

ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ «НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
УНИВЕРСИТЕТ
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»

Кафедра «Компьютерная безопасность»

**ОТЧЕТ
К ЛАБОРАТОРНОЙ РАБОТЕ №14**

по дисциплине

«Языки программирования»

Работу выполнил
студент группы СКБ-222

подпись, дата

А.С. Вагин

Работу проверил

подпись, дата

С.А. Булгаков

Москва 2023

Содержание

Постановка задачи	3
Основная часть	4
1 Описание хэш-функций	4
2 Описание функций	4
2.1 Вспомогательные функции для класса перечислений <i>Hashes</i>	4
2.2 Функция для получения файла в байтовом представлении	4
Приложение А	5
A.1 UML-диаграмма <i>clearStdin</i>	5
Приложение В	6
B.1 Код программы	6

Постановка задачи

Разработать консольную утилиту позволяющую выполнять проверку целостности файлов на основе механизма контрольных сумм. При запуске программа проверяет наличие файла 'Checksum.ini'. При его наличия выполняется проверка контрольных сумм для файлов указанных в нем, иначе если поток ввода не пуст, то из него считывается имя файла(ов) и, отделенные символом табуляции, их контрольные суммы (см. ключ '-a'). Если поток ввода пуст, проводится разбор параметров командной строки и, в зависимости от входных данных, выполняется действие либо выводится 'usage'.

Режимы запуска программы:

- * без параметров - использование файла 'Checksum.ini';
- * '-a algorithm', где 'algorithm' одно из 'crc32', 'md5', 'sha256' - задаёт алгоритм для расчёта контрольной суммы;
- * 'файл...' - имена файлов для вычисления контрольных сумм (см. ключ '-a').

Пример содержимого файла 'Checksum.ini':

```
'''  
; This is comment  
[CRC32]  
README.md=0x720C65BA  
'''
```

Основная часть

1 Описание хэш-функций

Для данной лабораторной работы были взяты готовые реализации требующихся хэш-функций, а именно SHA256, CRC32 и MD5.

Во всех классах была реализована функция *hash*, которая принимала на вход массив типа *unsigned char* и его длина, и возвращала *std::string*, который является хэшем в шестнадцатеричном формате.

Также был создан класс *HashFactory* для упрощенного доступа ко всем функциям.

Класс *HashFactory* принимает на вход нужные для хэширования данные, а также аргумент из класса перечислений *Hashes*, для определения нужной хэш-функции.

2 Описание функций

2.1 Вспомогательные функции для класса перечислений *Hashes*

Были разработаны функции *algorithmToText* и *parseAlgorithm* для преобразования аргумента класса в текст и обратно соответственно.

2.2 Функция для получения файла в байтовом представлении

Функция *getBytesFromFile* получает на вход имя файла в виде *std::string*. Если файла не существует, вызывается ошибка. Функция возвращает *std::vector* с байтами файла в формате *unsigned char*.

Приложение А

А.1 UML-диаграмма *clearStdin*

Приложение В

В.1 Код программы