**Курс «Объектно-ориентированное программирование на C++»**

Лабораторное задание

**Задание 1:**

Создайте приложение «Валидатор HTML-файлов». Приложение запрашивает у пользователя путь к файлу, после чего проверяет файл на валидность. Критерием валидности для вашего проекта являются правила: все открытые теги должны закрываться, если был знак < его должен закрыть знак >. По итогам валидации нужно отобразить результат проверки на экран. Используйте для работы с файлами потоки, а для работы со строками класс string.

Примеры:

**Содержимое валидного файла**

<html>

<body>To be or not to be</body>

</html>

**Содержимое невалидного файла**

<html>

<body To be or not to be</body>

</html>

Завдання

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <iostream>

#include<Windows.h>

#include<string>

#include<fstream>

#include<stack>

using namespace std;

bool Valid\_or\_invalid(const string txt)

{

bool b = true;

stack<char>sign;

char\* copy = new char[txt.size() + 1];

strcpy(copy, txt.c\_str());

for (int i = 0; i <= strlen(copy); i++)

{

if (copy[i] == '<')

sign.push(copy[i]);

if (copy[i] == '>')

{

if (sign.empty())

{

b = false;

break;

}

sign.pop();

}

}

if (!sign.empty())

{

delete[]copy;

return false;

}

if (b)

{

for (int i = 0; i <= strlen(copy); i++)

{

if (copy[i] == '<'&& copy[i+1] !='/')

sign.push('1');

if (copy[i] == '<' && copy[i + 1] == '/')

{

if (sign.empty())

{

b = false;

break;

}

sign.pop();

}

}

if (!sign.empty())

{

delete[]copy;

return false;

}

}

delete[]copy;

return b;

}

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

string place,txt, tmp;

fstream f;

cout << "->";

getline(cin, place);

place.append(".txt");

f.open(place,ios::in);

while (!f.eof())

{

f >> tmp;

cout << tmp << " ";

txt.append(tmp);

}

cout << endl;

cout << "-----------------------------\n";

if (Valid\_or\_invalid(txt))

cout << "Valid\n";

else

cout << "Invalid\n";

f.close();

system("pause");

return 0;

}