|  |
| --- |
| Міністерство освіти і науки, молоді та спорту України  Тернопільський національний технічний університет імені Івана Пулюя |
| (повне найменування вищого навчального закладу) |
| **кафедра комп’ютерних наук** |
| (повна назва кафедри) |

**КУРСОВА РОБОТА**

|  |  |  |
| --- | --- | --- |
| з | **Об’єктно-орієнтованого програмування** | |
| (назва дисципліни) | | |
| на тему: | | **«Захист тексту за допомогою шифрування»** |
|  | | |
|  | | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Студент (а) | | 2 | | | | | курсу, | | | групи | СН-21 | |
| напряму підготовки | | | | | | 6.050101 | | | | | | |
|  | | | | | | | | | | | | |
| спеціальності | | | «Комп’ютерні науки» | | | | | | | | | |
|  | | | | | | | | | | | | |
| Рожко Е.С. | | | | | | | | | | | | |
| (прізвище та ініціали) | | | | | | | | | | | | |
| Керівник: | к.т.н., доцент Бревус В.М. | | | | | | | | | | | |
|  | | | | | | | | | | | | |
| (посада, вчене звання, науковий ступінь, прізвище та ініціали) | | | | | | | | | | | | |
| Оцінка за національною шкалою | | | | | | | | |  | | | |
| Кількість балів: | | | |  | | | | Оцінка ECTS | | | |  |
| Члени комісії: |  | | | |  | Бревус В.М. | | | | | | | |
| (підпис) | | | |  | (прізвище та ініціали) | | | | | | | |
|  | | | |  | Баран І.О. | | | | | | | |
| (підпис) | | | |  | (прізвище та ініціали) | | | | | | | |
|  | | | |  |  | | | | | | | |
| (підпис) | | | |  | (прізвище та ініціали) | | | | | | | |

м. Тернопіль – 2015

**Код програми**

**Main.cpp**

#include <iostream>

#include "text.h" // Підключення класу 'Text'-- тексту із розшириними властивостями

#include "MyFunct.h" // Бібліотека із деякими моїми функціями

#include "RSA.h"

#include <conio.h>

using namespace std;

int main (int argc, char\* argv[])

{

Text new\_text(" - empty - \n");

MyFile data\_file("temp.dat", "rt");

MyFile settings("settings.keys","rt+");

RSA rsa( new\_text.get\_data(), 13, 17);

MainMenu(new\_text, data\_file, settings);

\_getch();

return 0;

}

**FileFunct.h**

#ifndef \_FileFunct\_H\_

#define \_FileFinct\_H\_

#include <string>

#include <fstream>

using namespace std;

class MyFile

{

protected:

FILE \*mf;

string standartName; // Ім'я файла

string mode; // Режим відкриття файла

public:

MyFile(string Name = "temp.dat", string Mode = "rt")

{

standartName = Name;

mode = Mode;

}

void CheckName(string& tempName)

{

tempName.c\_str();

unsigned int i;

for (i=0; i<tempName.length(); i++)

{

if (tempName[i] == ' ' ) tempName[i] = '\_';

}

}

bool FindFIle()

{

if ((mf = fopen( standartName.c\_str() , "rt" )) == NULL)

return true;

else return false;

}

string NameOfFile()

{

setlocale(LC\_ALL, "ukr");

cout << "\n Введiть назву файлу ( назва.розширення )"

<< "\n (Нажмiть Enter для залиження стандартного iменi \""

<< standartName << "\" ) : \n\t";

string \_NameOfFile("");

getline(cin, \_NameOfFile);

if (\_NameOfFile == "") \_NameOfFile = standartName;

CheckName(\_NameOfFile);

return \_NameOfFile;

}

string loadFile(int k)

{

setlocale(LC\_ALL, "urk");

while (!FindFIle())

{

cout << "\n\n Файл \"" << standartName <<"\" не знайдено! \n";

rename();

}

cout << "\n Почитається завантеження даних з файлу ... \n";

ifstream in( standartName.c\_str() );

string tempData;

in >> tempData;

in.close();

cout << "\n Данi завантажено! \n";

return tempData;

}

vector<unsigned long> loadFile(string temp = "")

{

freopen( standartName.c\_str(), "rt", stdin);

getline(cin,temp);

vector<unsigned long> temp2;

for(int i=0; i<temp.length(); i++)

cin >> temp2[i];

freopen(NULL, "rt",stdin);

return temp2;

}

int saveFile(Text& text, string atrubute = "auto")

{

if (atrubute != "auto" ) rename();

ofstream out( standartName.c\_str() );

out << text.get\_data()<<"\n";

for(int i=0;i<text.get\_data().length(); i++)

out << text.getCrypted()[i] << " ";

setlocale(LC\_ALL, "ukr");

cout << "\n Данi збереженi в файл \n";

return 0;

}

int saveFile(long temp, string atrubute = "auto")

{

if (atrubute != "auto" ) rename();

ofstream out( standartName.c\_str() );

out << temp << " ";

return 0;

}

void rename()

{

string temp = "ren " + standartName + " " ;

standartName = NameOfFile();

temp +=standartName;

system( temp.c\_str() );

}

int hideFile()

{

string temp("attrib -h ");

temp += standartName;

return system( temp.c\_str() );

}

};

#endif

**RSA.h**

#ifndef RSA\_H

#define RSA\_H

#include <iostream>

#include <string>

#include <cstdlib> // Для rand()

#include <time.h> // Для time

#include <cmath>

#include <vector>

using namespace std;

class RSA

{

private:

string Text;

vector<unsigned long> Crypted;

string CryptedText;

unsigned int d;

unsigned int e;

int p\_simple;

int q\_simple;

unsigned int p2;

unsigned int Fn;

unsigned int n;

public:

RSA(string \_data = "")

{

Text = \_data;

generateKeys();

}

RSA(string \_data, int openK, int secretK, int \_N, int \_Fn)

{

Text = \_data;

e = openK;

d = secretK;

n = \_N;

Fn = \_Fn;

}

RSA(string \_data, int P1, int P2)

{

Text = \_data;

generateKeys(P1,P2);

}

// Методи-аксесори {

void setSecretKey(int sk)

{

d = sk;

}

int getSecretKey()

{

return d;

}

void setOpentKey(int ok)

{

e = ok;

}

int getOpentKey()

{

return e;

}

int getN()

{

return n;

}

void setFN(unsigned int fn)

{

Fn = fn;

}

void setN(unsigned int \_n)

{

n = \_n;

}

unsigned int getFN()

{

return Fn;

}

string getText()

{

return Text;

}

void setText(string \_text)

{

Text = \_text;

}

void setCryptedText(string CryptedData)

{

CryptedText = CryptedData;

}

string getCryptedText()

{

return CryptedText;

}

vector<unsigned long> getCrypted()

{

return Crypted;

}

void setCrepted(vector<unsigned long> crypted)

{

Crypted = crypted;

}

// } Методи-аксесори

// Методи для генерації ключів

int sundaram(int n) // Алгоритм "Решето Сундарама"

{

int \*a = new int [n], i, j, k;

memset(a, 0, sizeof(int) \* n);

for(i = 1; 3\*i+1 < n; i++)

{

for(j = 1; (k = i+j+2\*i\*j) < n && j <= i; j++)

a[k] = 1;

}

// Повертання найближчого числа до заданого

for(i = n-1; i >= 1; i--)

if(a[i] == 0)

{

return (2 \* i + 1);

break;

}

delete [] a;

}

void generateKeys(int p = 0, int q = 0)

{

srand( (unsigned)time( NULL ) );

if (!p && !q)

{

p = rand()%100;

q = rand()%100;

}

p\_simple = sundaram(p);

q\_simple = sundaram(q);

//Находимо число n.

n = p\_simple\*q\_simple;

//Генерація числа d і провірка його на взаємопростоту

//з числом ((p\_simple-1)\*(q\_simple-1)).

int d\_simple = 0;

while (d\_simple !=1)

{

d = rand()%100;

d\_simple = gcd (d, ((p\_simple-1)\*(q\_simple-1)));

}

//Визначення числа e, для якого є істинним

//відношення (e\*d)%((p\_simple-1)\*(q\_simple-1))=1.

unsigned int e\_simple = 0;

e = 0;

while (e\_simple !=1)

{

e += 1;

e\_simple = (e\*d)%((p\_simple-1)\*(q\_simple-1));

}

}

int gcd(int a, int b) // Алгоритм Евклида

{

int c;

while (b)

{

c = a % b;

a = b;

b = c;

}

return abs(a);

}

// --------------------------

// Методи шифрування - дишифрування

string encrypt(string \_data = "") // Метод шифрування

{

if (\_data != "")

Text = \_data;

Text.c\_str();

Crypted.resize( Text.length());

unsigned long c, b =301;

for (int j = 0; j < Text.length(); j++)

{

c = 1;

unsigned int i = 0;

unsigned int ASCIIcode = (static\_cast<unsigned int>(Text[j]))+b;

while (i<e)

{

c = c\*ASCIIcode;

c = c%n;

i++;

}

Crypted[j] = c;

CryptedText.push\_back(static\_cast<char>(c));

b+=1;

}

return CryptedText;

}

string unecrypt(string \_data = "") // Метод для розшифрування

{

Text.c\_str();

int b = 301;

int m ;

for (int j = 0; j < Text.length(); j++)

{

m=1;

unsigned int i = 0;

while (i<d)

{

m = m \* Crypted[j];

m = m%n;

i++;

}

m = m-b;

b+=1;

Text[j] = static\_cast<char>(m);

}

return Text;

}

//---------------------------------

};

#endif

MyFunct.h

#ifndef \_MyFunct\_H\_

#define \_MyFunct\_H\_

#include <string>

#include <conio.h>

#include <windows.h>

#include "FileFunct.h"

#include "RSA.h"

// Прототипи функцiй

void printLine(); // Функція виводить стмвол '|' 78 раз, малює умовну лінію

void exitProg(Text& t, MyFile& f); // Функція реалізує завершення програми і зберігання дані у файли

void outputText(Text& text); // Функція, яка виводить інформацію про введені дані (текст)

void encrypt(Text& text); // Функція шифрування тексту

void unencrypt(Text& text); // Функція дешифрування тексту

void saveSattings(Text& t); // Функція зберігає ключі в файл

void loadFile(Text& text,MyFile& file); // Фукнція для завантеження даних (тексту) з файлу

// Опис функцiй

void MainMenu(Text& \_text, MyFile& \_file, MyFile& sett) // Функція головного меню

{

char answ = '0';

setlocale(LC\_ALL, "ukr");

do

{

system("color 1f");

cout << "\t\t ===== Виберiть комаду ===== \n"

<< "\n\t 1 - Вивести данi"

<< "\n\n\t 2 - Зашифрувати данi (текст)"

<< "\n\t 3 - Розшифрувати данi (текст)"

<< "\n\n\t 4 - Зберегти данi у файл"

<< "\n\t 5 - Загрузити данi з файлу"

<< "\n\n\t 0 - Вихiд \n";

answ = \_getch();

switch(answ)

{

case '1':

{

printLine();

outputText(\_text);

printLine();

\_getch();break;

}

case '2':

{

printLine();

encrypt(\_text);

printLine();

\_getch();break;

}

case '3':

{

printLine();

unencrypt(\_text);

printLine();

\_getch();break;

}

case '4':

{

printLine();

printLine();

\_getch();break;

}

case '5':

{

printLine();

loadFile(\_text,\_file);

printLine();

\_getch();break;

}

case '0':

{

printLine();

exitProg( \_text, \_file);

return;

}

}

system("cls");

} while (1);

}

void saveFile(Text& text, MyFile& file)

{

file.saveFile( text, "");

}

void printLine()

{

int i=0;

cout << "\n ";

while(i++ < 78)

cout << "|";

cout << " \n";

}

void exitProg(Text& t, MyFile& f)

{

f.saveFile(t);

saveSattings(t);

cout<< "\n Exit ... \n";

exit(1);

}

void encrypt(Text& text)

{

setlocale(LC\_ALL, "ukr");

if (text.show\_status()) cout << "\n Данi уже зашифроманi!\n";

else

{

cout << "\n Веведiть данi для шифрування : " ;

string temp;

getline(cin,temp);

text.set\_data(temp);

RSA rsa ( text.get\_data());

rsa.encrypt (temp);

text.setCrypted( rsa.getCrypted() );

text.setCryptedData( rsa.getCryptedText() );

text.setOpenKey( rsa.getOpentKey());

text.setSecretKey( rsa.getSecretKey());

text.setN( rsa.getN());

text.setFN( rsa.getFN());

text.change\_status();

}

cout <<"\n\n Результат : " << text.getCryptedData();

}

void outputText(Text& text)

{

cout << "\n Данi : \"" << text.get\_data();

if (!text.show\_status()) cout << "\" не";

cout << " зашифрованi!"

<< "\n E = " << text.getOpenKey()

<< "\n N = " << text.getN() << endl;

}

void saveSattings(Text& t)

{

MyFile settings("settings.key","wt");

settings.saveFile(t.getOpenKey());

settings.saveFile(t.getSecretKey());

settings.saveFile(t.getN());

settings.saveFile(t.getFN());

if (!settings.hideFile()) cout << "\n Ключi збережено \n";

}

void loadFile(Text& text, MyFile& file)

{

text.setCrypted( file.loadFile("") ) ;

MyFile settings("settings.key","rt");

string temp ("");

temp += settings.loadFile(0);

string temp2 ("");

int k=0;

for(int i=0; i<temp.length(); i++)

{

if (temp[i]!=' ')

temp2 += temp[i];

else

{

k++;

if (k==1) text.setOpenKey (atoi(temp2.c\_str()) );

if (k==2) text.setSecretKey (atoi(temp2.c\_str()) );

if (k==3) text.setN (atoi(temp2.c\_str()) );

if (k==4) text.setFN(atoi(temp2.c\_str()));

temp2 = "";

}

}

if (text.getN() && !text.show\_status()) text.change\_status();

}

void unencrypt(Text& text)

{

if (!text.show\_status()) cout << "\n Данi не зашифрованi!\n";

else

{

cout<<"\n Данi розшифровуються ... \n";

RSA rsa ( text.getCryptedData(), text.getOpenKey(), text.getSecretKey(), text.getN(), text.getFN());

rsa.setCrepted ( text.getCrypted());

text.set\_data(rsa.unecrypt());

text.change\_status();

cout <<"\n\n Результат : " <<

text.get\_data();

}

}

#endif

Text.h

#ifndef \_TEXT\_H\_

#define \_TEXT\_H\_

#include <iostream>

#include <cstring>

#include <vector>

using namespace std;

// Клас тексту з розшириними властивостями

class Text

{

private:

string Data;

string CryptedData;

vector<unsigned long> Crypted;

unsigned int openKey;

unsigned int secretKey;

unsigned int N;

unsigned int Fn;

bool crypted;

public:

Text( string new\_data = "",unsigned int new\_OKey = 0,unsigned int new\_SKey = 0, bool new\_h = 0)

{

openKey = new\_OKey;

secretKey = new\_SKey;

Data = new\_data;

crypted = new\_h;

Crypted.resize( Data.length());

}

void insert(int poshition, string \_temp )

{

Data.insert( poshition, \_temp) ;

}

void rewrite( string \_temp )

{

Data = \_temp;

}

void push\_back( string \_temp )

{

Data += \_temp;

}

void setCrypted(vector<unsigned long> crypted)

{

Crypted = crypted;

}

vector<unsigned long> getCrypted()

{

return Crypted;

}

unsigned int getFN()

{

return Fn;

}

void setFN(unsigned int fn)

{

Fn = fn;

}

void setN(unsigned int n)

{

N = n;

}

unsigned int getN()

{

return N;

}

bool show\_status()

{

if (crypted) return true;

else return false;

}

void change\_status()

{

if (crypted) crypted = 0;

else crypted = 1;

}

void set\_data( string new\_data )

{

Data = new\_data;

}

void setOpenKey( unsigned int new\_key )

{

openKey = new\_key;

}

void setSecretKey(unsigned int new\_key)

{

secretKey = new\_key;

}

unsigned int getOpenKey()

{

return openKey;

}

unsigned int getSecretKey()

{

return secretKey;

}

string get\_data()

{

return Data;

}

void set\_hided(bool temp)

{

crypted = temp;

}

void setCryptedData(string newData)

{

CryptedData = newData;

}

string getCryptedData()

{

return CryptedData;

}

};

#endif