Министерство образования Республики Беларусь
Учреждение Образования
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Кафедра электронных вычислительных средств

Лабораторная работа № 1
«Создание логгера»

Проверил:
Лихачёв Д. С.

Выполнил:
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 C++.

Создать консольное приложение с использованием C++ 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;
}
```

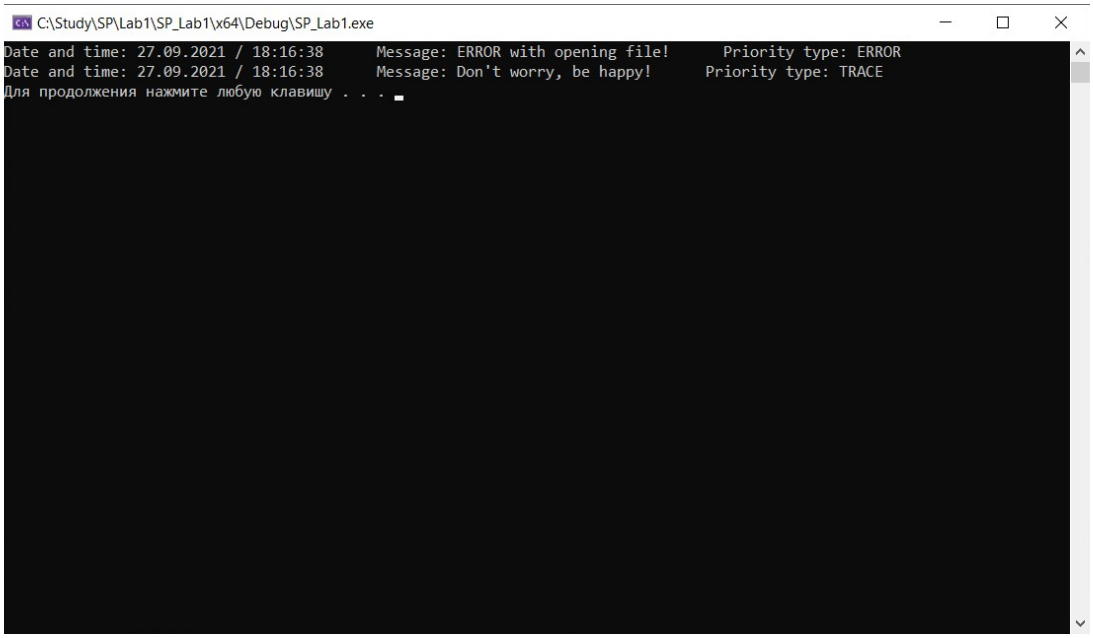

Рисунок 1 – Результат выполнения консольного приложения (вывод в консоль)

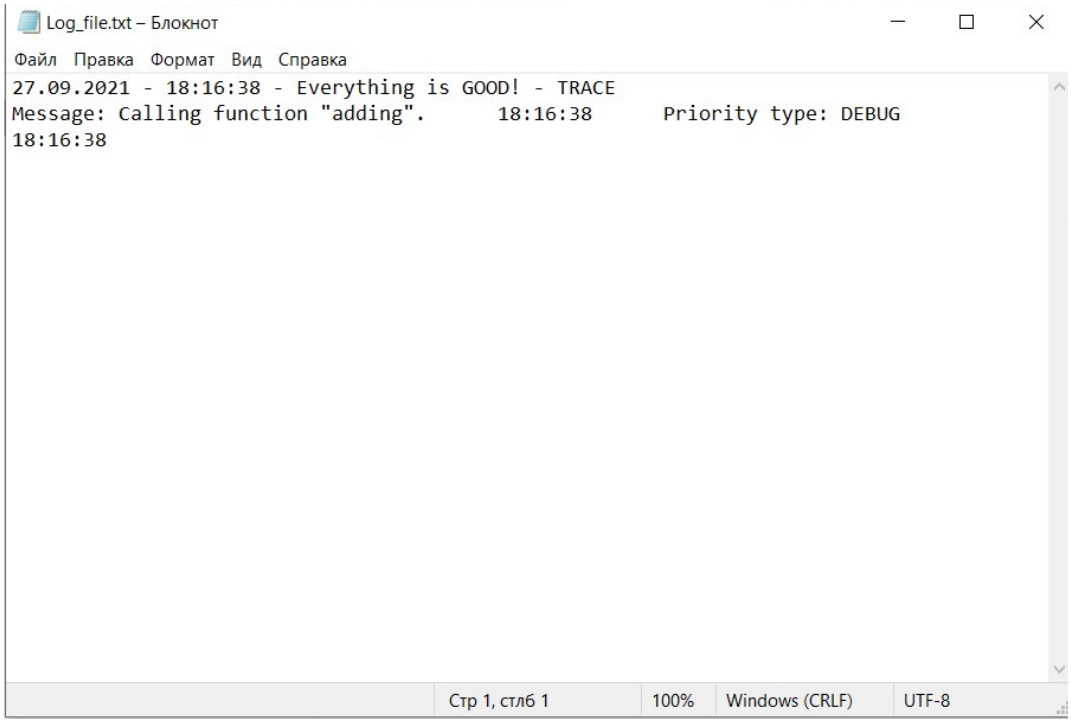

Рисунок 2 – Результат выполнения консольного приложения (запись в файл)

Вывод

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