Министерство науки и высшего образования РФ

Пензенский государственный университет

Кафедра «Вычислительная техника»

**ОТЧЕТ**

по лабораторной работе №--

по дисциплине «Основы операционных систем»

на тему «--»

Выполнили: студенты группы 22ВВП1

Беляев Д. И.

Демин М. С.

Приняли:

Деев М. В.

Патунин Д. В.

Пенза 2024

**Цель работы**

Изучение объектно-ориентированного программирования и получение навыков написания программ, содержащих файловый ввод-вывод, контейнеры, итераторы и регулярные выражения на примере языка программирования C++.

**Задание**

Написать программу, позволяющую пользователю извлекать из текстового документа, написанного с использованием языка разметки, определённое содержимое. Язык разметки и извлекаемое содержимое выбираются в соответствии с вариантом (таблица 1.1).

|  |  |  |
| --- | --- | --- |
| 8 | Markdown | заголовки,  абзацы,  ссылки |

Имя файла передаётся как аргумент командной строки.

Для создания и управления коллекциями объектов использовать контейнеры. Для обхода контейнеров использовать итераторы.

Поиск производить посредством регулярных выражений.

**Листинг**

#include <iostream>

#include <fstream>

#include <string>

#include <regex>

#include <vector>

#define EXAMPLE\_FILE\_PATH "markdown\_example.txt"

struct Content {

std::vector<std::string> headers;

std::vector<std::string> paragraphs;

std::vector<std::string> links;

};

Content extractContent(const std::string& filename) {

std::ifstream file(filename);

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

throw std::runtime\_error("Error when open file\n");

}

Content content;

std::string line;

std::regex headerRegex(R"(^#{1,6}\s\*(.\*))");

std::regex paragraphRegex(R"(^\s\*(\S.\*))");

std::regex linkRegex(R"(\[([^\]]+)\]\(([^)]+)\))");

while (std::getline(file, line)) {

std::smatch match;

if (std::regex\_match(line, match, headerRegex)) {

content.headers.push\_back(match[1]);

}

else if (std::regex\_match(line, match, paragraphRegex)) {

content.paragraphs.push\_back(match[1]);

}

auto words\_begin = std::sregex\_iterator(line.begin(), line.end(), linkRegex);

auto words\_end = std::sregex\_iterator();

for (std::sregex\_iterator i = words\_begin; i != words\_end; ++i) {

std::smatch match = \*i;

content.links.push\_back(match[2]);

}

}

return content;

}

int main(int argc, char\* argv[]) {

std::string path = argc == 2 ? argv[1] : EXAMPLE\_FILE\_PATH;

std::cout << "File path: " << path << std::endl << std::endl;

Content content = extractContent(path);

std::cout << "==Headers==" << std::endl;

for (const auto& header : content.headers) {

std::cout << header << std::endl;

}

std::cout << "\n==Paragraphs==" << std::endl;

for (const auto& paragraph : content.paragraphs) {

std::cout << paragraph << std::endl;

}

std::cout << "\n==Links==" << std::endl;

for (const auto& link : content.links) {

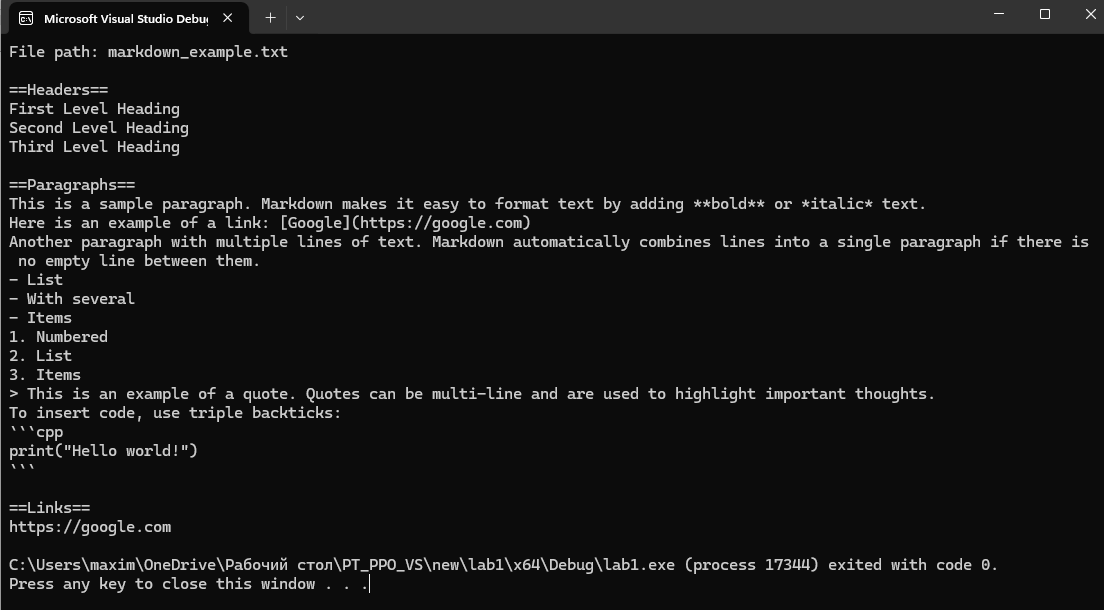
std::cout << link << std::endl;

}

return 0;

}

**Результат выполнения программы**

****

**Вывод**

Изучили объектно-ориентированное программирование и получили навыки написания программ, содержащих файловый ввод-вывод, контейнеры, итераторы и регулярные выражения на примере языка программирования C++.