Міністерство освіти і науки України

Харківський національний університет радіоелектроніки

Кафедра Інформатики

Звіт  
з лабораторної роботи №6  
з дисципліни «Об’єктно-орієнтоване програмування»

Варіант 12

Виконав:  
ст. гр. ІТІНФ-21-1  
Кошель В.О.

Перевірили:  
Кириченко І.Ю.

1. **Асоціативні контейнери**
   1. **Мета роботи:**

Ознайомитись з такими поняттями, як асоціативні контейнери стандартної бібліотеки STL. Отримати навички роботи з ними.

* 1. **Завдання:**

Розробити програму, яка зчитує текст і зберігає номери всіх рядків, в яких зустрілося дане слово, за винятком слів, зазначених у списку.

Слова в тексті розділяються пробілами.

Накопичення номерів рядків реалізовувати за допомогою асоціативного масиву, список слів для виключення реалізовувати за допомогою множини.

Кожен раз, коли зустрілося слово зі списку, виводити повідомлення.

Якщо слово зустрілося в одному рядку двічі - виводити повідомлення.

Створити набір функцій для виведення результатів і винести їх в окремий файл.

Оголошення функцій винести в заголовочний файл. Вихідні дані зчитати з файлу і вивести в файл

* 1. **Вхідні дані (***input.txt***)**

уж небо осенью дышало дышало

уж реже солнышко блистало

короче становился день

лесов таинственная сень

* 1. **Слова-винятки (***excepts.txt***)**

день,осенью,лесов,короче,Пушкин

* 1. **Хід виконання:**

Файл Source.cpp

#include <iostream>

#include <fstream>

#include <sstream>

#include <string>

#include <vector>

#include <map>

#include <set>

#include <algorithm>

#include "funcs.h"

using namespace std;

int main()

{

setlocale(LC\_ALL, "Russian");

ofstream f\_out("output.txt");

map<string, multiset<size\_t>> result; // Асоціативний контейнер результатів

set<string> excepts; // Множина слів-винятків

/\*

Зчитування слів-винятків

\*/

ifstream f\_excepts("excepts.txt");

while (!f\_excepts.eof())

{

string tmp;

getline(f\_excepts, tmp, ',');

excepts.emplace(tmp);

}

f\_excepts.close();

/\*

Зчитування тексту

\*/

ifstream f\_text("input.txt");

string tmp;

for (size\_t i = 1; !f\_text.eof(); i++)

{

getline(f\_text, tmp, '\n');

istringstream tmp\_stream(tmp);

while (!tmp\_stream.eof())

{

string tmp\_string;

getline(tmp\_stream, tmp\_string, ' ');

// Якщо знайдене слово не належить до множини винятків,

// то додаємо до масиву пару <слово, рядок>

if (excepts.find(tmp\_string) == excepts.end())

result[tmp\_string].emplace(i);

// Кожен раз, коли зустрілося слово зі списку, виводити повідомлення

else

Out::is\_in\_exception\_list(f\_out, tmp\_string, i);

}

}

map<string, set<size\_t>> how\_many\_double;

for (auto elem : result)

{

for (size\_t i : elem.second)

{

size\_t count\_double = count(elem.second.begin(), elem.second.end(), i);

if (count\_double == 2)

how\_many\_double[elem.first].emplace(i);

}

}

// Якщо слово зустрілося в одному рядку двічі - виводити повідомлення

cout << "\nДва раза:\n";

f\_out << "\nДва раза:\n";

for (auto elem : how\_many\_double)

Out::is\_double(f\_out, elem.first, elem.second);

return 0;

}

Файл funcs.h

#pragma once

#include <iostream>

#include <string>

#include <fstream>

#include <set>

using namespace std;

namespace Out

{

void is\_in\_exception\_list(ofstream& file, string str, size\_t line);

void is\_double(ofstream& file, string word, set<size\_t> lines);

}

Файл funcs.cpp

#include "funcs.h"

void Out::is\_in\_exception\_list(ofstream& file, string str, size\_t line)

{

cout << "\"" << str << "\" находится в списке исключений. В тексте – на " << line << " строке.\n";

file << "\"" << str << "\" находится в списке исключений. В тексте – на " << line << " строке.\n";

}

void Out::is\_double(ofstream& file, string word, set<size\_t> lines)

{

cout << word << " у ";

for (size\_t i : lines)

cout << i << "\t";

cout << "\n";

file << word << " у ";

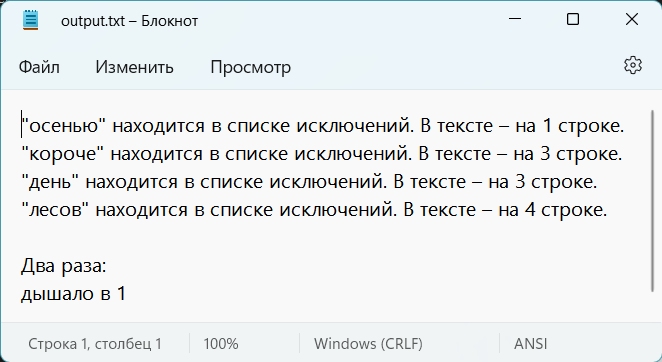
for (size\_t i : lines)

file << i << "\t";

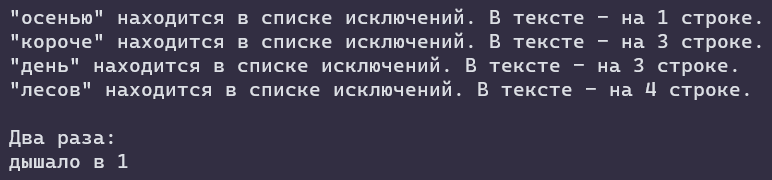
file << "\n";

}

* 1. **Результат виконання**
* **Вихідний файл (***output.txt***):**

****

* **Консоль:**

****

* 1. **Висновок.** В результаті виконання цієї лабораторної роботи ми ознайомилися із асоціативними контейнерами стандартної бібліотеки та навчилися використовувати отримані знання на практиці.