

**МИНОБРНАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«ТУЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»**

Институт прикладной математики и компьютерных наук

Кафедра информационной безопасности

ЯЗЫКИ ПРОГРАММИРОВАНИЯ

Отчет по выполнению лабораторной работы № 13

Вариант №5

Выполнила _____

ст. гр.230711 Павлова Виктория Сергеевна

Проверила _____

доц. каф. ИБ Басалова Галина Валерьевна

Тула 2022

ЛАБОРАТОРНАЯ РАБОТА №13.

ТИПЫ ДАННЫХ, ОПРЕДЕЛЯЕМЫЕ ПОЛЬЗОВАТЕЛЕМ

ЦЕЛЬ РАБОТЫ:

Изучение возможности переименования типов, способов объявления и использования перечислимых типов, структур и объединений в языке C++.

ЗАДАНИЕ НА РАБОТУ:

- 1) Ознакомиться с теоретическим материалом, а также с конспектом лекций и рекомендуемой литературой по данной теме.
- 2) Разработать программу с использованием структур, в которой будет храниться и обрабатываться информация о стипендии студентов разных групп.
- 3) Разработать программу по своему варианту. Показать результаты работы преподавателю. Оформить отчет по работе: текст задачи, описание разработанных структур данных и функций, текст кода программы, результаты работы программы.

ХОД РАБОТЫ:

Согласно заданию варианта №5, необходимо объявить структуру для хранения информации о выполнении плана: наименование изделия, код изделия, единица измерения, план выпуска, фактически выпущено. Объявить массив «Результаты выполнения плана» и поместить в него сведения о нескольких изделиях. Разработать программу, позволяющую выполнить создание новой записи, запись всех данных в файл, чтение данных из файла, выдачу информации по введенному коду изделия, подсчет суммарного отклонения от плана (недовыполнение, перевыполнение) для изделий,

имеющих введенные единицы измерения, а также упорядочения изделий по наименованию (в алфавитном порядке).

В ходе выполнения работы была объявлена структура `products`, описание которой представлено в таблице 1.

Таблица 1 – Описание разработанной структуры `products`

Название поля	Тип данных	Предназначение
<code>name</code>	<code>string</code>	Наименование изделия
<code>measurement</code>	<code>string</code>	Единицы измерения
<code>code</code>	<code>int</code>	Код изделия
<code>plan</code>	<code>int</code>	План выпуска
<code>fact</code>	<code>int</code>	Фактический выпуск

Для работы с данной структурой были написаны функции, описание и предназначение которых представлено в таблице 2.

Таблица 2 – Описание функций для работы со структурой `products`

Объявление	Предназначение
<code>void NewRecord()</code>	Создание новой записи об изделии, запись происходит в массив «Результаты выполнения плана», а также в файл.
<code>void PutDataInFile()</code>	Запись всех имеющихся сведений обо всех изделиях в файл вывода.
<code>void ReadDataFromFile()</code>	Считать все данные из файла
<code>void PrintData()</code>	Вывод всех имеющихся сведений обо всех изделиях на экран
<code>void GetInfoByCode (int code)</code>	Вывод информации об изделии по введённому коду изделия
<code>void DeviationCounter (string measurement)</code>	Подсчёт и вывод отклонения от плана для всех изделий с заданными единицами измерения

КОД ПРОГРАММЫ:

```
#include <iostream>
#include <vector>
#include <string>
#include <algorithm>
#include <fstream>
#include <Windows.h>
using namespace std;

struct products {
    string name;
    string measurement;
    int code;
    int plan;
    int fact;
};

products resultsOfPlan[30];
int i = 0;

void NewRecord()
{
    ofstream input("input_5.txt", fstream::app); //app -> без
    перезаписи

    cout << "\n\n\t\t\t\t\t---> Добавить новую запись <---\n";
    cout << "\n\t\t\tНаименование изделия: "; getline(cin >> ws,
resultsOfPlan[i].name);
    input << "\n" << resultsOfPlan[i].name << "\n";
    cout << "\n\t\t\tЕдиница измерения: "; getline(cin >> ws,
resultsOfPlan[i].measurement);
    input << resultsOfPlan[i].measurement << "\n";
    cout << "\n\t\t\tКод изделия: "; cin >> resultsOfPlan[i].code;
    input << resultsOfPlan[i].code << "\n";
    cout << "\n\t\t\tПлан выпуска: "; cin >> resultsOfPlan[i].plan;
    input << resultsOfPlan[i].plan << "\n";
    cout << "\n\t\t\tФактически выпущено: "; cin >>
resultsOfPlan[i].fact;
    input << resultsOfPlan[i].fact;

    cout << "\n\tДобавлена запись: "
        << "\n\t\t\tизделие " << resultsOfPlan[i].name
        << " (" << resultsOfPlan[i].measurement << "), код изделия: "
        << resultsOfPlan[i].code << ", план выпуска: "
        << resultsOfPlan[i].plan << ", фактически выпущено: " <<
resultsOfPlan[i].fact << "\n";

    i++; //счетчик введенных записей
    return;
}

void PutDataInFile()
{
    ofstream output("output_5.txt"); //fstream::app -> без перезаписи

    for (size_t j = 0; j < i; j++)
```

```

    {
        output << resultsOfPlan[j].name << '\n';
        output << resultsOfPlan[j].measurement << "\n";
        output << resultsOfPlan[j].code << "\n";
        output << resultsOfPlan[j].plan << "\n";
        output << resultsOfPlan[j].fact << "\n";
    }
    output.close();
    cout << "\n\n\t\t!!!! Данные об изделиях успешно выведены в
файл.\n\n";
}

void ReadDataFromFile()
{
    ifstream input("input_5.txt");
    while (!input.eof())
    {
        getline(input >> ws, resultsOfPlan[i].name);
        getline(input >> ws, resultsOfPlan[i].measurement);
        input >> resultsOfPlan[i].code;
        input >> resultsOfPlan[i].plan;
        input >> resultsOfPlan[i].fact;
        i++;
    }
    input.close();

    cout << "\n\n\t\t!!!! Данные об изделиях успешно считаны из
файла.\n\n";
    return;
};

void PrintData()
{
    vector <string> s;
    for (size_t j = 0; j < i; j++)
    {
        s.push_back
        (
            "Наименование изделия: " + resultsOfPlan[j].name + " ("
            + resultsOfPlan[j].measurement + "), " + "код "
            + to_string(resultsOfPlan[j].code) + ", план выполнения: "
            + to_string(resultsOfPlan[j].plan) + ", фактически
выпущено: "
            + to_string(resultsOfPlan[j].fact)
        );
    }
    sort(s.begin(), s.end());
    cout << "\n\n\t\t\t\t!!!! Сведения об изделиях: " << "\n\n";
    for (size_t j = 0; j < i; j++)
    {
        cout << "\t\t" << j + 1 << ". " << s[j] << "\n";
    }
    cout << "\n";
    return;
};

void GetInfoByCode(int code)
{
    cout << "\n\n\t\t\t\t!!!! Сведения об изделии с кодом: " << code
<< "\n\n";
}

```

```

        for (size_t j = 0; j < i; j++)
        {
            if (resultsOfPlan[j].code == code)
            {
                cout << "\n\t\tНаименование изделия: " <<
resultsOfPlan[j].name;
                cout << "\n\t\tКод изделия: " << code;
                cout << "\n\t\tЕдиницы измерения: " <<
resultsOfPlan[j].measurement;
                cout << "\n\t\tПлан выпуска: " << resultsOfPlan[j].plan;
                cout << "\n\t\tФактически выпущено: " <<
resultsOfPlan[j].fact;
            }
        }
        return;
    };

void DeviationCounter(string measurement)
{
    cout << "\n\n\t\t\t\t!!!! Суммарное отклонение от плана для
изделий, "
        << " (" << measurement << ")" << "\n\n";
    for (size_t j = 0; j < i; j++)
    {
        if (resultsOfPlan[j].measurement == measurement)
        {
            int deviation = resultsOfPlan[j].plan -
resultsOfPlan[j].fact;
            cout << "\n\t\tНаименование изделия: " <<
resultsOfPlan[j].name;
            cout << "\n\t\tКод изделия: " << resultsOfPlan[j].code;
            if (deviation < 0)
                cout << "\n\t\tСуммарное отклонение (недовыполнение):
" << abs(deviation);
            else {
                if (deviation > 0)
                    cout << "\n\t\tСуммарное отклонение
(перевыполнение): " << deviation;
                else
                    cout << "\n\t\tСуммарное отклонение: " <<
deviation;
            }
        }
    }
    return;
}

int main()
{
    SetConsoleCP(1251);
    SetConsoleOutputCP(1251);
    char option;
    do
    {
        cout << "\n\n\t\t\t\t---> Меню <---\n\n";
        cout << "\t\t1 - Прочитать сведения об изделиях из файла\n";
        cout << "\t\t2 - Добавить новую запись\n";
        cout << "\t\t3 - Вывести все данные об изделиях в файл\n";
        cout << "\t\t4 - Выдать информацию об изделиях по коду\n";
        cout << "\t\t5 - Подсчитать суммарное отклонение по единицам
измерения\n";
    }

```

```

        cout << "\t\t6 - Вывести все сведения об изделиях (в алф.
порядке)\n";
        cout << "\t\t7 - Выход\n";
        cout << "\t\t\t\t---> Номер действия: "; cin >> option;
        switch (option)
        {
            case '1': ReadDataFromFile(); break;
            case '2': NewRecord(); break;
            case '3': PutDataInFile(); break;
            case '4':
            {
                int code;
                cout << "\n\t\t\t---> Введите код для поиска информации:
";

                cin >> code;
                GetInfoByCode(code); break;
            }
            case '5':
            {
                string measurement;
                cout << "\n\t\t\t---> Введите единицы измерения: ";
                getline(cin >> ws, measurement);
                DeviationCounter(measurement); break;
            }
            case '6': PrintData(); break;
            default: option = '7';
        }
    } while (option != '7');
    cout << "\n\t\t\t\t\t!!!! Работа завершена успешно!\n";
    return 0; }

```

РЕЗУЛЬТАТЫ РАБОТЫ ПРОГРАММЫ:

1. Пример работы процедуры создания новой записи об изделии и вывода данных в файл:

```

C:\Users\Вика\Documents\VS\CPP\ConsoleApplication2\Debug\Students.exe
---> Меню <---
1 - Прочитать сведения об изделиях из файла
2 - Добавить новую запись
3 - Вывести все данные об изделиях в файл
4 - Выдать информацию об изделиях по коду
5 - Подсчитать суммарное отклонение по единицам измерения
6 - Вывести все сведения об изделиях (в алф. порядке)
7 - Выход
---> Номер действия: 2

---> Добавить новую запись <---

Наименование изделия: Глина

Единица измерения: тонны

Код изделия: 8796

План выпуска: 50

Фактически выпущено: 10

Добавлена запись:
изделие Глина (тонны), код изделия: 8796, план выпуска: 50, фактически выпущено: 10

```

После выполнения программы в файлах ввода (input) и вывода (output) появятся соответствующие записи об изделии:

Содержимое файла ввода:

```
input_5.txt – Блокнот
Файл  Правка  Формат  Вид  Справка
Доска
килограммы
423
500
450
Кирпич
тонны
510
1200
1500
Шифер
килограммы
9
8000
8000
Краска
литры
55
800
1235
Промышленный газ
метр^3
2
3432
5539
Глина
тонны
8796
50
10
```

Содержимое файла вывода:

```
output_5.txt – Блокнот
Файл  Правка  Формат  Вид  Справка
Доска
килограммы
423
500
450
Кирпич
тонны
510
1200
1500
Шифер
килограммы
9
8000
8000
Краска
литры
55
800
1235
Промышленный газ
метр^3
2
3432
5539
Глина
тонны
8796
50
10
```

2. Пример работы процедуры выдачи информации по введенному коду изделия:

```
C:\Users\Вика\Documents\VS\CPP\ConsoleApplication2\x64\Debug\Students.exe
---> Меню <---
1 - Прочитать сведения об изделиях из файла
2 - Добавить новую запись
3 - Вывести все данные об изделиях в файл
4 - Выдать информацию об изделиях по коду
5 - Подсчитать суммарное отклонение по единицам измерения
6 - Вывести все сведения об изделиях (в алф. порядке)
7 - Выход
---> Номер действия: 4
---> Введите код для поиска информации: 8796

!!!! Сведения об изделии с кодом: 8796

Наименование изделия: Глина
Код изделия: 8796
Единицы измерения: тонны
План выпуска: 50
Фактически выпущено: 10
```


3. Пример работы процедуры подсчёта и вывода суммарного отклонения по единицам измерения:

```
C:\Users\Вика\Documents\VS\CPP\ConsoleApplication2\x64\Debug\Students.exe

---> Меню <---

1 - Прочитать сведения об изделиях из файла
2 - Добавить новую запись
3 - Вывести все данные об изделиях в файл
4 - Выдать информацию об изделиях по коду
5 - Подсчитать суммарное отклонение по единицам измерения
6 - Вывести все сведения об изделиях (в алф. порядке)
7 - Выход

---> Номер действия: 5

---> Введите единицы измерения: тонны

!!!! Суммарное отклонение от плана для изделий, (тонны)

Наименование изделия: Кирпич
Код изделия: 510
Суммарное отклонение (недовыполнение): 300
Наименование изделия: Глина
Код изделия: 8796
Суммарное отклонение (перевыполнение): 40
```

4. Пример работы процедуры вывода всех данных в алфавитном порядке:

```
C:\Users\Вика\Documents\VS\CPP\ConsoleApplication2\x64\Debug\Students.exe

---> Меню <---

1 - Прочитать сведения об изделиях из файла
2 - Добавить новую запись
3 - Вывести все данные об изделиях в файл
4 - Выдать информацию об изделиях по коду
5 - Подсчитать суммарное отклонение по единицам измерения
6 - Вывести все сведения об изделиях (в алф. порядке)
7 - Выход

---> Номер действия: 6

!!!! Сведения об изделиях:

1. Наименование изделия: Глина (тонны), код 8796, план выполнения: 50, фактически выпущено: 10
2. Наименование изделия: Доска (килограммы), код 423, план выполнения: 500, фактически выпущено: 450
3. Наименование изделия: Кирпич (тонны), код 510, план выполнения: 1200, фактически выпущено: 1500
4. Наименование изделия: Краска (литры), код 55, план выполнения: 800, фактически выпущено: 1235
5. Наименование изделия: Промышленный газ (метр^3), код 2, план выполнения: 3432, фактически выпущено: 5539
6. Наименование изделия: Шифер (килограммы), код 9, план выполнения: 8000, фактически выпущено: 8000
```