

**Федеральное государственное бюджетное образовательное учреждение  
высшего профессионального образования  
"Уфимский государственный авиационный технический университет"**

**Кафедра** Высокопроизводительных вычислительных технологий и систем

**Дисциплина:** Программирование

**Отчет по лабораторной работе № 8**

**Тема:** «Ввод/вывод в C++.  
Работа с файлами в C++.»

Группа МКН-113	Фамилия И.О.	Подпись	Дата	Оценка
Студент	Шамаев И.Р.			
Принял	Гайнетдинова А.А.			

**Уфа 2020**

**Цель:** изучить средства языка C++ для работы с файлами и варианты их использования.

### **Теоретический материал**

Манипуляторы форматирования:

setw(n) (ширина поля вывода);

setprecision(n) (количество цифр (n-1) в дробной части числа);

dec / oct / hex (основания систем счисления);

right / left (выравнивание);

boolalpha / noboolalpha (отображение логических переменных 0 / 1 в true / false);

showpos / noshowpos (вывод знака + для положительных чисел);

uppercase / lowercase (прописные / строчные буквы в записях чисел);

setfill(c) (заполнитель), fixed / scientific (фикс. / экспоненциальная форма вывода вещ. чисел).

Функции-элементы для установки, снятия флагов:

cout.setf(ios::flag);

cout.unsetf(ios::flag).

Манипуляторы: setiosflags(ios::flags), resetiosflags(ios::flags).

Функции-элементы для установки указателя позиции в файле:

seekg() в istream;

seekp() в ostream.

tellp и tellpg возвращают текущие позиции указателей get и put.

Функции-элементы write и read соответственно выводят и считывают байты потока с заданной позиции в файле.

### **Ответы на контрольные вопросы**

- 1. Что такое поток?** В языке C++ производится ввод-вывод потоков байтов. Поток – это последовательность байтов. Специальные типы данных для работы с файлами.
- 2. Что представляет собой файловый указатель?** Они показывают позицию для последующего ввода (указатель get) или вывода (указатель put).

### **3. Перечислить режимы доступа к файлу?**

ios::in - открыть файл в режиме чтения данных, по умолчанию для потоков ifstream);

ios::out — открыть файл в режиме записи данных (при этом информация в существующем файле уничтожается) (по умолчанию для потоков ofstream);

ios::app — открыть файл в режиме записи данных в конец файла;

ios::ate — передвинуться в конец уже открытого файла;

ios::trunc — очистить файл, это же происходит в режиме ios::out;

`ios::nocreate` — не выполнять операцию открытия файла, если он не существует;

`ios::noreplace` — не открывать существующий файл.

4. **Как открыть и как закрыть файл?** При объявлении с помощью конструкторов либо после объявления, используя функции-элемент `open()`. Закрыть с помощью функции-элемента `close()`.
5. **Как организуется форматированный ввод-вывод в C++?** С помощью флагов форматного ввода-вывода и манипуляторов форматирования.
6. **Какой заголовочный файл C++ содержит инструкции файлового ввода/вывода?** `fstream`.
7. **Что такое `vector`?** Это класс, который предназначен для облегчения работы с массивами.

## Индивидуальное задание №1

### Задание:

1. Считать данные из первого файла, дополнив информацией из второго, так чтобы получилось 7 характеристик (Time, Point ID, X, Y, Z, Temperature, Displacement X) в числовом формате (double, int) для всех строчек первого файла. Порядок хранения характеристик в каждой строке - на усмотрение разработчика.

Вывести на экран данные для заданной пользователем точки (по координатам X, Y, Z) и момента времени.

2. Записать итоговые данные в новый файл в виде колонок, отделённых друг от друга символами табуляции.
3. Записать в новый файл выборку из считанных данных согласно условию и в формате:

Вариант	Описание	Условие	Формат вывода
12.	Информация по температуре и перемещению вдоль заданной линии, параллельной оси Y, в каждый момент времени	Координаты X и Z задаются параметром (вводятся пользователем)	Time Y Displacement_X Temperature (через табуляцию) Time - в формате с плавающей точкой с 3 знаками после запятой, Y - в формате с плавающей точкой с 5 знаками после запятой, Displacement X - в экспоненциальном формате с 2 знаками после запятой в мантиссе, Temperature - в формате с плавающей точкой с 2 знаками после запятой

### Исходный код программы:

```
#include <vector>
#include <iostream>
#include <fstream>
#include <string>
#include <iomanip>
#include <sstream>

using namespace std;

typedef struct
{
    int Point_ID;

    double Temperature;

    double Time, Displacement_X, X, Y, Z , X_a, Z_a;
}Point;

vector<Point> point_information(ifstream&, ifstream&);

//void CreatNewFile(vector<Point>&);

void OutputFile(vector<Point>&);
```

```

vector<Point> point_information(ifstream& file_1, ifstream& file_2) {

    string line;//Строчка текста
    int line_ID;
    vector<Point> dtb;
    Point T_Point;
    int mark_1, mark_2, count = 0, i;
    double comparison_time, X, Y, Z ,Z_a, X_a;

    getline(file_1, line);
    getline(file_2, line);

    mark_1 = file_1.tellg();//Возвращает текущую позицию чтения
    mark_2 = file_2.tellg();//Возвращает текущую позицию чтения
    file_1 >> comparison_time;

    file_1.seekg(mark_1);//Перемещаемся в позицию mark_1

    while (file_1 >> T_Point.Time >> T_Point.Point_ID >> T_Point.Temperature >>
T_Point.Displacement_X)// Пока есть данные, которые мы можем прочитать
    {

        if (T_Point.Time == comparison_time) {
            file_2 >> line_ID >> T_Point.X >> T_Point.Y >> T_Point.Z;
        }
        else {
            file_2.seekg(mark_2);//Перемещаемся в позицию mark_2
            file_2 >> line_ID >> T_Point.X >> T_Point.Y >> T_Point.Z;
            comparison_time = T_Point.Time;
        }

        dtb.push_back(T_Point);//Добавление в конец вектора
    }

    //comparison_time = 0;

    cout << "Введите X и Z:" << endl;

    cin >> X >> Z;
    dtb.at(1).X_a = X;
    dtb.at(1).Z_a =Z;
    for (i = 0; i < dtb.size(); i++)

```

```

{
    if (dtb.at(i).X == X && dtb.at(i).Z == Z) {

        cout <<
        "-----"
        "-----" << endl;

        cout << right << setw(5) << "Time" << setw(15) << "Point ID" <<
        setw(15) << "Temperature"
        << setw(15) << "Displacement X" << setw(15) << setw(15)
        << "X" << setw(15) << "Y" << setw(15) << "Z" << endl;

        cout << right << setw(5) << dtb.at(i).Time << setw(15) <<
        dtb.at(i).Point_ID << setw(15) << dtb.at(i).Temperature
        << setw(15) << dtb.at(i).Displacement_X << setw(15) <<
        dtb.at(i).X
        << setw(15) << dtb.at(i).Y << setw(15) << dtb.at(i).Z <<
        setw(15) << endl;

        cout <<
        "-----"
        "-----" << endl;

        count++;

    }

}

if (count == 0) {
    cerr << setw(70) << "Нет данных" << endl;
}

return dtb;
}
/*
void CreatNewFile(vector<Point>& dtb) {
    ofstream file("New_BD.txt", ios::out);

    file << "Time" << setw(10) << "\tID" << setw(10) << "\tTemperature" <<
    setw(10) << "\tDisplacement X" << setw(10) <<
    "\tX" << setw(10) << "\tY" << setw(10) << "\tZ" << endl;

    if (!file)
    {
        cerr << "New_BD.txt can not be opened." << endl;
        exit(1);
    }

    for (int i = 0; i < dtb.size(); i++)

```

```

    {
        file << dtb.at(i).Time << "\t" << setw(10) << dtb.at(i).Point_ID <<
"\t" << setw(10) << dtb.at(i).Temperature << "\t"
        << setw(10) << dtb.at(i).Displacement_X << "\t" << setw(10) <<
dtb.at(i).X << "\t" << setw(10) << dtb.at(i).Y
        << "\t" << setw(10) << dtb.at(i).Z << endl;
    }

    file.close();

    cout << "Complete" << endl;
}
*/

void OutputFile(vector<Point>& dtb) {
    ofstream file("Output_BD.txt", ios::out);

    file << "Point_ID" << "\t" << "Time" << "\t" << "Y" << "\t" << "Displacement X"
<< "\t" << "Temperature" << endl;

    vector <Point> general;

    Point T_Point;

    ifstream file2("New_BD.txt");
    if (file2.is_open())//Если открытие файла прошло успешно
    {
        string line;//Строчка текста

        while (getline(file2, line))
        {
            int Point_ID;
            double X, Y, Z, Displacement_X, Temperature, Time;

            //Создадим поток для считывания данных из строчки
            stringstream iss(line);

            //За раз всё считаем
            iss >> Time >> Point_ID >> Temperature >> Displacement_X >> X >> Y
>> Z;

            if ((dtb.at(1).X_a == X) && (dtb.at(1).Z_a == Z)) {
                T_Point.Time = Time;
                T_Point.Displacement_X = Displacement_X;
                T_Point.Y = Y;
                T_Point.Point_ID = Point_ID;
                T_Point.Temperature = Temperature;
                general.push_back(T_Point);
            }
        }
    }
    else
    {
        cout << "Не удалось открыть файл." << endl;
    }
}

```

```

        for (int i = 0; i < general.size(); i++)
        {
            file << fixed << setprecision(3) << general.at(i).Point_ID << "\t";
            file << fixed << setprecision(3) << general.at(i).Time << "\t";
            file << fixed << setprecision(5) << general.at(i).Y << "\t";
            file << scientific << setprecision(2) << general.at(i).Displacement_X
<< "\t";
            file << fixed << setprecision(2) << general.at(i).Temperature <<
endl;
        }

        file.close();

        cout << "Успешно!" << endl;
    }

    int main()
    {
        setlocale(LC_ALL, "RUSSIAN");
        vector <Point> dtb;

        ifstream BD("BD.txt", ios::in);
        ifstream BD_Coords("BD_Coords.txt", ios::in);
        if (!BD)
        {
            cerr << "BD.txt не удалось открыть." << endl;
            exit(1);
        }
        if (!BD_Coords)
        {
            cerr << "BD_Coords.txt не удалось открыть." << endl;
            exit(1);
        }

        dtb = point_information(BD, BD_Coords);
        //CreatNewFile(dtb);
        OutputFile(dtb);
        return 0;
    }

```



## Пример выполнения программы:

```
Введите X и Z:
0.013
0.0065
-----
Time    Point ID    Temperature Displacement X      X      Y      Z
0.05    22          335.054    -3.435e-06    0.013    0      0.0065
-----
Time    Point ID    Temperature Displacement X      X      Y      Z
0.05    473          293.639    -2.957e-06    0.013    0.001    0.0065
-----
Time    Point ID    Temperature Displacement X      X      Y      Z
0.05    945          306.207    -3.319e-06    0.013    0.000333333    0.0065
-----
Time    Point ID    Temperature Displacement X      X      Y      Z
0.05    946          295.999    -3.149e-06    0.013    0.000666667    0.0065
-----
Time    Point ID    Temperature Displacement X      X      Y      Z
0.05    1826         293.456      0      0.013    0.005    0.0065
-----
Time    Point ID    Temperature Displacement X      X      Y      Z
0.05    2403         293.569    -2.641e-06    0.013    0.0015    0.0065
-----
Time    Point ID    Temperature Displacement X      X      Y      Z
0.05    2404         293.577    -2.301e-06    0.013    0.002    0.0065
-----
Time    Point ID    Temperature Displacement X      X      Y      Z
0.07    8167          294.295    1.909e-06    0.013    -0.0025    0.0065
-----
Time    Point ID    Temperature Displacement X      X      Y      Z
0.07    8168          294.288    1.53e-06     0.013    -0.003    0.0065
-----
Time    Point ID    Temperature Displacement X      X      Y      Z
0.07    8169          294.277    1.147e-06    0.013    -0.0035    0.0065
-----
Time    Point ID    Temperature Displacement X      X      Y      Z
0.07    8170          294.26     7.63e-07     0.013    -0.004    0.0065
-----
Time    Point ID    Temperature Displacement X      X      Y      Z
0.07    8171          294.241    3.8e-07      0.013    -0.0045    0.0065
-----
Успешно!

C:\Users\MSI\source\repos\ConsoleApplication17\Debug\ConsoleApplication17.exe (процесс 10360) завершил работу с кодом 0.
Чтобы автоматически закрывать консоль при остановке отладки, включите параметр "Сервис" ->"Параметры" ->"Отладка" ->"Автоматически закрыть консоль при остановке отладки".
Нажмите любую клавишу, чтобы закрыть это окно...
```

Point_ID		Time	Y	Displacement X	Temperature
1	0.050	0.00000	-6.05e-06	309.12	
452	0.050	0.00100	-5.05e-06	293.36	
905	0.050	0.00033	-5.73e-06	300.28	
906	0.050	0.00067	-5.39e-06	295.33	
1805	0.050	0.00500	0.00e+00	293.36	
2263	0.050	0.00150	-4.54e-06	293.35	
2264	0.050	0.00200	-4.02e-06	293.33	
2265	0.050	0.00250	-3.49e-06	293.32	
2266	0.050	0.00300	-2.93e-06	293.30	
2267	0.050	0.00350	-2.34e-06	293.29	
2268	0.050	0.00400	-1.69e-06	293.29	
2269	0.050	0.00450	-9.49e-07	293.32	
5464	0.050	0.00000	1.00e-06	309.31	
5915	0.050	-0.00100		5.54e-07	293.97
6417	0.050	-0.00033		8.25e-07	307.91
6418	0.050	-0.00067		7.01e-07	293.50
7268	0.050	-0.00500		0.00e+00	294.07
8025	0.050	-0.00150		3.39e-07	293.68
8026	0.050	-0.00200		1.46e-07	293.76
8027	0.050	-0.00250		-1.50e-08	293.80
8028	0.050	-0.00300		-1.35e-07	293.83
8029	0.050	-0.00350		-2.06e-07	293.90
8030	0.050	-0.00400		-2.18e-07	293.99
8031	0.050	-0.00450		-1.60e-07	294.10
1	0.051	0.00000	-6.15e-06	309.60	
452	0.051	0.00100	-5.09e-06	293.49	
905	0.051	0.00033	-5.80e-06	300.59	
906	0.051	0.00067	-5.44e-06	295.46	
1805	0.051	0.00500	0.00e+00	293.48	
2263	0.051	0.00150	-4.58e-06	293.44	
2264	0.051	0.00200	-4.06e-06	293.45	
2265	0.051	0.00250	-3.53e-06	293.43	
2266	0.051	0.00300	-2.96e-06	293.41	
2267	0.051	0.00350	-2.36e-06	293.39	
2268	0.051	0.00400	-1.70e-06	293.39	
2269	0.051	0.00450	-9.60e-07	293.42	
5464	0.051	0.00000	1.00e-06	309.31	

**Вывод:** изучил на практике организацию вводы-вывода C++,  
организация работы с файлами C++.

### **Список использованной литературы**

1. Стивен Прата :*Язык программирования C++. Лекции и упражнения*, 5-е изд. Пер. с англ. М. : 000 "И.Д. Вильяме", 2007. - 1184 с.
2. Дейтел Х. М. *Как программировать на C++* / Х. М. Дейтел, П. Дж. Дейтел. – М. : Бином, 2008. – 1456 с.