МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ

РОССИЙСКОЙ ФЕДЕРАЦИИ

ФГАОУ ВО «СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

ИНСТИТУТ ИНФОРМАЦИОННЫХ

ТЕХНОЛОГИЙ И ТЕЛЕКОММУНИКАЦИЙ

КАФЕДРА ПРИКЛАДНОЙ ИНФОРМАТИКИ

**Лабораторная работа №5**

по дисциплине

«Объектно-ориентированное программирование»

**Выполнил:**

Ермолаев Кирилл Александрович

Студент 2 курса группы \_ПИН-б-о-22-1

Направления подготовки

09.03.03 Прикладная информатика

очной формы обучения

Ставрополь, 2023 г.

**Тема:** Основы объектно-ориентированного программирования на ЯП С++.

**Цель работы:** Файловые и строковые потоки. Строки класса string.

**Ход работы:**

**Вариант 7 -** Написать программу, которая считывает текст из файла и определяет, сколько в нем слов, состоящих из не более чем четырех буквами.

// main.cpp

#include <iostream>

#include "FileReader.h"

#include "WordCounter.h"

int main() {

std::string filename;

std::cout << "Enter the file name: ";

std::cin >> filename;

FileReader fileReader(filename);

if (!fileReader.isOpen()) {

return 1;

}

std::vector<std::string> lines = fileReader.getLines();

int totalShortWords = 0;

for (const auto& currentLine : lines) {

totalShortWords += WordCounter::countShortWords(currentLine);

}

std::cout << "The number of words of no more than four letters: " << totalShortWords << std::endl;

return 0;

}

// FileReader.h

#pragma once

#ifndef FILEREADER\_H

#define FILEREADER\_H

#include <fstream>

#include <vector>

#include <string>

class FileReader {

public:

explicit FileReader(const std::string& filename);

std::vector<std::string> getLines() const;

bool isOpen() const;

~FileReader();

private:

std::ifstream file;

};

#endif

// FileReader.cpp

#include "FileReader.h"

#include <iostream>

#include <fstream>

#include <iterator>

FileReader::FileReader(const std::string& filename) {

file.open(filename);

if (!file.is\_open()) {

std::cerr << "The file could not be opened." << std::endl;

}

}

std::vector<std::string> FileReader::getLines() const {

std::vector<std::string> lines;

std::string line;

// Используем итераторы для считывания строк

std::istreambuf\_iterator<char> it(file.rdbuf()), end;

while (it != end) {

// Считываем строку до символа новой строки

line.clear();

while (it != end && \*it != '\n') {

line += \*it++;

}

lines.push\_back(line);

// Пропускаем символ новой строки, если он есть

if (it != end) {

++it;

}

}

return lines;

}

bool FileReader::isOpen() const {

return file.is\_open();

}

FileReader::~FileReader() {

if (file.is\_open()) {

file.close();

}

}

// WordCounter.h

#pragma once

#ifndef WORDCOUNTER\_H

#define WORDCOUNTER\_H

#include <string>

class WordCounter {

public:

static int countShortWords(const std::string& text);

};

#endif

// WordCounter.cpp

#include "WordCounter.h"

#include <sstream>

int WordCounter::countShortWords(const std::string& text) {

std::istringstream iss(text);

std::string word;

int count = 0;

while (iss >> word) {

if (word.length() <= 4) {

count++;

}

}

return count;

}

// UnitTest4.cpp

#include "pch.h"

#include "CppUnitTest.h"

#include "..\ConsoleApplication1\FileReader.h"

#include "..\ConsoleApplication1\WordCounter.h"

#include "..\ConsoleApplication1\FileReader.cpp"

#include "..\ConsoleApplication1\WordCounter.cpp"

using namespace Microsoft::VisualStudio::CppUnitTestFramework;

TEST\_CLASS(WordCounterTest) {

public:

TEST\_METHOD(CountShortWords) {

// Подготовка данных

std::string text = "This is a test file. \

It contains words of varying lengths. \

Some short words are here. \

Longer words are also included. \

Check the program's output for the correct count.";

// Выполнение теста

int result = WordCounter::countShortWords(text);

// Проверка результата

Assert::AreEqual(13, result);

}

};

TEST\_CLASS(FileReaderTest) {

public:

TEST\_METHOD(GetLines) {

// Подготовка данных

const std::string filename = "testfile.txt";

std::ofstream testFile(filename);

testFile << "Line 1\nLine 2\nLine 3";

testFile.close();

FileReader fileReader(filename);

// Выполнение теста

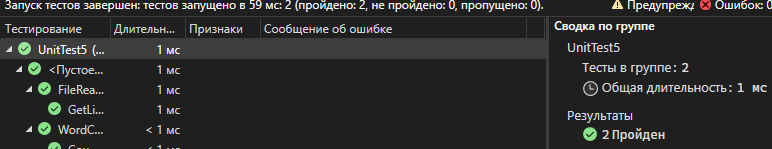
std::vector<std::string> lines = fileReader.getLines();

// Проверка результата

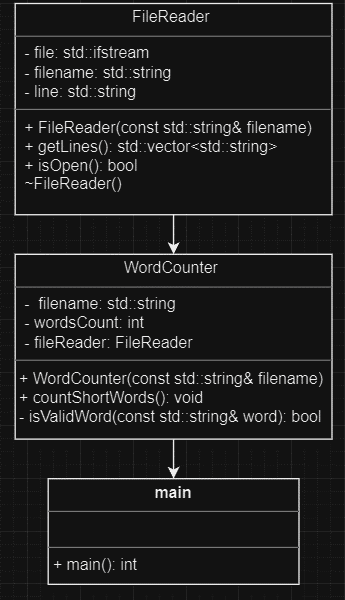
Assert::AreEqual(static\_cast<size\_t>(3), lines.size()); // Замените 3 на ожидаемое количество строк

}

};



Тестовый файл – <test.txt>



<https://github.com/MoeTomatoki/OOP>

**Вывод:** получил базовые знания по работе с файловыми и строковыми потоками. Научился работать со строками класса string.