статические члены

- static bool check_stroka (const string data)

  Статический метод, проверяющий строку при шифровании или расшифровании на наличие ошибки

Дополнительные унаследованные члены

4.3.1 Подробное описание

Класс для исключений строки при шифрования или расшифрования

4.3.2 Конструктор(ы)

4.3.2.1 ExceptionStroka() ExceptionStroka::ExceptionStroka (
const int & num,
const string & error,
const string & correction)

Конструктор с параметрами

Аргументы

num	- целочисленное число, хранящее информацию о коде ошибки.
error	- строка, хранящая описание ошибки.
correction	- строка, хранящая информацию об исправлении ошибки.

4.3.3 Методы

4.3.3.1 check_stroka() bool ExceptionStroka::check_stroka (const string data) [static]

Статический метод, проверяющий строку при шифровании или расшифровании на наличие ошибки

Строка проверяются на пустату при помощи обычного условия. Если строка является пустой, то с помощью ключего слова "throw" возбуждается исключение типа "ExceptionStroka".

Аргументы

data	- std::string, строка, которую нужно проверить при шифровании или расшифровании.

Возвращает

значение "true", если проверка завершилась успешно.

Исключения

ExceptionStroka,если	строка, пришедшая на вход оказалось пустой.

Объявления и описания членов классов находятся в файлах:

- ExceptionStroka.h
- ExceptionStroka.cpp

4.4 Класс Help

Справка о режимах работы программы

#include <Help.h>

Открытые члены

- void PrintHelp ()

  Выводит справку о режимах работы программы, если она понадобится пользователю.

Закрытые данные

- const string help = "Справка о работе программы:\n Encode - режим шифрования шифром табличной маршрутной перестановки.\n Decode - режим расшифрования шифра табличной маршрутной перестановки.\n exit - завершить работу программы.\n"

  атрибут, хранящий справку о режимах работы программы

4.4.1 Подробное описание

Справка о режимах работы программы

Объявления и описания членов классов находятся в файлах:

- Help.h
- Help.cpp

4.5 Класс PermutationCipher

Класс для шифрования и расшифрования шифра табличной маршрутной перестановки

#include <PermutationCipher.h>

Открытые члены

- PermutationCipher ()=delete

  Запрещающий конструктор без параметров
- PermutationCipher (int k)

  Конструктор для ключа
- wstring EncodePermutationCipher (PermutationCipher key, wstring &data)

  Метод , предназначенный для шифрования шифром табличной маршрутной перестановки
- wstring DecodePermutationCipher (PermutationCipher key, wstring &data)

  Метод , предназначенный для расшифрования шифра табличной маршрутной перестановки

Закрытые данные

- int key

  атрибут, хранящий ключ для шифрования или расшифрования

4.5.1 Подробное описание

Класс для шифрования и расшифрования шифра табличной маршрутной перестановки

4.5.2 Конструктор(ы)

4.5.2.1 PermutationCipher() PermutationCipher::PermutationCipher (int k)

Конструктор для ключа

Аргументы

целочисленное	число ключ

число, которое пришло на вход записывается в "private" атрибут с названием "key"

4.5.3 Методы

4.5.3.1 DecodePermutationCipher() wstring PermutationCipher::DecodePermutationCipher (PermutationCipher key, wstring & data)

Метод , предназначенный для расшифрования шифра табличной маршрутной перестановки

Аргументы

экземляр	класса "PermutationCipher", в котором установился ключ
std::wstring	- строка, которую нужно расшифровать

Сначала вычисляется количество строк для таблицы по формуле.
```
const int stroki = ((data.size()-1)/key.key)+1; // количество строк по формуле
```

Затем создаётся двумерный массив типа "wchar_t", который имеет необходимый размер: количество строк вычисляется по формуле, а количество столбцов - это ключ, который устанавливается в экземляре класса "PermutationCipher".
```
wchar_t matr[stroki][key.key];
```

Далее в созданный двумерный массив записываются символы строки, которую нужно расшифровать. Запись символов просходит по следующему маршруту: сверху-вниз. То есть, запись происходит по столбцам.
```
for (auto i = 0; i < key.key; i++) {
 for (auto j = 0; j < stroki; j++ ) {
    }
}
```

Примечение - если в таблице остаются незаполненные "ячейки", то в них записыватеся символ "пробела". Такая ситуации может произойти, если длина строки, которая пришла на вход не кратна значению ключа.
```
if (index < data.size()) {
        matr[j][i] = data[index];
        index++;
     } else {
        matr[j][i] = ' ';
        index++;
    }
```

В конечном итоге происходит процесс расшифрования. Он заключается в том, что символы, которые находятся в двумерном массиве записываются в строку типа wstring с именем "Result" по следующему маршруту: слева-направа. То есть, чтение происходит по строкам.
```
for(auto i = 0; i < stroki; i++) {
  for (auto j = 0; j< key.key; j++) {
     if (index<data.size())
        Result.push_back(matr[i][j]);
       index++;
   }
  }
```

Возвращает

расшифрованная строка типа "wstring"

4.5.3.2 EncodePermutationCipher() wstring PermutationCipher::EncodePermutationCipher (PermutationCipher key, wstring & data)

Метод , предназначенный для шифрования шифром табличной маршрутной перестановки

Аргументы

экземляр	класса "PermutationCipher", в котором установился ключ
std::wstring	- строка, которую нужно зашифровать

Сначала вычисляется количество строк для таблицы по формуле.
```
const int stroki = ((data.size()-1)/key.key)+1; // количество строк по формуле
```

Затем создаётся двумерный массив типа "wchar_t", который имеет необходимый размер: количество строк вычисляется по формуле, а количество столбцов - это ключ, который устанавливается в экземляре класса "PermutationCipher".
```
wchar_t matr[stroki][key.key];
```

Далее в созданный двумерный массив записываются символы строки, которую нужно зашифровать. Запись символов просходит по следующему маршруту: слева-направа. То есть, запись происходит по строкам.
```
for (auto i = 0; i <stroki; i++) {
 for (auto j = 0; j < key.key; j++ ) {
    }
}
```

Примечение - если в таблице остаются незаполненные "ячейки", то в них записыватеся символ "пробела". Такая ситуации может произойти, если длина строки, которая пришла на вход не кратна значению ключа.
```
if (index < data.size()) {
      matr[i][j] = data[index]; // записываем символы строки в таблицу (слева-направо)
      index++;
     } else {
          matr[i][j] =' ';
     }
```

В конечном итоге происходит процесс шифрования. Он заключается в том, что символы, которые находятся в двумерном массиве записываются в строку типа wstring с именем "Result" по следующему маршруту: сверху-вниз. То есть, чтение происходит по столбцам.
```
for (auto i = 0; i < key.key; i++) {
  for (auto j = 0; j < stroki; j++ ) {
      if (index <= data.size())
         Result.push_back(matr[j][i]);
          index++;
    }
  }
```

Возвращает

зашифрованная строка типа "wstring"

Объявления и описания членов классов находятся в файлах:

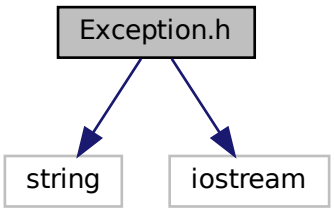
- PermutationCipher.h
- PermutationCipher.cpp

5 Файлы

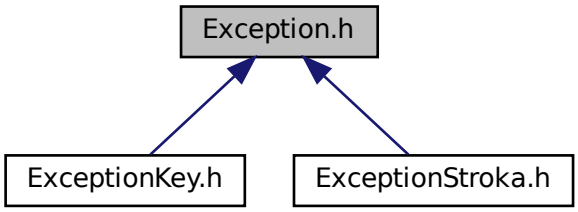
5.1 Файл Exception.h

Описание абстрактного класса Exception.

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



Граф файлов, в которые включается этот файл:



Классы

- class Exception

  Абстрактный класс для исключений

5.1.1 Подробное описание

Описание абстрактного класса Exception.

Автор

Авдонин А.Д.

Версия

1.0

Дата

20.05.2021

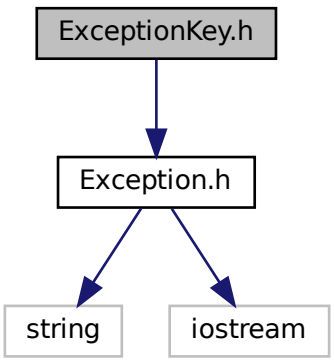
Авторство

ИБСТ ПГУ

5.2 Файл ExceptionKey.h

Описание класса ExceptionKey.

#include "Exception.h"
Граф включаемых заголовочных файлов для ExceptionKey.h:



Классы

- class ExceptionKey

5.2.1 Подробное описание

Описание класса ExceptionKey.

Автор

Авдонин А.Д.

Версия

1.0

Дата

20.05.2021

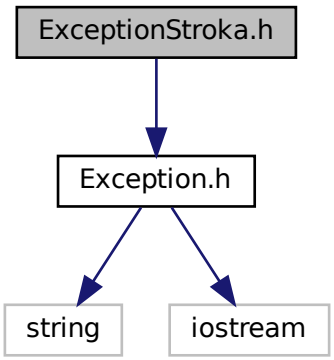
Авторство

ИБСТ ПГУ

5.3 Файл ExceptionStroka.h

Описание класса ExceptionStroka.

#include "Exception.h"
Граф включаемых заголовочных файлов для ExceptionStroka.h:



Классы

- class ExceptionStroka

  Класс для исключений строки при шифрования или расшифрования

5.3.1 Подробное описание

Описание класса ExceptionStroka.

Автор

Авдонин А.Д.

Версия

1.0

Дата

20.05.2021

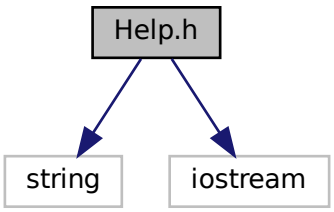
Авторство

ИБСТ ПГУ

5.4 Файл Help.h

Описание класса Help.

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



Классы

- class Help

  Справка о режимах работы программы

5.4.1 Подробное описание

Описание класса Help.

Автор

Авдонин А.Д.

Версия

1.0

Дата

20.05.2021

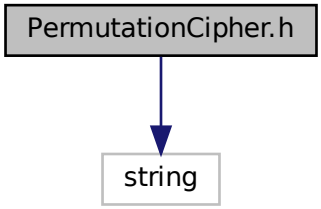
Авторство

ИБСТ ПГУ

5.5 Файл PermutationCipher.h

Описание класса PermutationCipher.

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



Классы

- class PermutationCipher

  Класс для шифрования и расшифрования шифра табличной маршрутной перестановки

5.5.1 Подробное описание

Описание класса PermutationCipher.

Автор

Авдонин А.Д.

Версия

1.0

Дата

20.05.2021

Авторство

ИБСТ ПГУ

Предметный указатель