**Міністерство освіти і науки України**

**Національний університет «Львівська політехніка»**

**Кафедра ЕОМ**



Звіт

до лабораторної роботи № 8

з дисципліни «Основи алгоритмізації та програмування, частина 1»

на тему:

«Розробка на С багатофайлових проектів»

Варіант № 3

Виконав:

ст.гр. КІ-17

Жлуктенко Дмитро Миколайович

Прийняв:

асистент Бойко Г.В.

**Львів 2016**

**Мета роботи:**

#### **познайомитися із засадами розробки багатофайлових проектів в мові програмування С;**

#### **Теоретичний вступ:**

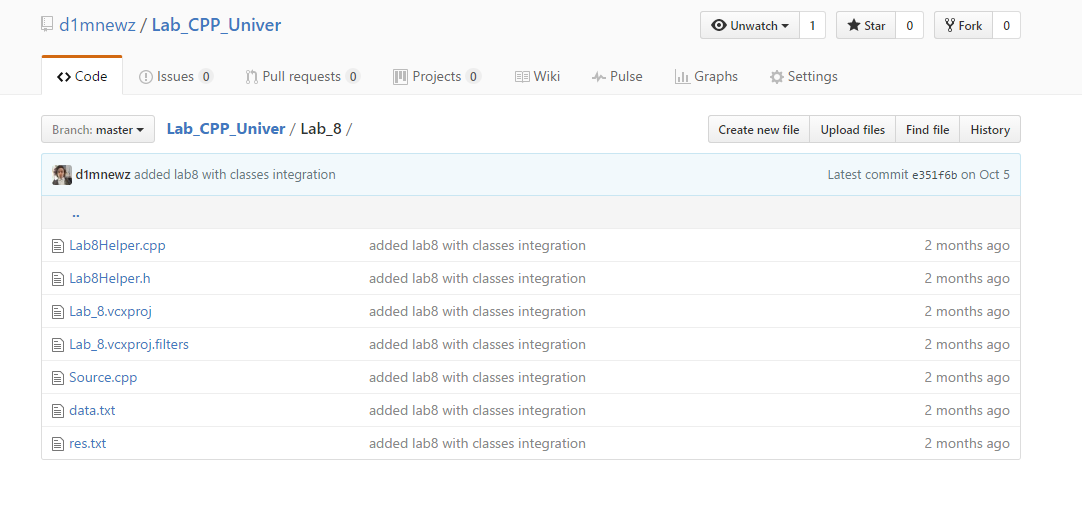
Великі прграми на мові С (особливо ті, які складаються з тисяч, десятків тисяч чи більше рядків вхідного тексту) як правило розбивають на окремі функції, які, в свою чергу, можуть розбиватися на на ще дрібніші функції і так дальше. Таке структурування (при грамотному розбитті) значно спрощує завдання кодування та відлагодження програми. Але відлагодження всіх функцій ніколи не виконується одночасно. Крім того, зберігати всі функції програми в одному файлі незручно і недоцільно. Тому системи розробки програм (в тому числі і Visual C++ 6.0) передбачають засоби розробки, в яких окремі функції (чи група функцій) зберігаються в окремих файлах.

**Індивідуальне завдання №1 . (Варіант 9)**

Індивідуальні завдання 9 ті ж самі що і індивідуальні завдання 6 чи індивідуальні завдання 5 (узгодити з викладачем), які виконувалися у лабораторній роботі №5. Відмінність полягає у тому, що обчислення, які вимагає індивідуальне завдання, необхідно оформити у вигляді багатофайлового проекту, якому є декілька файлів з функціями програми та заголовочні файли. Задача, яка задана в індивідуальному завданні має бути реалізована у вигляді окремої функції (чи функцій) і розміщатися в окремому файлі (чи файлах). Функція main має демонструвати працездатність основної задачі. Тобто, задавати вхідні параметри, звертатися до підпрограми тестованої задачі та виводити результати. Якщо задача буде себе по різному вести при різних вхідних параметрах, то функція main має тестувати всі можливі варіанти.

Ввід даних можна виконувати так само як і у лабораторній роботі 7 - із (текстового) файлу .

**Структура проекту:**



**Код програми:**

**LabHelper.h**

#pragma once

#include <iostream>

#include <vector>

#include <fstream>

#include <string>

using namespace std;

class Lab8Helper

{

private:

vector<int> data;

char \* inFile;

char \* outFile;

int lines;

int max;

int count;

public:

void InitData();

void AnalyzeData();

void PrintResults();

Lab8Helper();

Lab8Helper(char \* iFile, char \* oFile);

void PrintData();

~Lab8Helper();

};

#pragma once

**LabHelper.cpp**

#include "Lab8Helper.h"

void Lab8Helper::InitData()

{

ifstream in(inFile);

int tmp = 0;

in >> this->lines; //number of lines

for (int i = 1; i <= this->lines; i++)

{

in >> tmp;

data.push\_back(tmp);

}

in.close();

}

void Lab8Helper::AnalyzeData()

{

for (int i = 0; i < this->lines; i++)

{

if (data[i] > this->max)

{

this->max = this->data[i];

this->count = 1;

}

else if (this->data[i] == this->max)

{

this->count++;

}

}

}

void Lab8Helper::PrintResults()

{

cout << "Max value = " << this->max << "; Count = " << this->count << endl;

}

Lab8Helper::Lab8Helper()

{

}

Lab8Helper::Lab8Helper(char \* iFile, char \* oFile)

{

inFile = iFile;

outFile = oFile;

max = 0;

count = 0;

}

void Lab8Helper::PrintData()

{

for (int i = 0; i < this->data.size(); i++)

{

cout << this->data[i] << endl;

}

}

Lab8Helper::~Lab8Helper()

{

}

**Source.cpp**

#include "Lab8Helper.h"

using namespace std;

void main()

{

Lab8Helper obj("data.txt", "res.txt");

obj.InitData();

obj.PrintData();

obj.AnalyzeData();

obj.PrintResults();

}

Результат роботи:

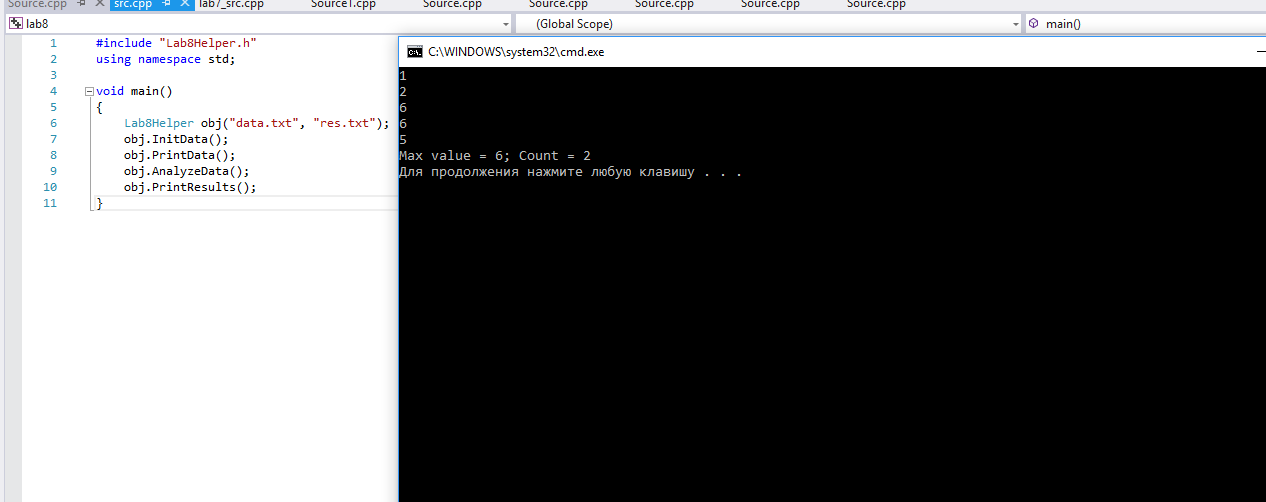


Рисунок 1.1 Ескіз вікна файлу

**Висновок:** На даній лабораторній роботі я ознайомився iз засобами розробки багатофайлових проектів в мові С.