Министерство образования Республики Беларусь

Учреждение Образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Кафедра электронных вычислительных средств

Лабораторная работа № 1

«Создание логгера»

|  |  |
| --- | --- |
| Проверил: | Выполнил: |
| Лихачёв Д. С. | ст. гр. 850701 |
|  | Филипцов Д. А. |
|  | |

Минск 2021

# Цель работы:

Ознакомиться с основными принципами работы с файлами и файловой системой.

# **Ход работы:**

Создать класс для логгирования с выводом в консоль и файл форматированных текстовых сообщений. В том случае, если сообщения выводятся в файл, при создании класса в конструктор должно передаваться имя лог-файла. Класс должен содержать следующие методы для вывода текстовых сообщений в лог-файл или в консоль:

printError(std::string text)

printDebug(std::string text)

printTrace(std::string text)

Текстовые сообщения, которые выводятся с помощью методов printError(…), printDebug(…) и printTrace(…) должны форматироваться в соответствии с форматом, устанавливаемым с помощью метода setFormat(std::string format). Строка format может содержать следующие теги:

\*\*message \*\* - сообщение

%data% - текущая дата

%%time%% - текущее время

\*prior\* - приоритет сообщения (Trace, Debug или Error)

Указанные теги в строке format могут идти в любом порядке и в любом количестве. Для определения вхождения указанных тегов в строку формата использовать регулярные выражения С++.

Создать консольное приложение с использованием С++ для демонстрации работы разработанного класса логгера.

Код программы, включая класс Logger и консольное приложение для демонстрации работы.

#include <iostream>

#include <fstream>

#include <regex>

#include <ctime>

using namespace std;

class Logger

{

private:

string m\_filename;

ofstream m\_file;

string m\_format;

regex m\_reg\_data;

regex m\_reg\_time;

regex m\_reg\_message;

regex m\_reg\_prior;

void printFileCml(string text)

{

if (m\_filename == "")

{

cout << text << "\n";

}

else

{

m\_file << text << "\n";

}

}

string formatting(string message, string prior)

{

time\_t now = time(0);

tm date\_time;

localtime\_s(&date\_time, &now);

char date[20];

strftime(date, sizeof(date), "%d.%m.%Y", &date\_time);

char time[20];

strftime(time, sizeof(time), "%H:%M:%S", &date\_time);

string text = regex\_replace(m\_format, m\_reg\_data, date);

text = regex\_replace(text, m\_reg\_time, time);

text = regex\_replace(text, m\_reg\_message, message);

text = regex\_replace(text, m\_reg\_prior, prior);

return text;

}

public:

Logger() // Конструктор для вывода в консоль

{

m\_format = "%data% - %%time%% - \*\*message\*\* - \*prior\*";

m\_reg\_data = regex("(%data%)");

m\_reg\_time = regex("(%%time%%)");

m\_reg\_message = regex("(\\\*\\\*message\\\*\\\*)");

m\_reg\_prior = regex("(\\\*prior\\\*)");

}

Logger(string filename): Logger() // Конструктор для вывода в файл

{

m\_filename = filename;

m\_file.open(m\_filename, ios::app);

}

~Logger()

{

m\_file.close();

}

void setFormat(string format)

{

m\_format = format;

}

void printError(string text)

{

printFileCml(formatting(text, "ERROR"));

}

void printDebug(string text)

{

printFileCml(formatting(text, "DEBUG"));

}

void printTrace(string text)

{

printFileCml(formatting(text, "TRACE"));

}

};

int main()

{

Logger logger1;

logger1.setFormat("Date and time: %data% / %%time%% Message: \*\*message\*\* Priority type: \*prior\*");

logger1.printError("ERROR with opening file!");

logger1.printTrace("Don't worry, be happy!");

Logger logger2("Log\_file.txt");

logger2.printTrace("Everything is GOOD!");

logger2.setFormat("Message: \*\*message\*\* %%time%% Priority type: \*prior\* %%time%% ");

logger2.printDebug("Calling function \"adding\".");

system("pause");

return 0;

}

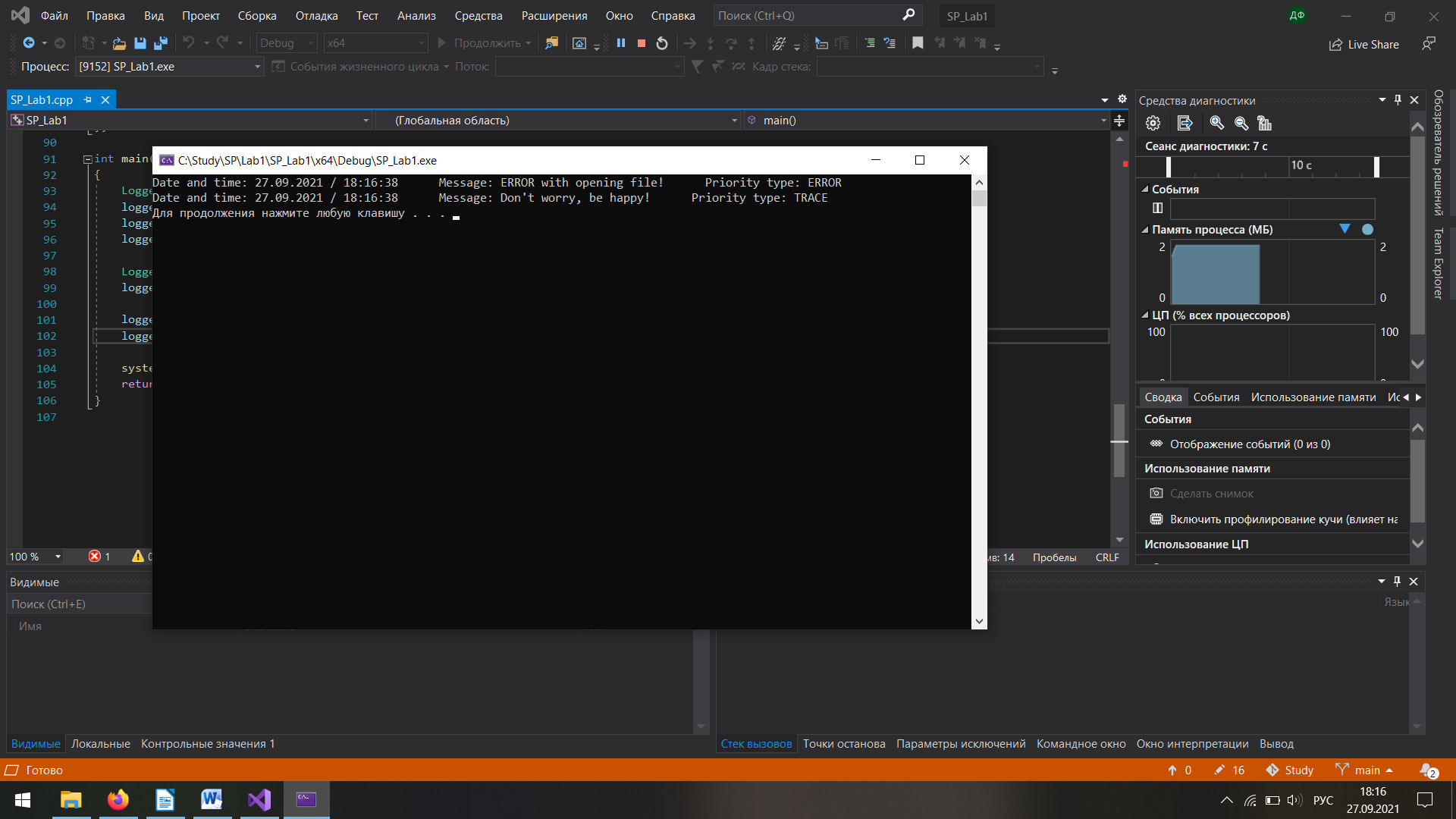


Рисунок 1 – Результат выполнения консольного приложения (вывод в консоль)

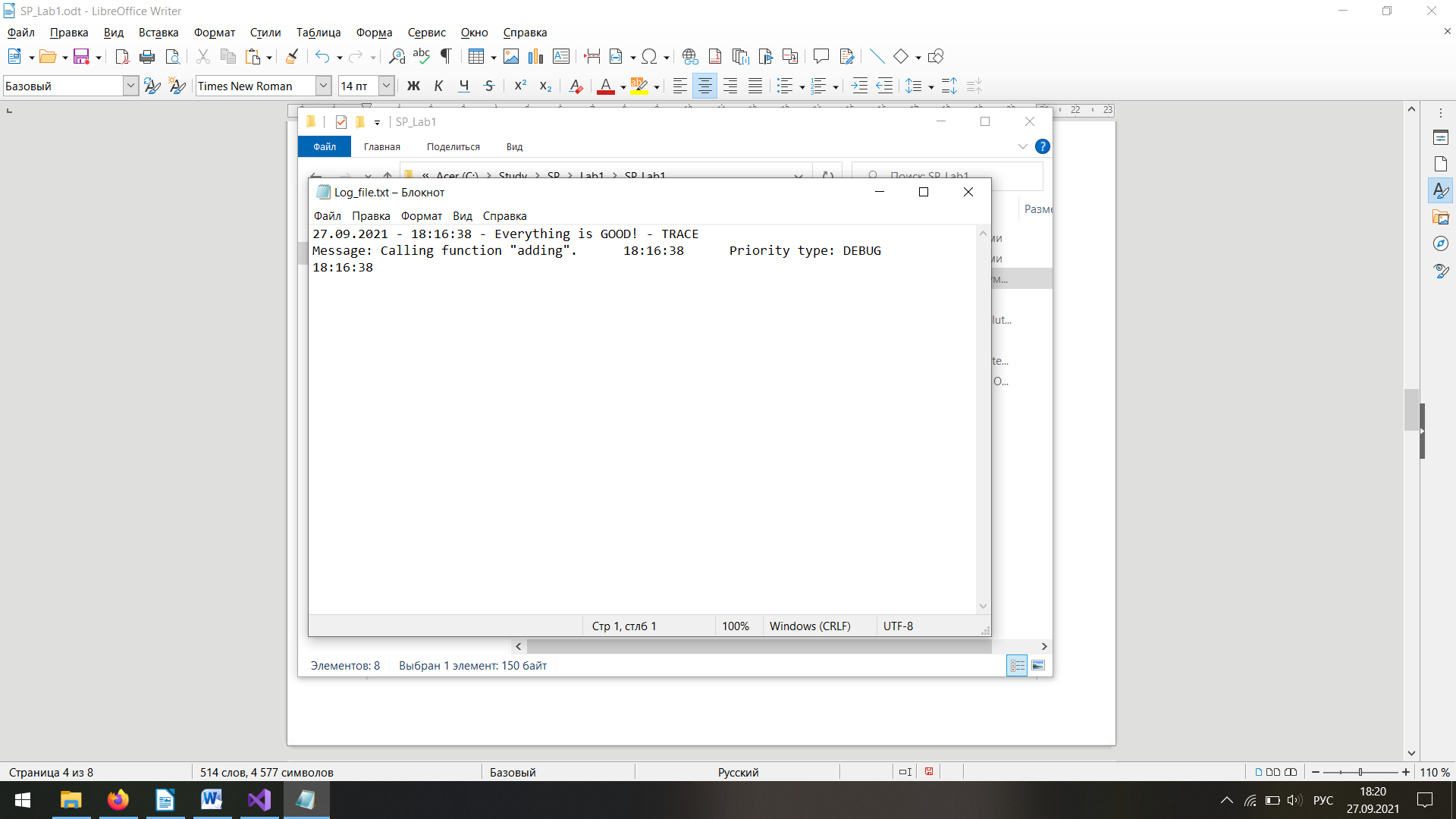


Рисунок 2 – Результат выполнения консольного приложения (запись в файл)

# Вывод

В ходе лабораторной работы были приобретены навыки работы с файлами и файловой системой, был усвоен принцип работы логгера.