**ЗВІТ З ЛАБОРАТОРНОЇ РОБОТИ**

за курсом «Об’єктно орієнтоване програмування»

студента групи ПА-21-1

Афанасьєва Максима Юрійовича

кафедра комп’ютерних технологій, ДНУ

2023/2024

Зміст

1. Постановка задачі. . .

2. Опис розв’язку. . .

3. Опис інтерфейсу програми (керівництво користувача). . .

4. Опис тестових прикладів. . .

# 

# 1. Постановка задачі

Формулювання завдання: Створити проект "Hello, git!" на мові C++,

використовуючи наданий шаблон інфраструктури з інтеграцією до github.

Налаштувати інфраструктуру (IDE + git).

Взаємодія з проектами, зокрема, збереження проекту в іншій папці та переміщення його з одного комп'ютера на інший. Змінити параметри IDE (наприклад, кольори, клавішні комбінації) та параметри проекту (наприклад, ключі компілятора, командний рядок). Створити гілку (branch), ініціювати pull request (ПР).

Написати програму, що викликає функції, описані в інших (окремо компільованих) файлах. Здатність відкривати/закривати проект і додавати новий файл до існуючого проекту. Передавати і обробляти параметри функцій, зокрема функції main, і повертати результати роботи функцій.

# 2. **Опис** розв’язку

Програма отримує послідовність аргументів через параметри командного рядка, якщо параметри не були введені, програма взаємодіє з користувачем, отримуючи від нього ім'я файлу з послідовністю. Це введення здійснюється через консольний діалог. Якщо серед елементів є нечисельний вираз, програма інкрементує значення змінної end, що забороняє виконуватися решті коду та видає помилку.

#include <iostream>

#include <fstream>

#include <string>

#include <algorithm>

#include "Second.h"

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

{

const std::string numbers = "0123456789., ";

std::string filename;

double data[10]{};

int b = 0;

int end = 0;

if (argc < 2)

{

std::cout << "No arguments were passed, please choose a file with arguments: ";

std::cin >> filename;

std::ifstream myfile;

myfile.open(filename);

if (myfile.is\_open())

{

std::ifstream infile(filename);

std::string line, varName;

std::string delimiter = " ";

std::getline(infile, line);

if (line.find\_first\_not\_of(numbers) != std::string::npos)

{

std::cout << std::endl << "Error! Arguments must be numbers!";

end++;

}

else

{

size\_t pos = 0;

while ((pos = line.find(delimiter)) != std::string::npos)

{

varName = line.substr(0, pos);

line.erase(0, pos + delimiter.length());

data[b] = std::stod(varName);

b += 1;

continue;

}

data[b] = std::stod(line);

}

}

else std::cout << std::endl << "Unable to open file";

}

Після цього програма перевіряє щоб в послідовності були тільки дійсні числа та переносить їх у масив data, якщо аргументи були получені через параметри командного рядка.

if (argc >= 2)

{

for (int i = 1; i < argc; i++)

{

if (std::string(argv[i]).find\_first\_not\_of(numbers) != std::string::npos)

{

std::cout << "Error! Arguments must be numbers!";

}

}

for (int i = 1; i < argc; i++)

{

data[i] = atof(argv[i]);

}

}

Програма розраховує кількість елементів масиву, що знадобиться для двух наступних функцій, визначених в окремому файлі Second.cpp. Після чого виконує їх. Якщо змінна end була інкрементована раніше, то програма не виконає решту коду.

int n = sizeof(data) / sizeof(data[0]);

mathin result = means(data, n);

std::sort(data, data + n, comp);

Вміст Second.cpp:

#include <iostream>

#include <string>

struct mathin {

double avg;

double havg;

};

bool comp(const double& a, const double& b) {

int sumA;

int sumB;

std::string numStr = std::to\_string(a);

size\_t decimalPos = numStr.find('.');

if (decimalPos != std::string::npos) {

sumA = 0;

for (size\_t i = decimalPos + 1; i < numStr.length(); ++i) {

sumA += numStr[i] - '0';

}

}

numStr = std::to\_string(b);

decimalPos = numStr.find('.');

if (decimalPos != std::string::npos) {

sumB = 0;

for (size\_t i = decimalPos + 1; i < numStr.length(); ++i) {

sumB += numStr[i] - '0';

}

}

return sumA < sumB;

}

mathin means(double arr[], int a)

{

mathin result;

double temp[10]{};

double n = 0;

for (int i = 0; i < a; i++)

{

temp[i] = arr[i];

if (temp[i] != 0)

{

n++;

}

}

double sum = 0;

result.avg = 0;

for (int i = 0; i <= n; i++)

{

sum += arr[i];

}

result.avg = sum / n;

double harm\_mean = 0;

for (size\_t e = 0; e < n; ++e)

{

if (arr[e] != 0)

{

harm\_mean += 1.0 / arr[e];

}

}

result.havg = n / harm\_mean;

return result;

}

Програма отримує ім’я файлу для виводу даних у діалозі з користувачем, після чого, саме, виводить в нього дані.

std::cout << std::endl << "Please choose a file to output to: ";

std::cin >> filename;

std::ofstream myfile;

myfile.open(filename);

if (myfile.is\_open())

{

myfile << "Sorted parameters: ";

for (int i = 0; i < n; i++)

{

if (data[i] != 0)

{

myfile << data[i] << " ";

}

}

myfile << std::endl << "Average: " << result.avg;

myfile << std::endl << "Harmonic mean: " << result.havg;

}

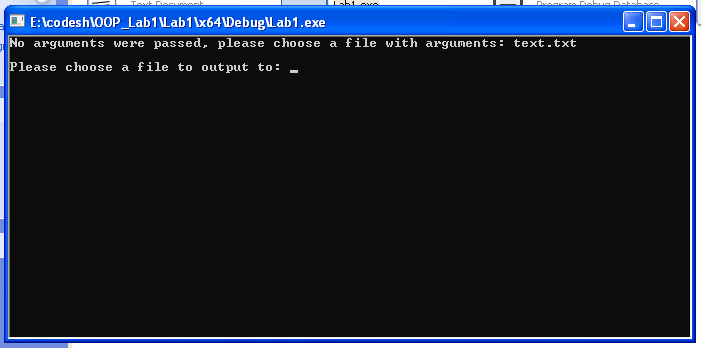
else std::cout << std::endl << "Unable to open file";

}

# 3. Опис інтерфейсу програми (керівництво

# **Користувача)**

Якщо користувач не ввів параметрів командного рядка, то його зустріне повідомлення «No arguments were passed, please choose a file with arguments:», після чього він повинен ввести до 11 чисел. Якщо серед введених даних є нечисельний вираз, то програма видасть помилку «Error! Arguments must be numbers!». Якщо все пройшло без помилок, програма виведе текст «Please choose a file to output to:» та запросить користувача ввести назву файлу для виводу даних.



# 4. Опис тестових прикладів

1) Вхідні дані: 0.2222 0.11111 0.33 0.999999 0.4444 0.7777

Результат: Відсортовані дані: 0.11111 0.33 0.2222 0.4444 0.7777 0.999999

Середнє аріфметичне: 0.480901

Середнє гармонічне: 0.284807

2) Вхідні дані: 785.457 47.7546 0.8789 123.123 11.44

Результат: Відсортовані дані: 123.123 11.44 785.457 47.7546 0.8789

Середнє аріфметичне: 242.163

Середнє гармонічне: 3.20664

3) Вхідні дані: 57.122 37.123 24.333 11.121 23.22

Результат: Відсортовані дані: 11.121 23.22 57.122 37.123 24.333

Середнє аріфметичне: 30.5838

Середнє гармонічне: 22.8805

4) Вхідні дані: 456.456 666.666 777.777 69.420 78.456

Результат: Відсортовані дані: 69.42 78.456 456.456 666.667 777.778

Середнє аріфметичне: 409.755

Середнє гармонічне: 155.63

5) Вхідні дані: 7894.4564 2456.78 223.4587

Результат: Відсортовані дані: 2456.78 7894.46 223.459

Середнє аріфметичне: 3524.9

Середнє гармонічне: 598.945