

Федеральное государственное бюджетное образовательное учреждение высшего
образования

Казанский государственный энергетический университет

Кафедра информационных технологий и интеллектуальных систем

Файловая структура

Методические указания для выполнения лабораторных работ



Казань, 2024

Лабораторная работа. Работа с файлами.

Цель работы: научиться создавать пользовательские типы данных в виде структур, освоить приёмы работы со структурными *динамическими массивами* и вывода данных в файловые потоки.

План лабораторной работы

1. Отличие статического и динамического способа выделения памяти.
2. Способы выделения памяти под одномерный массив.
3. Особенности выделения памяти под двумерный массив.
4. Понятие указателя на указатель.
5. Способы передачи массивов в функции.

1. Краткие теоретические сведения о пользовательском типе - структуры.

Объявление структуры

Структура — это тип данных, который может содержать в себе разнотипные элементы. Создается самим пользователем. Шаблон для объявления структуры в языке C++ имеет следующий вид:

```
struct Name
{
    type atrib;
    // остальные элементы структуры
};
```

Объявление структурного типа можно совместить с объявлением соответствующих переменных:

```
struct Name
{
    type atrib;
    // остальные элементы структуры
} structVar1, structVar2, ...;
```

Здесь:

`struct` — ключевое слово, которое начинает определение структуры

`Name` — имя структуры

`type` — тип данных элемента структуры

`atrib` — элемент (поле/атрибут) структуры

`structVar1-2` — структурные переменные

Доступ к элементам структуры

Доступ к элементам структуры реализуется через *оператор доступа* «точка». Предположим, что создана структурная переменная с именем `car` и у нее есть поле с именем `speed`, тогда доступ к нему реализуется так: 1

```
car.speed;
```

Объявление указателя на структуру

Указатель на структуру в C++ объявляется так же, как и указатель на стандартный тип:

```
structName *structVar; // указатель на структуру structName
```

Чтобы получить доступ к элементам структуры через указатель на структуру, вместо оператора «точка», используется оператор «стрелка» `->`:

```
carPtr->speed;
```

Возможности создания и применения структурных переменных рассмотрим на примерах конкретных задач. Комментарии к операторам программ позволят проанализировать код программы и не только понимать ход работы программы, но и включить в свой инструментарий используемые в коде методы.

Для работы со строковыми типами разрешено пользоваться классом `string`.

Пример 1. Создать структурный тип для представления анкеты ребенка, состоящей из его имени, пола и роста. Ввести информацию по заданному количеству детей. Вывести средний рост всех детей.

NB! Обратите внимание на ещё один смены кодовых страниц для букв кириллицы: `SetConsoleCP(1251)` и `SetConsoleCP(866)`;

Решение задачи:

```
#include <iostream> // для консольного ввода-вывода
#include <string> // для переменных типа данных string
#include <Windows.h> // для кодировок (поддержка кириллицы)
#include <iomanip> // для форматированного вывода

// подключаем стандартное пространство имен
using namespace std;

// определяем структуру "Ребенок"
struct Child
{
    string name; // имя ребенка
    char sex; // пол ребенка (М - мужской; Ж - женский)
    int height; // рост ребенка
};
```

```

// запрашивается от пользователя ввод информации о 10 детях
void Input_children(Child* children, int children_count);
// вывод информации обо всех детях на экран в табличном формате
void Print_children(Child* children, int children_count);
// получение среднего роста всех детей
double Get_average_height(Child* children, const int children_count);
int main()
{
    // русификация всех диалогов программы
    setlocale(LC_ALL, "Russian");

    // отвечает за количество детей
    int CHILDREN_COUNT;
    cout << "Ввести количество детей: "; cin >> CHILDREN_COUNT;

    // хранит информацию о детях
    Child *children = new Child[CHILDREN_COUNT];
    Input_children(children, CHILDREN_COUNT);
    Print_children(children, CHILDREN_COUNT);
    // выводим средний рост всех детей
    cout << "Средний рост всех детей: " << setprecision(3) << fixed <<
    Get_average_height(children, CHILDREN_COUNT) << endl << endl;

    system("pause");
    return 0;
}
// запрашивает от пользователя ввод информации о 20 детях
void Input_children(Child* children, int children_count)
{
    cout << "Вам предстоит ввести информацию о " << children_count << " детях." <<
endl;
    for (int i = 0; i < children_count; i++)
    {
        cout << "\t\t Ввод информации о ребенке № " << (i + 1) << endl;

        cout << "\t\t- введите имя: ";
        SetConsoleCP(1251); //ещё один способ смены кодовой страницы

        cin >> children[i].name;
        SetConsoleCP(866);

        cout << "\t\t - введите пол (М - мужской, Ж - женский): ";
        SetConsoleCP(1251);
        cin >> children[i].sex;
        SetConsoleCP(866);

        cout << "\t\t - введите рост (натуральное число), мм: ";
        cin >> children[i].height;
        cout << endl;
    }
}

```

```

// вывод информации обо всех детях на экран в табличном формате
void Print_children(Child* children, const int children_count)
{
    cout << endl << "Информация о детях(" << children_count << " чел.) имеет вид:"
<< endl;
    SetConsoleCP(866);

    cout << "-----" << endl;
    cout << "      №           Имя           Пол           Рост, мм" << endl;
    cout << "-----" << endl;
    for (int i = 0; i < children_count; i++)
    {
        cout << setw(6) << (i + 1) << "\t";
        cout << setw(17) << children[i].name << "\t";
        cout << setw(4) << children[i].sex << "\t";
        cout << setw(12) << children[i].height << endl;
    }
    cout << "-----" << endl;
}

// получение среднего роста всех детей
double Get_average_height(Child* children, const int children_count)
{
    // отвечает за среднее значение возраста всех детей
    double average = 0;

    // перебираем всех детей
    for (int i = 0; i < children_count; i++)
        average += children[i].height; // добавляем возраст текущего ребенка к
результату

    // чтобы получить среднее значение, надо сумму разделить на количество
return (average / (children_count * 1.0));
}

```

Результаты работы программы

```

Ввести количество детей: 3
Вам предстоит ввести информацию о 3 детях.
    Ввод информации о ребенке № 1
    - введите имя: Айрат
    - введите пол (М - мужской, Ж - женский): М
    - введите рост (натуральное число), мм: 145

    Ввод информации о ребенке № 2
    - введите имя: Георгий
    - введите пол (М - мужской, Ж - женский): М
    - введите рост (натуральное число), мм: 137

    Ввод информации о ребенке № 3
    - введите имя: Римма
    - введите пол (М - мужской, Ж - женский): Ж
    - введите рост (натуральное число), мм: 127

Информация о детях(3 чел.) имеет вид:
-----
      №           Имя           Пол           Рост, мм
-----
      1           Айрат           М           145
      2           Георгий          М           137
      3           Римма           Ж           127
-----
Средний рост всех детей: 136.333

```

Пример 2. Создать структурный тип, описывающий студенческую ведомость (Ф.И.О., оценки за три экзамена, средний балл). Заполнить структурный массив $n > 2$ записями. Переписать из исходного массива в другой массив, информацию только о тех студентах, фамилии которых оканчиваются на «ова». Затем новый массив отсортировать по среднему баллу.

Пример 1 показал, насколько громоздко писать код для возможности работать с буквами кириллицы. Поэтому в данном примере обойдёмся латинскими буквами.

NB! Обратите внимание!

1. Объявляется глобальная константа `MARKS_COUNT`, избавляющая от необходимости создавать динамический массив в качестве атрибута структуры и обеспечивающая уменьшение количество формальных параметров.
2. Используется метод возвращения массива из функции по указателю `Statement* Get_ova_statements`.
3. Используется один из методов сортировки массива `Selection_sort`.
4. Для передачи константной строки в функцию необходимо использовать строковую константу `void Print_statements(Statement statements[], int STATEMENTS_COUNT, const string message)`; Эта функция печати вызывается несколько раз с разными заголовками

Решение задачи

```
#include <iostream>           // для консольного ввода вывода
#include <string>              // для работы с типом данных string
#include <iomanip>             // для форматированного вывода данных на экран

// подключаем стандартное пространство имен
using namespace std;

const int MARKS_COUNT = 3; // количество оценок за экзамены у каждого студента

// описание сущности "Студенческая ведомость"
struct Statement
{
    string surname; // фамилия студента
    string name;    // имя студента
    string father_name; // отчество студента
    int marks[MARKS_COUNT]; // оценки студента по 3-ем экзаменам
    double average_ball; // средний балл
};

//Прототипы записываются ПОСЛЕ объявления структурного типа
// ввод информации о студенческих ведомостях
void Input_statements_from_keyboard(Statement statements[], int STATEMENTS_COUNT);
// вывод информации о студенческих ведомостях на экран
```

```

void Print_statements(Statement statements[], int STATEMENTS_COUNT, const string
message);
// определяем кол-во студентов, чьи фамилии заканчиваются на "ova"
int Get_statements_ova_count(Statement statements[], int STATEMENTS_COUNT);
// получаем все ведомости о студентах, чьи фамилии заканчиваются на "ova"
Statement* Get_ova_statements(Statement statements[], int STATEMENTS_COUNT, int
ova_count);
// сортировка ведомостей по среднему баллу по возрастанию
void Selection_sort(Statement ova_statements[], int ova_count);

// главная функция программы (точка входа)
int main()
{
    // количество студенческих ведомостей
    int STATEMENTS_COUNT;
    setlocale(LC_ALL, "Russian");
    cout << "ввести кол-во студентов в группе";
    cin >> STATEMENTS_COUNT;

    // массив, содержащий информацию о студенческих ведомостях
    Statement *statements=new Statement[STATEMENTS_COUNT];

    // вводим информацию о студенческих ведомостях с клавиатуры

    Input_statements_from_keyboard(statements, STATEMENTS_COUNT);

    // выводим информацию обо всех студенческих ведомостях на экран
    Print_statements(statements, STATEMENTS_COUNT, "Информация обо всех
студенческих ведомостях (");

    // считаем количество студентов, чьи фамилии заканчиваются на "ova"
    int ova_count = Get_statements_ova_count(statements, STATEMENTS_COUNT);

    if (ova_count > 0)
    {
        // получаем массив ведомостей студентов, чьи фамилии заканчиваются на
"ova"
        Statement* ova_statements = Get_ova_statements(statements,
STATEMENTS_COUNT, ova_count);

        // выводим информацию о студентах, чьи фамилии заканчиваются на "ova"
        Print_statements(ova_statements, ova_count, "Информация о студентах, чьи
фамилии оканчиваются на \"ova\" (");

        // сортируем ведомости по среднему баллу по возрастанию (методом выбора)
        Selection_sort(ova_statements, ova_count);

        // выводим информацию о студентах, чьи фамилии заканчиваются на "ova",
отсортированных по среднему баллу по возрастанию
        Print_statements(ova_statements, ova_count, "Информация о студентах
(отсортированных по среднему баллу), чьи фамилии оканчиваются на \"ova\" (");
    }
}

```



```

        // забываем удалить память из-под динамического массива о ведомостях
        delete[] ova_statements;
    }
    else
        cout << endl << "В данных студенческих ведомостях нет ни одного
студента, чья фамилия оканчивается на \"ova\"";

    cout << endl << endl;
    system("pause");
    // завершение выполнение программы и передача управления в ОС (код 0 - все
прошло успешно!)
    return 0;
}

// ввод информации о студенческих ведомостях
void Input_statements_from_keyboard(Statement statements[], int STATEMENTS_COUNT)
{
    cout << "Вам предстоит ввести информацию о " << STATEMENTS_COUNT << "
студенческих ведомостях." << endl << endl;
    for (int i = 0; i < STATEMENTS_COUNT; i++)
    {
        cout << "\tВвод информации о студенческой ведомости №" << (i + 1) <<
endl;

        cout << "\t\t - введите фамилию студента: ";
        cin >> statements[i].surname;

        cout << "\t\t - введите имя студента: ";
        cin >> statements[i].name;

        cout << "\t\t - введите отчество студента: ";
        cin >> statements[i].father_name;

        cout << "\t\t - введите через пробел 3 оценки за экзамены (числа от 1 до
5): ";
        cin >> statements[i].marks[0] >> statements[i].marks[1] >>
statements[i].marks[2];

        // после ввода оценок сразу рассчитаем средний балл
        statements[i].average_ball = 0;
        for (int imark = 0; imark < MARKS_COUNT; imark++)
            statements[i].average_ball += statements[i].marks[imark];
        statements[i].average_ball /= (MARKS_COUNT * 1.);
        cout << endl;
    }
}

// вывод информации о студенческих ведомостях на экран
void Print_statements(Statement statements[], int STATEMENTS_COUNT, const string
message)
{
    cout << endl << message << STATEMENTS_COUNT << " шт.) имеет вид:" << endl;
}

```

```

        cout << "-----" << endl;
    -----" << endl;
    cout << "      №                ФИО студента                Оценки за экзамены
Средний балл      " << endl;
    cout << "-----" << endl;
    -----" << endl;
    for (int i = 0; i < STATEMENTS_COUNT; i++)
    {
        cout << setw(6) << (i + 1);
        cout << setw(20) << statements[i].surname << " " <<
statements[i].name[0] << "." << statements[i].father_name[0] << ".";
        cout << setw(24);
        for (int imark = 0; imark < MARKS_COUNT; imark++)
        {
            cout << statements[i].marks[imark];
            if (imark < (MARKS_COUNT - 1))
                cout << ", ";
        }
        cout << setw(22) << setprecision(3) << fixed <<
statements[i].average_ball;
        cout << endl;
    }
    cout << "-----" << endl;
    -----" << endl;
}

// определяем кол-во студентов, чьи фамилии заканчиваются на "ova"
int Get_statements_ova_count(Statement statements[], int STATEMENTS_COUNT)
{
    const string end_search = "ova";
    int count = 0;
    for (int i = 0; i < STATEMENTS_COUNT; i++)
        if ((statements[i].surname.length() >= end_search.length()) &&
            (statements[i].surname.substr(statements[i].surname.length() -
end_search.length()) == end_search))
            count++;

    return count;
}

// получаем все ведомости о студентах, чьи фамилии заканчиваются на "ova"
Statement* Get_ova_statements(Statement statements[], int STATEMENTS_COUNT, int
ova_count)
{
    const string end_search = "ova";
    Statement* ova_statements = new Statement[ova_count];
    int index = 0;
    for (int i = 0; i < STATEMENTS_COUNT; i++)
        if ((statements[i].surname.length() >= end_search.length()) &&
            (statements[i].surname.substr(statements[i].surname.length() -
end_search.length()) == end_search))
            {

```

```

        ova_statements[index] = statements[i];
        index++;
    }

    return ova_statements;
}
// сортировка ведомостей по среднему баллу по возрастанию
void Selection_sort(Statement ova_statements[], int ova_count)
{
    Statement swap;
    int min_index;

    for (int i = 0; i < (ova_count - 1); i++)
    {
        min_index = i;
        for (int j = (i + 1); j < ova_count; j++)
            if (ova_statements[j].average_ball <
ova_statements[min_index].average_ball)
                min_index = j;
        if (i != min_index)
        {
            swap = ova_statements[i];
            ova_statements[i] = ova_statements[min_index];
            ova_statements[min_index] = swap;
        }
    }
}
}

```

Результаты работы программы

Вам предстоит ввести информацию о 3 студенческих ведомостях.

Ввод информации о студенческой ведомости №1

- введите фамилию студента: Rakhimova
- введите имя студента: Roza
- введите отчество студента: Nailevna
- введите через пробел 3 оценки за экзамены (числа от 1 до 5): 4 4 5

Ввод информации о студенческой ведомости №2

- введите фамилию студента: Borzov
- введите имя студента: Oleg
- введите отчество студента: Sergeevich
- введите через пробел 3 оценки за экзамены (числа от 1 до 5): 5 4 5

Ввод информации о студенческой ведомости №3

- введите фамилию студента: Izotova
- введите имя студента: Elena
- введите отчество студента: Markovna
- введите через пробел 3 оценки за экзамены (числа от 1 до 5): 3 3 4

Информация обо всех студенческих ведомостях (3 шт.) имеет вид:

№	ФИО студента	Оценки за экзамены	Средний балл
1	Rakhimova R.N.	4, 4, 5	4.333
2	Borzov O.S.	5, 4, 5	4.667
3	Izotova E.M.	3, 3, 4	3.333

Информация о студентах, чьи фамилии оканчиваются на "ova" (2 шт.) имеет вид:

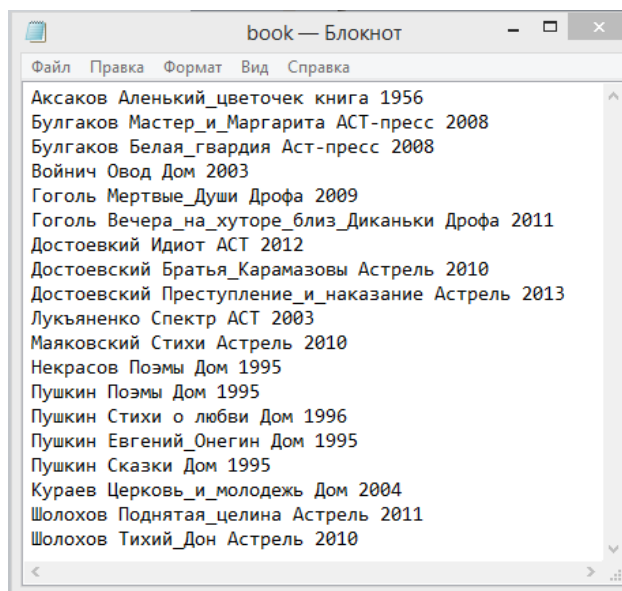
№	ФИО студента	Оценки за экзамены	Средний балл
1	Rakhimova R.N.	4, 4, 5	4.333
2	Izotova E.M.	3, 3, 4	3.333

Информация о студентах (отсортированных по среднему баллу), чьи фамилии оканчиваются на "ova" (2 шт.) имеет вид:

№	ФИО студента	Оценки за экзамены	Средний балл
1	Izotova E.M.	3, 3, 4	3.333
2	Rakhimova R.N.	4, 4, 5	4.333

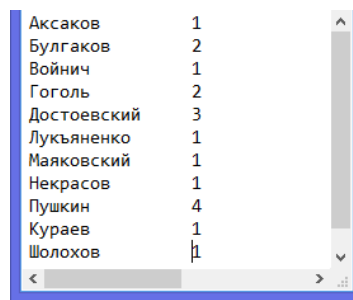
Пример 2 наглядно демонстрирует неудобство работы с консольным вводом при большом наборе входных данных. Следующий, Пример 3, закрепляет навыки формирования размера и инициализации динамического массива внутри функции и показывает пример работы с файловым потоком.

Пример 3. В файле находятся данные со списком книг



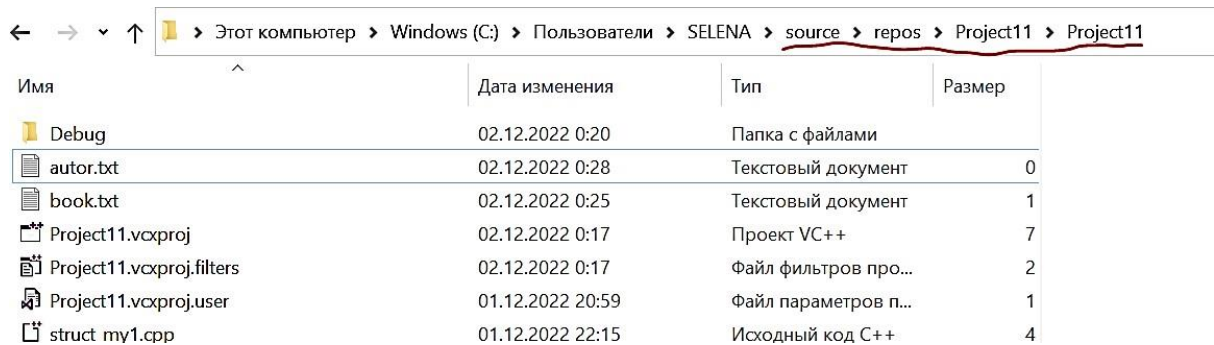
Строка исходного файла содержит следующую информацию о книгах: автор, наименование, издательство, год издания. Все перечисленные параметры отделены пробелами.

Требуется составить перечень авторов с указанием числа их книг. Этот перечень вывести на консоль и в файл.



Выходной файл должен выглядеть так:

При работе с файлами, у которых указано только имя, необходимо размещать их в той директории, где находится exe-файл вашего проекта, по умолчанию, это директория source:



Имя	Дата изменения	Тип	Размер
Debug	02.12.2022 0:20	Папка с файлами	
autor.txt	02.12.2022 0:28	Текстовый документ	0
book.txt	02.12.2022 0:25	Текстовый документ	1
Project11.vcxproj	02.12.2022 0:17	Проект VC++	7
Project11.vcxproj.filters	02.12.2022 0:17	Файл фильтров про...	2
Project11.vcxproj.user	01.12.2022 20:59	Файл параметров п...	1
struct_my1.cpp	01.12.2022 22:15	Исходный код C++	4

Если вы размещаете файл в собственной папке, то необходимо указывать полный путь к файлу, что на C++ делается нестандартно. В данном случае, полный путь к файлу необходимо записать так:

C://Users//SELENA//source//repos//Project11//Project11//book.txt

Поскольку мы не знаем заранее, сколько записей в файле, то необходимо использовать динамический массив структур.

Размер структуры будет определяться в программе чтения данных из файла. Для работы с массивами структур, размеры которых заранее неизвестны, опять используются указатели при передаче структур в функции.

Словесный алгоритм задачи:

- 1 Описать структуру Book, состоящую из 4-х атрибутов согласно содержанию файла
- 2 Создать структуру autor, состоящую из двух атрибутов
- 3 В главной программе создать указатели на построенные структуры
- 4 Создать функцию для чтения данных из файла в массив структур
- 5 Создать функцию для вычисления списка авторов с количеством их книг – перечень авторов
- 6 Создать функцию вывода на консоль
- 7 Создать функцию вывода файл.

Решение задачи имеет следующий код:

```
#include <conio.h>//для _getch()
#include <iostream>
#include <string>
#include <stdlib.h>
#include <fstream>// Библиотека для работы с файлами
#include <iomanip>
using namespace std;
// описание структур
struct book //структура книги
{
    char avtor[15];
    char name[15];
    char izdat[15];
    int god;
};
struct autor //структура автор
{
    char avtor[15];
    int kolvo;
};

// прототипы функций, работающих с динамическими массивами структур
book *inputbookfile(int &k);//прототип функции с возвратом значения типа book - ввод массива структур из файла
autor *perech(book x[], int k, int &k2);//прототип функции типа autor - формирование списка авторов с количеством их книг
void outputperech(autor y[], int k2);//прототип функции - вывод полученного списка на консоль
void outputperechfile(autor y[], int k2);//прототип функции - вывод полученного списка в файл
int main()
{
    setlocale(0, ""); // русский язык на консоли
```

```

book *ps;// указатель типа book
int n;
autor *pp;// указатель типа autor
int l;
ps = inputbookfile(n);// Читаем список книг из файла
pp = perech(ps, n, l);// Составляем перечень авторов
outputperech(pp, l);// выводим полученный перечень на консоль и в файл
outputperechfile(pp, l);
delete[] ps;// освобождаем память, занятую массивами структур
delete[] pp;
system("pause");
}

// коды функций
book *inputbookfile(int &k)//ф-ция с возвращением значения структуры типа book и переменной k из функции
//k - размер массива структур - считываем атрибуты массива структур из файла
{
    int i;
    book t;//переменная типа book
    book *ps;// указатель на тип book
    ifstream fin;// поток для ввода из файла
    char file[10];// имя файла
    cout << "Name of input file:\n";
    cin >> file;// ввести имя файла
    fin.open(file);// открыть файл
    if (!fin) { cout << file << " " << "Can't open file\n"; _getch(); exit(1); }//Функция getch() возвращает очередной
//символ, считанный с консоли, но не выводит этот сим-вол на экран.
// действия при отсутствии файла

    k = 0;
    do {

```

```

        fin >> t.avtor >> t.name >> t.izdat >> t.god;// ввод атрибутов book из файла
        k++;
    } while (fin.good());//предиктная функция good() возвращает true, если чтение из файла прошло успешно.
    k++;// определили число записей k в файле
    fin.close();
    fin.open(file);
    ps = new book[k];// создаем массив структур book из k элементов
    if (ps == NULL) { cout << "No memory\n"; _getch(); exit(1); }
    for (i = 0; i < k; i++)//считывание атрибутов структуры book из файла
        fin >> ps[i].avtor >> ps[i].name >> ps[i].izdat >> ps[i].god;
    fin.close();
    return ps;// возврат массива структур
}
autor *perech(book x[], int k, int &k2)//в функцию передается массив структур типа book x размера k
        //из функции возвращается массив структур типа autor и его размер k2
{
    int i, j, fl;
    autor *t; //указатель на тип autor
    t = new autor[k];//массив структур типа autor размером k(это максим. размер - для случая, если ВСЕ книги ОДНОГО
автора
    if (t == NULL) { cout << "No memory"; _getch(); exit(1); }
    k2 = 0;
    for (i = 0; i < k; i++)
    {
        fl = 0;
        for (j = 0; j < k2; j++)
            if (strcmp(x[i].avtor, t[j].avtor) == 0){ fl = 1; t[j].kolvo++;}// если в массиве структур book атрибут
.avtor = соответствующему
        // атрибуту структуры autor, то увеличивается атрибут .kolvo в массиве структур t типа autor, флаг fl включен
        if (fl == 0)

```



```

        {
            strcpy_s(t[k2].avtor, x[i].avtor); // копируем автора в массив t из список книг, если автор встретился впервые
            t[k2].kolvo = 1;
            k2++;
        }
    }
    return t;
}
void outputperech(avtor y[], int k2) // функция вывода полученного списка авторов на консоль
{
    int i;
    cout << "АВТОР          Koli4ECTBO\n";
    for (i = 0; i < k2; i++)
        cout << y[i].avtor << "    " << y[i].kolvo << endl;
    return;
}
void outputperechfile(avtor y[], int k2) // функция вывода полученного списка авторов в файл
{
    ofstream fout; // поток для вывода
    int i;
    char file[10]; // имя файла для вывода
    cout << "Name of output file:\n";
    cin >> file;
    fout.open(file);
    if (!fout.good()) { cout << file << "Not created\n"; _getch(); exit(1); } // если что-то не в порядке
                                        //с файлом, то сообщение и выход из программы

    for (i = 0; i < k2; i++)
        fout << setw(15) << y[i].avtor << setw(10) << y[i].kolvo << endl;
    fout.close();
    return;
}

```

3. Методика выполнения самостоятельной работы

1. Ознакомиться с условием задачи и примерами решения аналогичных задач из методической части данной лабораторной работы. Особое внимание уделить возможности динамического выделения памяти. Оценить преимущества выделения однотипных операций с массивами в отдельные функции.
2. Для всех типов задач необходимы составить контрольные примеры, на основании которых построить упрощенную модель описываемой ситуации.
3. Проверить полноту задачи: рассмотреть все возможные исходы решения в зависимости от исходных данных, предусмотреть случаи возможного зависания, заикливания программы и запрограммировать корректную реакцию программы на эти ситуации.
4. Записать словесный алгоритм или составить блок-схему алгоритма.
5. Записать код программы на C++.
6. Запустить программу, провести синтаксическую отладку.
7. Проверить работоспособность программы путём сравнения результатов с контрольным примером на все возможные случаи исходных данных.
8. Завершить работу составлением Отчёта, где будут описаны все этапы выполнения самостоятельного задания и приведены распечатки консольного вывода.

4. Задания для самостоятельной работы

Задание 1. Создать структурный тип (структуру) в соответствии со своим вариантом и решить поставленную задачу, используя структурный подход программирования. Значения для строчных переменных задавать буквами кириллицы (см. Пример 1). Номер варианта определяется исходя из **15** заданий.

1. Создать структурный тип для представления анкеты ребенка, состоящей из его имени, пола и роста. Ввести информацию по N детям. Вывести средний рост мальчиков.

2. Составить программу, выводящую на экран ведомость начисленной заработной платы (Ф.И.О., должность, год и дата рождения, заработная плата). Найти среднюю зарплату. И вывести фамилии с зарплатой выше средней.

3. Создать структурный тип для представления информации по горным вершинам, состоящей из названия вершины и ее высоты. Ввести информацию по 20 вершинам. Вывести среднее значение высот всех 20 вершин. Далее вывести названия всех вершин ниже среднего.

4. Составить программу, выводящую на экран ведомость начисленной заработной платы (Ф.И.О., должность, года и дата рождения, заработная плата). Вывести фамилию сотрудника с самой маленькой зарплатой.

5. Составить программу, выводящую на экран ведомость начисленной заработной платы (Ф.И.О., должность, года и дата рождения, заработная плата). Вывести фамилию сотрудника с самой большой зарплатой.

6. Создать структурный тип для представления анкеты ребенка, состоящей из его имени, пола и роста. Ввести информацию по N детям. Вывести имя самой высокой девочки.

7. Создать структурный тип для представления информации по горным вершинам, состоящей из названия вершины и ее высоты. Ввести информацию по 50 вершинам. Вывести название самой низкой вершины из всех 50.

8. Создать структурный тип для представления анкеты ребенка, состоящей из его имени, пола и роста. Ввести информацию по N детям. Вывести средний рост девочек.

9. Создать структурный тип для представления информации по горным вершинам, состоящей из названия вершины и ее высоты. Ввести информацию по N вершинам. Вывести среднее значение высот всех N вершин. Далее вывести названия всех вершин выше среднего.

10. Создать структурный тип для представления информации по горным вершинам, состоящей из названия вершины и ее высоты. Ввести информацию по N вершинам. Вывести название самой высокой вершины из всех N.

11. Создать структурный тип для представления анкеты ребенка, состоящей из его имени, пола и роста. Ввести информацию по N детям. Вывести средний рост девочек. Далее вывести имена всех девочек выше среднего.

12. Создать структурный тип для представления информации по горным вершинам, состоящей из названия вершины и ее высоты. Ввести информацию по N вершинам. Вывести среднее значение высот всех N вершин. Вывести название самой маленькой вершины из всех N вершин.

13. Создать структурный тип для представления анкеты ребенка, состоящей из его имени, пола и роста. Ввести информацию по N детям. Вывести имя самого высокого мальчика. Вывести средний рост мальчиков. Далее вывести имена всех мальчиков ниже среднего.

14. Составить программу, выводящую на экран ведомость начисленной заработной платы (Ф.И.О., должность, год и дата рождения, заработная плата). Вывести все фамилии, начинающиеся на букву <A> и их зарплату.

15. Составить программу, выводящую на экран ведомость начисленной заработной платы (Ф.И.О., должность, год и дата рождения, заработная плата). Вывести дату рождения сотрудника с самой маленькой зарплатой.

Задание 2. Создать структурный тип (структуру) в соответствии со своим вариантом и решить поставленную задачу, используя структурный подход программирования. Значения для строчных переменных можно задавать буквами латиницы (см. Пример 2) Номер варианта определяется исходя из **14** заданий.

1. Создать структурный тип, описывающий гостиничный номер (название гостиницы, номер, комфортность (люкс, полулюкс, стандарт, эконом), количество человек, стоимость). Заполнить структурный массив $N > 2$ записями. Переписать из исходного массива в другой массив, информацию только о тех гостиничных номерах, название гостиницы которых начинается с сочетания букв «City». Затем новый массив отсортировать по номеру.

2. Создать структурный тип, описывающий расписание полетов самолетов (пункт назначения, время отправления, время прибытия, время полета, стоимость билета). Заполнить структурный массив $N > 2$ записями. Переписать из исходного массива в другой массив, информацию только о тех рейсах, пункт назначения которых содержит по 2 буквы <a>. Затем новый массив отсортировать по пункту назначения по алфавиту.

3. Создать структурный тип, описывающий гостиничный номер (название гостиницы, номер, комфортность (люкс, полулюкс, стандарт, эконом), количество человек, стоимость). Заполнить структурный массив $N > 2$ записями. Переписать из исходного массива в другой массив, информацию только о тех гостиничных номерах, в названии гостиницы которых есть по 2 буквы <a>. Затем новый массив отсортировать по названию гостиницы по алфавиту.

4. Создать структурный тип, описывающий музыкальные DVD-диски (название альбома, исполнитель, стиль, год выпуска, длительность, стоимость). Заполнить структурный массив $N > 2$ записями. Переписать из исходного массива в другой массив, информацию только о тех DVD-дисках, название альбома которых начинается на сочетание букв (3—4) введенных пользователем. Затем новый массив отсортировать по стилю алфавита.

5. Создать структурный тип, описывающий книги домашней библиотеки (автор, название книги, издательство, год издания, стоимость). Заполнить структурный массив $N > 2$ записями. Переписать из исходного массива в другой массив, информацию только о тех книгах, в названии которых есть по 3 буквы <o>. Затем вывести информацию, отсортированную по названию издательства по алфавиту.

6. Создать структурный тип, описывающий книги домашней библиотеки (автор, название книги, издательство, год издания, стоимость). Заполнить структурный массив $N > 2$ записями. Переписать из исходного массива в другой массив, информацию только о тех книгах, в названии

издательства которых есть 1 буква <к>. Затем новый массив отсортировать по названию книги по алфавиту.

7. Создать структурный тип, описывающий гостиничный номер (название гостиницы, номер, комфортность (люкс, полулюкс, стандарт, эконом), количество человек, стоимость). Заполнить структурный массив $N > 2$ записями. Переписать из исходного массива в другой массив, информацию только о тех гостиничных номерах, название гостиницы которых начинается на букву <P>. Затем новый массив отсортировать по возрастанию стоимости.

8. Создать структурный тип, описывающий книги домашней библиотеки (автор, название книги, издательство, год издания, стоимость). Заполнить структурный массив $N > 2$ записями. Переписать из исходного массива в другой массив, информацию только о тех книгах, название которых начинается на «Фент». Затем новый массив отсортировать по стоимости.

9. Создать структурный тип, описывающий гостиничный номер (название гостиницы, номер, комфортность (люкс, полулюкс, стандарт, эконом), количество человек, стоимость). Заполнить структурный массив $N > 2$ записями. Переписать из исходного массива в другой массив, информацию только о тех гостиничных номерах, название гостиницы которых оканчивается на сочетание букв «plaza». Затем новый массив отсортировать по возрастанию стоимости.

10. Создать структурный тип, описывающий музыкальные DVD-диски (название альбома, исполнитель, стиль, год выпуска, длительность, стоимость). Заполнить структурный массив $N > 2$ записями. Переписать из исходного массива в другой массив, информацию только о тех DVD-дисках, название стиля которых начинается на сочетание букв (3—4) введенных пользователем. Затем новый массив отсортировать по исполнителю по алфавиту.

11. Создать структурный тип, описывающий гостиничный номер (название гостиницы, номер, комфортность (люкс, полулюкс, стандарт, эконом), количество человек, стоимость). Заполнить структурный массив $N > 2$ записями. Переписать из исходного массива в другой массив, информацию только о тех гостиничных номерах, название гостиницы которых оканчивается на сочетание букв «hostel». Затем новый массив отсортировать по комфортности по алфавиту.

12. Создать структурный тип, описывающий книги домашней библиотеки (автор, название книги, издательство, год издания, стоимость). Заполнить структурный массив $N > 2$ записями. Переписать из исходного массива в другой массив, информацию только о тех книгах, название издательства которых начинается на «Ф». Затем новый массив отсортировать по названию книги по алфавиту.

13. Создать структурный тип, описывающий музыкальные DVD-диски (название альбома, исполнитель, стиль, год выпуска, длительность, стоимость). Заполнить структурный массив $N > 2$ записями. Переписать из исходного массива в другой массив, информацию только о тех DVD-дисках, исполнитель которых начинается на букву <Б>. Затем новый массив отсортировать по стоимости.

14. Создать структурный тип, описывающий расписание полетов самолетов (пункт назначения, время отправления, время прибытия, время полета, стоимость билета). Заполнить структурный массив $N > 2$ записями. Переписать из исходного массива в другой массив, информацию только о тех рейсах, пункт назначения которых оканчивается сочетанием «град». Затем новый массив отсортировать по времени полета.

Задание 3. Создать структурный тип (структуру) в соответствии со своим вариантом и решить поставленную задачу, используя структурный подход программирования. Файлы, содержащие информацию для решения задачи, должны иметь не менее 10 записей. В тексте отчёта показать исходный и выходной файлы. Имена функций и переменных должны иметь mnemonic значение (см. Пример 3). Номер варианта определяется исходя из **12** заданий.

1. Дана информация о студентах, записанная в файле. Число записей заранее неизвестно. Запись имеет вид: фамилия, год рождения, факультет. Составить список факультетов с количеством студентов на каждом из них новая структура: факультет и кол-во студентов на нем

2. Дана информация о школах, записанная в файле. Число записей заранее неизвестно. Запись имеет вид: номер школы, год, количество выпускников, число поступивших в ВУЗы. Составить список школ, в которых процент поступивших в ВУЗ выше $n\%$ – новая структура: школа и процент поступивших.

3. Дана информация о комнатах в общежитии, записанная в файле. Запись имеет вид: фамилии, номер комнаты, факультет, площадь. Получить список факультетов с количеством студентов, проживающих в общежитии – новая структура: факультет, кол-во студентов.

4. Дана информация о рабочих цеха, записанная в файле. Запись имеет вид: фамилия, размер зарплаты, стаж работы. Составить список рабочих, размер зарплаты которых не выше средней по списку: новая структура фамилия, зарплата, разница со средней зарплатой.

5. Дана информация о вузах, записанная в файле. Запись имеет вид: название вуза, число студентов, количество факультетов, количество кафедр.

Составить список вузов, в которых среднее количество кафедр на факультет не превышает числа N . Новая структура: вуз, среднее количество кафедр.

6. Дана информация о вкладчиках в сберкассы, записанная в файле. Запись имеет вид: фамилия, район, социальное положение (рабочий, служащий, крестьянин, пенсионер), величина вклада. Составить список вкладчиков-пенсионеров, у которых вклад ниже среднего по списку. Новая структура: фамилия, вклад, разница вклада со средним по списку.

7. Дана информация о больных, записанная в файле. Запись имеет вид: фамилия, возраст, пол, давление. Составить список больных с повышенным давлением (больше 140) и найти в нем человека с наивысшим давлением.

8. Дана информация о квартирах, записанная в файле. Запись имеет вид: фамилия владельца, площадь, число комнат, этаж. Составить список квартир со средней площадью комнаты, меньше S кв.м. Новая структура: фамилия, средняя площадь комнаты.

9. Дана информация о студентах, записанная в файле. Запись имеет вид: фамилия, год рождения, место рождения, факультет. Составить список студентов старше 20 лет и найти в нем самого молодого студента.

10. Дана информация о школах, записанная в файле. Запись имеет вид: номер школы, год, количество выпускников, число поступивших в ВУЗы. Составить список школ, в которых есть поступившие в вуз в текущем году. Новая структура: школа, число поступивших в вуз. Найти школу с наивысшим показателем в текущем году.

11. Дана информация о комнатах в общежитии, записанная в файле. Запись имеет вид: фамилии, номер комнаты, факультет, площадь. Составить список студентов с заданного факультета и найти среди них тех, кто живет в комнате с наименьшей площадью. Новая структура: фамилия, площадь комнаты.

12. Дана информация о квартирах, записанная в файле. Запись имеет вид: фамилия владельца, количество членов семьи, площадь квартиры, этаж. Составить список владельцев на заданном этаже и найти среди них того, у кого наиболее стесненные условия по площади. Новая структура: фамилия, площадь на 1 человека.

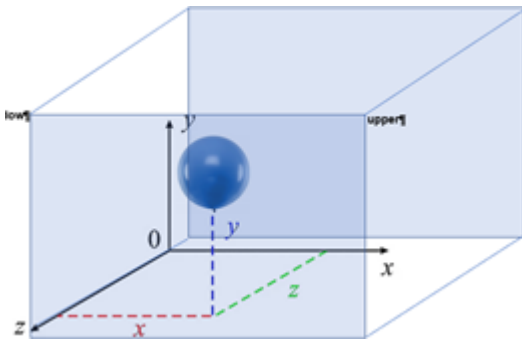
* * *

5. Контрольные вопросы

1. Отличие структуры от массива.
2. Как реализуется доступ к атрибутам структуры?
3. Перечислите функции работы с файлами?
4. Каким образом программа узнаёт, что достигнут конец файла?
5. Как из функции передать массив структур – два способа?

6. Домашнее задание

Задача 1. В рамках структурного подхода создать программу по обстреливанию мишени-шара, заданного своими координатами центра и радиусом, расположенного внутри куба, стороны которого параллельны осям координат. Вывести количество сделанных выстрелов и количество промахов.



1. Параметры шара: центр в точке (a, b, c) . Радиус R
2. Условие попадания точки в шар:
$$(x - a)^2 + (y - b)^2 + (z - c)^2 \leq R^2$$
3. Координаты пуль получаем случайным образом в диапазон от low до upper: $x \in [\text{low}.x; \text{upper}.x]$ и т.д
4. Для удобства разработки создадим две структурных переменных:
Точка – point - с тремя координатами
Шар – sphere – три координаты центра и радиус

Задача 2. Создать структуру для представления информации о кости домино, состоящей из левой половинки и правой половинки. Поля «левая» и «правая» половинки хранят информацию о количестве точек на половинках. Описать массив из 28 элементов (кости домино). Заполнить массив случайными числами или ввести его значения из файла. Исключить из массива одинаковые кости, если они в нем появились в ходе инициализации. Отсортировать оставшиеся элементы массива так, чтобы правая цифра очередной кости равнялась левой цифре следующей кости. Составить контрольный пример для этой задачи.

<http://www.proglabs.ru/russia/r/rgups/si-plus-plus-igniteva/lab6/?unapproved=161&moderation-hash=2fb5d5b0e9c2f828f8c3c8d92bab60d9#comment-161>