

Министерство науки и высшего образования РФ
ФГБОУ ВО ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
Кафедра «Информационная безопасность систем и технологий»

ОТЧЁТ

о лабораторной работе №1

Создание многомодульных проектов на языке C++

Дисциплина: Технологии и методы
программирования

Группа: 18ПИ1

Выполнил: Асаян А.В.

Количество баллов:

Дата сдачи:

Проверил: к.т.н., доцент Лупанов М.Ю.

Пенза, 2019

1 Цель работы

1.1 Освоить процесс создания многомодульных проектов.

2 Задания к практической работе

2.1 Модифицировать рассмотренную в качестве примера программу для работы с текстами на русском языке.

2.2 Разработать многомодульную программу, реализующую шифр табличной маршрутной перестановки. В качестве ключа взять количество столбцов таблицы. Маршрут записи: по горизонтали слева направо, сверху вниз. Маршрут считывания: сверху вниз, справа налево.

2.2.1 Разработать UML-диаграмму вариантов использования.

2.2.2 Спроектировать шифр табличной маршрутной перестановки в виде класса и построить для него диаграмму классов. Установку ключа выполнять в конструкторе.

2.2.3 Разработать диаграммы деятельности для методов зашифрования и расшифрования.

2.2.4 Реализовать шифр табличной маршрутной перестановки в виде отдельного модуля.

2.2.5 Спроектировать пользовательский интерфейс программы и реализовать его в главном модуле.

3 Результат выполнения работы

3.1 Программа, данная в качестве примера в методических указаниях к лабораторной работе, была модифицирована для работы с русским языком. Для этого все переменные типа `string` были заменены на переменные типа `wstring`, значение атрибута `numAlpha` класса `modAlphaCipher` было изменено на строку, представляющую собой последовательность всех букв русского алфавита. Была подключена русская локаль. На рисунке 1 представлены результаты работы программы для ключа «ЭХО» и шифруемого слова «ПРИВЕТ». Содержимое заголовочного файла `modAlphaCipher.h`:

```
#pragma once
#include <vector>
#include <string>
#include <map>
#include <locale>
```

```

class modAlphaCipher
{
private:
    std::wstring numAlpha =
        L"АБВГДЕЁЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯ";
    //алфавит по порядку
    std::map <wchar_t,int> alphaNum;
    //ассоциативный массив "номер по символу"
    std::vector <int> key;
    //ключ
    std::vector<int> convert(const std::wstring&
s); //преобразование строка-вектор
    std::wstring convert(const std::vector<int>&
v); //преобразование вектор-строка
public:
    modAlphaCipher()=delete;
    //запретим конструктор без параметров
    modAlphaCipher(const std::wstring& skey);
    //конструктор для установки ключа
    std::wstring encrypt(const std::wstring&
open_text); //зашифрование
    std::wstring decrypt(const std::wstring&
cipher_text); //расшифрование
};

```

Содержимое файла modAlphaCipher.cpp:

```

#include "modAlphaCipher.h"
modAlphaCipher::modAlphaCipher(const std::wstring&
skey)
{
    for (unsigned i=0; i<numAlpha.size(); i++) {
        alphaNum[numAlpha[i]]=i;
    }
    key = convert(skey);
}
std::wstring modAlphaCipher::encrypt(const
std::wstring& open_text)
{
    std::vector<int> work = convert(open_text);
    for(unsigned i=0; i < work.size(); i++) {
        work[i] = (work[i] + key[i % key.size()]) %
alphaNum.size();
    }
    return convert(work);
}

```

```

    }
    std::wstring modAlphaCipher::decrypt(const
std::wstring& cipher_text)
    {
        std::vector<int> work = convert(cipher_text);
        for(unsigned i=0; i < work.size(); i++) {
            work[i] = (work[i] + alphaNum.size() - key[i
% key.size()]) % alphaNum.size();
        }
        return convert(work);
    }
    inline std::vector<int> modAlphaCipher::convert(const
std::wstring& s)
    {
        std::vector<int> result;
        for(auto c:s) {
            result.push_back(alphaNum[c]);
        }
        return result;
    }
    inline std::wstring modAlphaCipher::convert(const
std::vector<int>& v)
    {
        std::wstring result;
        for(auto i:v) {
            result.push_back(numAlpha[i]);
        }
        return result;
    }

```

Содержимое файла main.cpp:

```

#include <iostream>
#include <cctype>
#include "modAlphaCipher.h"
using namespace std;
// проверка, чтобы строка состояла только из
прописных букв
bool isValid(const wstring& s){
    std::locale loc("ru_RU.UTF-8");
    std::locale::global(loc);
    for(auto c:s)
        if ((c<L'A' || c>L'Я') and c!=L'Ё')
            return false;
    return true;
}

```

```

    }
    int main(int argc, char **argv)
    {std::locale loc("ru_RU.UTF-8");
    std::locale::global(loc);
        wstring key;
        wstring text;
        unsigned op;
        wcout<<"Cipher ready. Input key: ";
        wcin>>key;
        wcout<<key<<endl;
        if (!isValid(key)) {
            wcerr<<"key not valid\n";
            return 1;
        }
        wcout<<"Key loaded\n";
        modAlphaCipher cipher(key);
        do {
            wcout<<"Cipher ready. Input operation (0-
exit, 1-encrypt, 2-decrypt): ";
            wcin>>op;
            if (op > 2) {
                wcout<<"Illegal operation\n";
            } else if (op > 0) {
                wcout<<"Cipher ready. Input text: ";
                wcin>>text;
                if (isValid(text)) {
                    if (op==1) {
                        wcout<<"Encrypted text:
"<<cipher.encrypt(text)<<endl;
                    } else {
                        wcout<<"Decrypted text:
"<<cipher.decrypt(text)<<endl;
                    }
                } else {
                    wcout<<"Operation aborted: invalid
text\n";
                }
            }
        } while (op!=0);
        return 0;
    }

```

```
Терминал
Cipher ready. Input key: ЭХО
ЭХО
Key loaded
Cipher ready. Input operation (0-exit, 1-encrypt, 2-decrypt): 1
Cipher ready. Input text: ПРИВЕТ
Encrypted text: МЁЧЯЬБ
Cipher ready. Input operation (0-exit, 1-encrypt, 2-decrypt): 2
Cipher ready. Input text: МЁЧЯЬБ
Decrypted text: ПРИВЕТ
Cipher ready. Input operation (0-exit, 1-encrypt, 2-decrypt): █
```

Рисунок 1 — Результат работы шифра Гронсфельда.

2.2 Была разработана многомодульная программа, реализующая шифр маршрутной табличной перестановки, состоящая из главного модуля main.cpp, дополнительного Perestанovka.cpp и заголовочного файла Perestанovka.h.

2.2.1 Была построена диаграмма вариантов использования, представленная на рисунке 2.

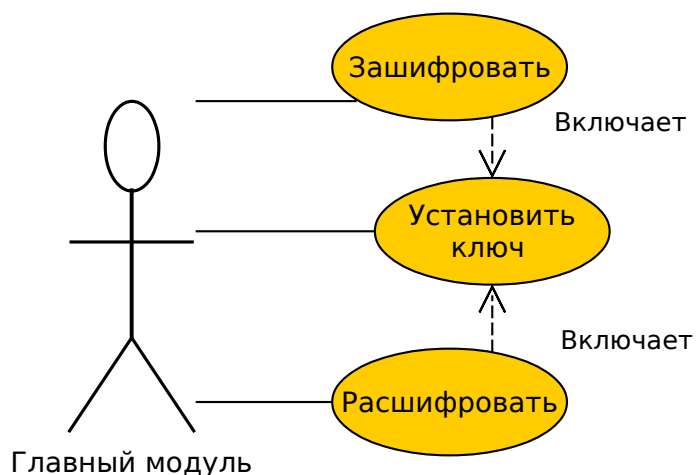


Рисунок 2 — Диаграмма вариантов использования.

2.2.2 Шифр маршрутной перестановки был реализован в виде класса. Объявление класса, его методов и атрибутов происходит в заголовочном

файле Perestankovka.h, методы описываются в файле Perestankovka.cpp.

Содержимое файла Perestankovka.h:

```
#pragma once
#include <string>
class Perestankovka{
private:
    int k;
public:
    Perestankovka()=delete;
    Perestankovka(const int k);
    std::string shifr(const std::string& m);
    std::string rashifr(const std::string& m);
};
```

Диаграмма классов представлена на рисунке 3.

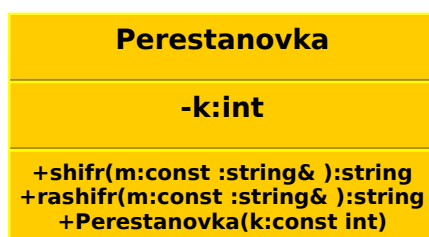


Рисунок 3 — Диаграмма классов.

2.2.3 Была разработана диаграмма деятельности для методов расшифрования и зашифрования. Диаграмма представлена на рисунке 4.

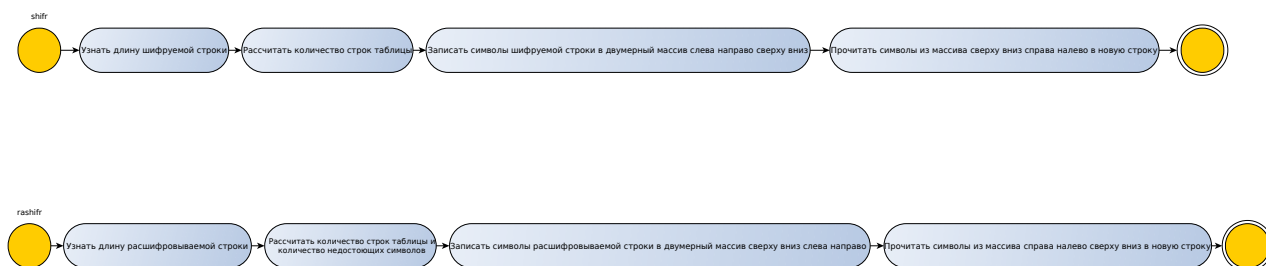


Рисунок 4 — Диаграмма деятельности.

2.2.4 Шифр маршрутной табличной перестановки был реализован в виде отдельного модуля. Файл Perestankovka.cpp содержит методы класса Perestankovka. Содержимое класса Perestankovka.cpp :

Содержимое файла Perestankovka.cpp:

```
#include "Perestankovka.h"
```

```

#include <iostream>
Perestanovka::Perestanovka(const int key)
{
    k=key;
}
std::string Perestanovka::shifr(const std::string& m)
{
    std::string chipher_text;
    chipher_text.clear();
    int sim=0;
    sim=m.length();
    int strok=(sim-1)/k + 1;
    char **masivl= new char* [strok];
    for (int i = 0; i < strok; i++) {
        masivl[i] = new char [k];
    }
    for (int i=0; i<strok; i++) {
        int p=i*k;
        for (int j=0; j<k; j++) {
            if(p+j<sim and m[p+j]>='A' and
m[p+j]<='Z')
                masivl[i][j]=m[p+j];
            else {
                masivl[i][j]='*';
            }
        }
        for (int j=k-1; j>=0; j--) {
            for (int i=0; i<strok; i++)
                if(masivl[i][j]!='*')
                    chipher_text=chipher_text+masivl[i]
[j];
        }
        for (int i = 0; i<strok; i++)
            delete[] masivl[i];
        delete[] masivl;
        return chipher_text;
    }
std::string Perestanovka::rashifr(const std::string&
m)
{
    std::string chipher_text;
    std::string temp;
    temp.clear();

```



```

chipher_text.clear();
int sim=0;
sim=m.length();
for(int i=sim-1; i>=0; i--)
    temp=temp+m[i];
int strok=(sim-1)/k + 1;
int d=k*strok-sim;
char **masiv1= new char* [strok];
for (int i = 0; i < strok; i++) {
    masiv1[i] = new char [k];
}
int p=0;
int t=0;
for (int j=0; j<k; j++) {
    p=j*strok;
    for (int i=0; i<strok; i++) {
        if(i==0 and j>k-1-d) {
            masiv1[i][j]='*';
            t++;
        } else
            masiv1[i][j]=temp[p+i-t];
    }
}
for (int i=strok-1; i>=0; i--) {
    for (int j=0; j<k; j++)
        if(masiv1[i][j]!='*')
            chipher_text=chipher_text+masiv1[i]
[j];
}
for (int i = 0; i<strok; i++)
    delete[] masiv1[i];
delete[] masiv1;
return chipher_text;
}

```

2.2.5 Пользовательский интерфейс программы был спроектирован в главном модуле. Содержание файла main.cpp:

```

#include <iostream>
#include "Perestанovka.h"
using namespace std;
int main(int argc, char **argv)
{
    string s;
    int key;

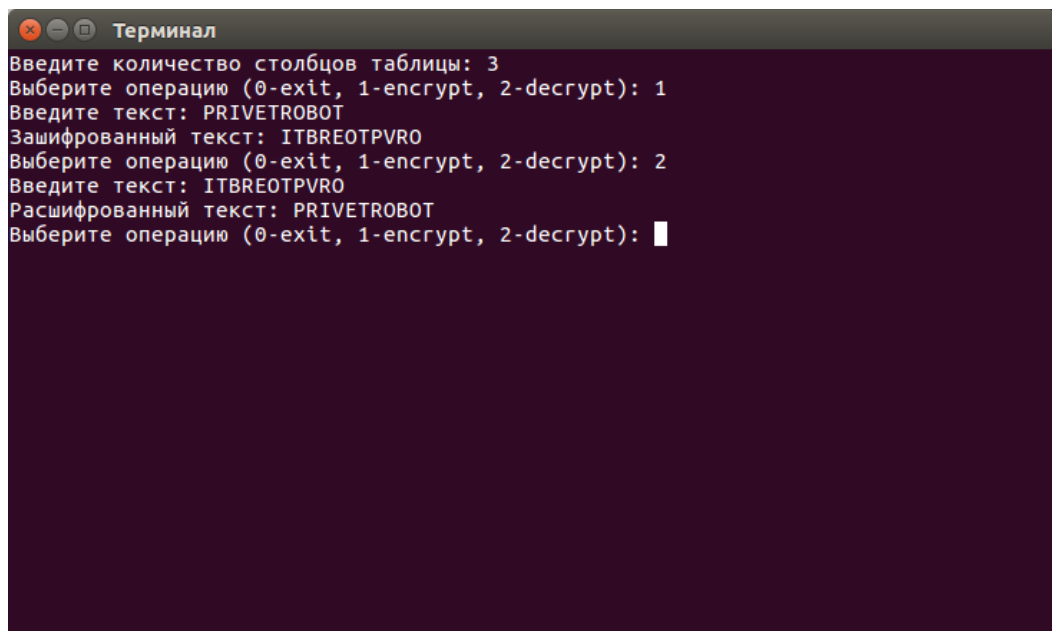
```

```

int op;
cout<<"Введите количество столбцов таблицы: ";
cin>>key;
Perestанovka chipher(key);
do {
    cout<<"Выберите операцию (0-exit, 1-encrypt,
2-decrypt): ";
    cin>>op;
    if (op > 2) {
        cout<<"Illegal operation\n";
    } else if (op >0) {
        cout<<"Введите текст: ";
        cin>>s;
        if (op==1) {
            cout<<"Зашифрованный текст:
"<<chipher.shifr(s)<<endl;
        } else {
            cout<<"Расшифрованный текст:
"<<chipher.rashifr(s)<<endl;
        }
    }
} while (op!=0);
}

```

На рисунке 5 представлены результаты работы программы для строки «PRIVETROBOT» и ключе 3.



```

Терминал
Введите количество столбцов таблицы: 3
Выберите операцию (0-exit, 1-encrypt, 2-decrypt): 1
Введите текст: PRIVETROBOT
Зашифрованный текст: ITBREOTPVRO
Выберите операцию (0-exit, 1-encrypt, 2-decrypt): 2
Введите текст: ITBREOTPVRO
Расшифрованный текст: PRIVETROBOT
Выберите операцию (0-exit, 1-encrypt, 2-decrypt): █

```

Рисунок 5 — Результат работы шифра маршрутной табличной перестановки.

4. Вывод

В процессе выполнения лабораторной работы были изучены возможности языка C++ по созданию многомодульных программа, были освоено подключение заголовочных файлов, были получены практические навыки по созданию UML-диаграмм и созданию модулей.