

Математические основы защиты информации и информационной безопасности.

Лабораторная работа №1.

Подмогильный Иван Александрович.

Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение лабораторной работы	7
4	Выводы	11

List of Figures

3.1	Caesar cipher	7
3.2	Caesar decipher	7
3.3	Atbash cipher	8
3.4	Atbash decipher	8
3.5	Header file	8
3.6	CmakeLists.txt file	8
3.7	main.cpp	9
3.8	main.cpp	9
3.9	tests	10

List of Tables

1 Цель работы

Освоить на практике шифрование шифров Цезаря и Атбаша.

2 Задание

1. Реализовать шифр Цезаря
2. Реализовать шифр Атбаш

3 Выполнение лабораторной работы

Написал код для зашивровки кодов шифром Цезаря

```
#include "../include/CipherHelper.h"
#include <iostream>

const std::string CipherHelper::engAlphabetUpper = "ABCDEFGHIJKLMNOPQRSTUVWXYZ ";

void CipherCaesar::cipher(const std::string& message, int Key, std::string& encrypted){
    // char with index j => char with index (j + k) / mod 26
    if (!encrypted.empty()){
        throw std::invalid_argument( "encrypted is not empty!" );
    }

    if (!message.empty()){
        for (auto character : message){
            // test it, if it returns right index
            auto index = engAlphabetUpper.find(character);
            encrypted += engAlphabetUpper[(index + Key) % engAlphabetUpper.size()];
        }
    }
    else{
        encrypted = "";
    }
}
```

Figure 3.1: Caesar cipher

Написал код для дешифровки кодов шифром Цезаря

```
24 void CipherCaesar::decipher(const std::string& message, int Key, std::string& decrypted) {
25     if (!decrypted.empty()){
26         throw std::invalid_argument( "decrypted is not empty!" );
27     }
28
29     if (!message.empty()){
30         for (auto character : message){
31             auto index = engAlphabetUpper.find(character);
32             decrypted += engAlphabetUpper[ (engAlphabetUpper.size() + (index - Key)) % engAlphabetUpper.size() ];
33         }
34     }
35     else{
36         decrypted = "";
37     }
38 }
```

Figure 3.2: Caesar decipher

Написал код для зашивровки кодов шифром Атбаша

```

41 void CipherAtbash::cipher(const std::string &message, std::string &encrypted) {
42     if (lencrypted.empty()){
43         throw std::invalid_argument( "encrypted is not empty!" );
44     }
45
46     if (!message.empty()){
47         for (auto character : message){
48             // test it, if it returns right index
49             auto index = engAlphabetUpper.find(character);
50             encrypted += engAlphabetUpper[engAlphabetUpper.size() - 1 - index];
51         }
52     }
53     else{
54         encrypted = "";
55     }
56 }

```

Figure 3.3: Atbash cipher

Написал код для дешифровки кодов шифром Атбаша

```

57
58 void CipherAtbash::decipher(const std::string &message, std::string &decrypted) {
59     cipher(message, &decrypted);
60 }

```

Figure 3.4: Atbash decipher

Написал заголовочный файл для класса реализации CipherHelper

```

1  #ifndef LAB01_CIPHERHELPER_H
2  #define LAB01_CIPHERHELPER_H
3
4  #include <string>
5
6
7  class CipherHelper{
8  public:
9      static const std::string engAlphabetLower;
10     static const std::string engAlphabetUpper;
11 };
12
13 class CipherCaesar : CipherHelper {
14 public:
15     static void cipher (const std::string& message, int Key, std::string& encrypted);
16     static void decipher (const std::string& message, int Key, std::string& decrypted);
17 };
18
19 class CipherAtbash : CipherHelper {
20 public:
21     static void cipher (const std::string& message, std::string& encrypted);
22     static void decipher (const std::string& message, std::string& decrypted);
23 };
24
25
26
27 #endif //LAB01_CIPHERHELPER_H

```

Figure 3.5: Header file

Написал CMakeLists.txt файл, который создаёт библиотеку из класса CipherHelper и бинарник main

```

1  cmake_minimum_required(VERSION 3.20)
2  project(lab01)
3
4  set(CMAKE_CXX_STANDARD 14)
5
6  add_library(lab01 src/CipherHelper.cpp)
7
8  add_executable(main src/main.cpp)
9  target_link_libraries(main lab01)

```

Figure 3.6: CmakeLists.txt file

Написал main.cpp файл, в котором есть тесты реализованных функций. Часть шифра Цезаря:

```
4 int main(){
5     std::string msg1 = "HELLO WORLD";
6     std::string msg2 = "I LIKE CATS";
7     std::string msg3 = "INFBEZ KAIF";
8
9     std::string enc1 = "", enc2 = "", enc3 = "";
10    std::string dec1 = "", dec2 = "", dec3 = "";
11
12    CipherCaesar::cipher(msg1, Key: 3, &enc1);
13    CipherCaesar::cipher(msg2, Key: 3, &enc2);
14    CipherCaesar::cipher(msg3, Key: 3, &enc3);
15
16    CipherCaesar::decipher(enc1, Key: 3, &dec1);
17    CipherCaesar::decipher(enc2, Key: 3, &dec2);
18    CipherCaesar::decipher(enc3, Key: 3, &dec3);
19
20    std::cout << enc1 << std::endl;
21    std::cout << enc2 << std::endl;
22    std::cout << enc3 << std::endl << std::endl;
23
24    std::cout << dec1 << std::endl;
25    std::cout << dec2 << std::endl;
26    std::cout << dec3 << std::endl << std::endl;
```

Figure 3.7: main.cpp

Часть шифра Атбаша:

```
28 // Atbash part
29 enc1 = "", enc2 = "", enc3 = "";
30 dec1 = "", dec2 = "", dec3 = "";
31
32 CipherAtbash::cipher(msg1, &enc1);
33 CipherAtbash::cipher(msg2, &enc2);
34 CipherAtbash::cipher(msg3, &enc3);
35
36 std::cout << enc1 << std::endl;
37 std::cout << enc2 << std::endl;
38 std::cout << enc3 << std::endl << std::endl;
39
40 CipherAtbash::decipher(enc1, &dec1);
41 CipherAtbash::decipher(enc2, &dec2);
42 CipherAtbash::decipher(enc3, &dec3);
43
44 std::cout << dec1 << std::endl;
45 std::cout << dec2 << std::endl;
46 std::cout << dec3 << std::endl << std::endl;
47 }
```

Figure 3.8: main.cpp

Результаты тестов. Первые три строки это зашифрованные сообщения шифром Цезаря. Следующие три строки, это расшифрованные сообщения. Следующие три строки, это те же сообщения, но зашифрованные шифром Атбаш. И последние три строки - расшифрованные сообщения.

```
/home/pi/education/pfur_masters/mat0snovyInfBez/Labs/lab01/cmake-build-debug/main
KH00RGZRU0G
LC0LNHCFDWV
LQIEHBCNDLI

HELLO WORLD
I LIKE CATS
INFBEZ KAIF

TWPPMAEKJPX
SAPSQWAY HI
SNVZWBAQ SV

HELLO WORLD
I LIKE CATS
INFBEZ KAIF
```

Figure 3.9: tests

4 Выводы

Освоил на практике применение методов шифрования Цезаря и Атбаша.