

Тема 10: СТРУКТУРЫ. РАБОТА С ФАЙЛАМИ

Цель занятия: получение практических навыков работы с файлами; программирования задач с использованием структур.

Теоретические сведения

Структуры

Структура является набором элементов произвольных типов.

Например:

```
struct address
{
    char* name;
    long int number;
    char* street;
    char* town;
};
```

Здесь обязательна ; после }.

Переменные типа объявленной структуры могут объявляться обычным образом:

```
void f()
{
    address jd;           // jd – переменная структуры
    jd.name = "Jim Dandy"; // присвоить значение полям структуры
    jd.number = 63;
}
```

Доступ к элементам структур обеспечивается с помощью уточненных имен (имя_структуры.имя_элемента): `jd.name`, `jd.number`.

Инициализация может выполняться так:

```
address jd = {"Jim Dandy", 63, "South St", "New Providence"};
```

Значения полей указываются в порядке их объявления в структуре.

Работа с файлами

Файловый ввод-вывод ничем не отличается от консольного. За единственным исключением – если данные читаются из файла, то в любой момент можно вернуться к началу файла и считать все заново.

Для того, чтобы в C++ работать с файлами, необходимо подключить заголовочный файл **fstream**:

```
#include <fstream>
```

После этого можно объявлять объекты, привязанные к файлам: для чтения данных из файла используются объекты типа *ifstream* (аббревиатура от *input file stream*, для записи данных в файл используются объекты типа *ofstream* (*output file stream*).

Например:

```
ifstream in;           // Поток in будет использоваться для чтения
ofstream out;          // Поток out будет использоваться для записи
```

Чтобы привязать тот или иной поток к файлу (открыть файл для чтения или для записи) используется метод **open**, которому необходимо передать параметр – текстовую строку, содержащую имя открываемого файла.

```
in.open("input.txt");
out.open("output.txt");
```

После открытия файлов и привязки их к файловым потокам, работать с файлами можно так же, как со стандартными потоками ввода-вывода *cin* и *cout*. Например, чтобы вывести значение переменной *x* в поток *out*, используется следующая операция:

```
out<<x;
```

А чтобы считать значение переменной из потока *in*

```
in>>x;
```

Для закрытия ранее открытого файла используется метод **close()** без аргументов:

```
in.close();
out.close();
```

Закрытый файловый поток можно открыть заново при помощи метода **open**, привязав его к тому же или другому файлу.

При считывании данных из файла может произойти ситуация достижения конца файла (**end of file**, сокращенно **EOF**). После достижения конца файла никакое чтение из файла невозможно. Для того, чтобы проверить состояние файла, необходимо вызвать метод **eof()**. Данный метод возвращает **true**, если достигнут конец файла или **false**, если не достигнут. Кроме того, состояние файлового потока можно проверить, если просто использовать идентификатор потока в качестве логического условия:

```
if (in)
{
}
```

Также можно использовать в качестве условия результат, возвращаемый операцией считывания. Если считывание было удачным, то результат считается истиной, а если неудачным – ложью. Например, организовать считывание последовательности целых чисел можно так:

```
int d;
while(in>>d)
{
}
```

А организовать считывание файла построчно (считая, что строка заканчивается символом перехода на новую строку) так:

```
char S[100];
while ( getline(in,S))
{
}
```

Пример 10.1. Создать структуру Картина с элементами Название, Автор, Цена. Информацию о нескольких картинах прочитать из текстового файла.

Пример содержимого текстового файла (значения должны быть разделены пробелами, табуляциями или переводом строки):

```
Dali Vilabertran 20000000.00
Malevich Woodcutter 34560000.00
Klimt Sunflower 16500000.00
```

```
#include "stdafx.h"
#include <iostream>
#include <fstream> //для работы с файлами
#include <windows.h>
using namespace std;
struct painting
{
    char title[100];
    char author[100];
    float price;
};
int main()
{
    SetConsoleOutputCP(1251);
    SetConsoleCP(1251);
    ifstream fin; //файл для чтения
    fin.open("e:\\инф.txt");
    if(!fin.is_open()) //если не удалось открыть файл
    {
        cout<<"Файл не найден";
        return 1;
    }
    painting pic1;
    painting pic2;
    //записать в поля структур значения из файла
    fin>>pic1.author>>pic1.title>>pic1.price;
    fin>>pic2.author>>pic2.title>>pic2.price;
    //вывод значений полей структуры
    cout<<"Картина 1: "<<pic1.title<<" Художник: "<<pic1.author<<" Стоимость: "<<
" Цена: "<<pic1.price<<endl;
    cout<<"Картина 2: "<<pic2.title<<" Художник: "<<pic2.author<<" Стоимость: "<<
```

```
" Цена: "<<pic2.price<<endl;
//закрывать файл
fin.close();
return 0;
}
```

```
Картина 1: ViIabertran Художник: Dal1 Стоимость: Цена: 2e+007
Картина 2: Woodcutter Художник: Malevich Стоимость: Цена: 3.456e+007
```

Пример 10.2. Создать структуру Дробь с элементами а – числитель, b – знаменатель. Реализовать с дробями арифметические сложение.

```
#include "stdafx.h"
#include <iostream>
#include <fstream> //для работы с файлами
#include <conio.h>
#include <windows.h>
using namespace std;
float Ppol(float c1,float c2,float z1,float z2)
{
    float f = 0;
    if(z1==z2) f=(c1+c2)/z1; else
        if(z1!=z2) f=(c1*z2+c2*z1)/(z1*z2);
    return f;
}
struct tr
{
    float a,b;
};
int main()
{
    SetConsoleOutputCP(1251);
    SetConsoleCP(1251);
    ifstream fin; //файл для чтения
    fin.open("e:\\x.txt");
    if(!fin.is_open()) //если не удалось открыть файл
    {
        cout<<"Файл не найден";
        return 1;
    }
    tr d1,d2;
    while(!fin.eof()) //пока не конец файла
    {
        fin>>d1.a>>d1.b; //записать в поля структур значения из файла
        cout<<"дробь 1: "<<d1.a<<"/"<<d1.b<<endl;
        fin>>d2.a>>d2.b; //записать в поля структур значения из файла
        cout<<"дробь 2: "<<d2.a<<"/"<<d2.b<<endl;
        float s = Ppol(d1.a,d2.a,d1.b,d2.b);
        cout<<"сумма="<<s<<endl;
    }
    fin.close(); //закрывать файл
    getch();
    return 0;
}
```

```
дробь 1: 2/3
дробь 2: 1/3
сумма=1
дробь 1: 3/4
дробь 2: 6/2
сумма=3.75
дробь 1: 7/4
дробь 2: 6/8
сумма=2.5
```

Задание 10.1. Создать структуру Треугольник с тремя элементами а, b, с – длины 3-х сторон. Для N заданных треугольников найти: k1 – количество равносоставленных треугольников и k2 – количество прямоугольных треугольников.

Пример 10.3. Создать структуру Книга: название, имя автора, цена, количество экземпляров книги в книжном магазине. Найти самую дорогую книгу. Создать новый файл с информацией о книгах, упорядоченной по полю «количество экземпляров» (отсортировать массив структур).

```
#include "stdafx.h"
#include <fstream>
#include <iostream>
#include <iomanip>
#include <string.h>
#include <conio.h>
#include <windows.h>
using namespace std;
struct book
{
    char title[100];
    char author[100];
    float price;
    int count;
};
// сортировка массива структур по полю «количество экземпляров»
void sort(book arr[], int size)
{
    for (int i=size-1; i > 0; --i)
        for (int j=0; j < i; ++j)
            if (arr[j].count>arr[j+1].count)
            {
                book tmp = arr [j];
                arr [j] = arr [j+1];
                arr [j+1] = tmp;
            }
}
// поиск самой дорогой книги
void max_price(book arr[], int size)
{
    float max=arr[0].price;
    char max_title[100];
    strcpy(max_title,arr[0].title);
    char max_author[100];
    strcpy(max_author,arr[0].author);
    for (int j=0; j < size; ++j)
        if (max<arr[j].price)
        {
            max=arr[j].price;
            strcpy(max_title,arr[j].title);
            strcpy(max_author, arr[j].author);
        }
}
cout<<"\nСАМАЯ ДОРОГАЯ КНИГА - "<<max_title<<" "<<max_author<<" ЕЕ ЦЕНА "<<max<<"\n";
}
int main()
{
    SetConsoleCP(1251);
    SetConsoleOutputCP(1251);
    ifstream fin("c:\\my_books.txt"); //исходный файл
    ofstream bookf; //файл для записи отсортированного массива
    bookf.open("c:\\BOOKINFO.txt");
    if(!fin.is_open())
    {
        cout<<"Файл не найден";
        return 1;
    }
    book b[10]; //массив из 10 структур book
    int N=0; //первое значение в файле - размер массива структур
```

```

fin>>N;
for(int i = 0; i < N; i++)
{
    fin.ignore();          //пропустить символ '\n' (значения в файле разделены переводом строки)
    fin.getline(b[i].title,100,'\n');    //прочитать строку с названием книги
    fin.getline(b[i].author,100, '\n');    //прочитать строку с именем автора
    fin>>b[i].price;        // прочитайте строку с ценой книги
    fin>>b[i].count;        // прочитайте строку с числом экземпляров книги
}
cout<<"ИСХОДНЫЙ МАССИВ\n\n";
cout<<setw(25)<<left<<"title"<<setw(10)<<left<<"author"<<"\t"<<"price"<<"\t"<<"count"<<"\n\n";
//вывод массива на экран
for(int i = 0; i < N; i++)
cout<<setw(25)<<left<<b[i].title<<setw(10)<<left<<b[i].author<<"\t"<<b[i].price<<"\t"<<b[i].count<<"\n";
sort(b, N);
cout<<"\nМАССИВ ОТСОРТИРОВАННЫЙ ПО ПОЛЮ count\n\n";
//вывод массива на экран и запись в файл
for(int i = 0; i < N; i++)
{
    cout<<setw(25)<<left<<b[i].title<<setw(10)<<left<<b[i].author<<"\t"<<b[i].price<<"\t"<<b[i].count<<"\n";
    bookf<<setw(25)<<left<<b[i].title<<setw(10)<<left<<b[i].author<<"\t"<<b[i].price<<"\t"<<b[i].count<<"\n";
}
max_price(b, N);
fin.close();          //закрыть файл
bookf.close();
getch();
return 0;
}

```

Методические указания

При подготовке к занятию необходимо изучить способы описания структур; особенности файлового ввода-вывода.

Аудиторные и домашние задания

1. Создать структуру Дата с элементами День, Месяц, Год. Для N человек с заданной датой рождения вывести сообщение, в каком сезоне и в каком веке родился человек.
2. Создать структуру Треугольник с тремя элементами a, b, c – длины 3-х сторон. Найти, у какого из N заданных треугольников площадь наибольшая?
3. Информацию о наличии на складе автомашин можно представить с помощью таких полей: модель (marka), год выпуска (year), цвет (color) и цена машины (price). Описать структуру avto. Создать файл из шести-семи записей, который содержит информацию об автомобилях. Вывести содержимое файла на экран для контроля. Используя созданный файл, составить программу для получения следующих сведений:
 - а) Определить количество и вывести на экран названия моделей, которые выпускались в 1999 году.
 - б) Вывести в файл Y информацию о машинах черного цвета, выпущенных за последние пять лет.
 - в) Получить информацию о самой дорогой и самой дешевой машине.
 - г) Получить информацию о самой старой и самой современной машине.
 - д) Получить информацию о машинах данной модели и данного цвета.
4. Дан файл f, содержащий сведения о веществах: название вещества, его удельный вес и проводимость (проводник, полупроводник, изолятор). Переписать в новый файл удельные веса и названия всех полупроводников.
5. Дан файл, содержащий сведения о сдаче студентами сессии: индекс группы, фамилия студента, оценки по пяти экзаменам. Определить средний балл, полученный каждым студентом группы X, и всей группой в целом.
6. Дан файл, содержащий сведения о телефонах абонентов: фамилия, год установки телефона, номер телефона. Определить количество установленных телефонов с XXXX года и список абонентов.
7. Дан файл f, содержащий даты: число, месяц, год.
 - а) Найти год с наименьшим номером.
 - б) В файл g переписать осенние даты с четными числами, а в файл h – все весенние с нечетными числами.
8. Дан файл f, содержащий сведения о пациентах глазной клиники: фамилия, пол, возраст, место проживания (город), диагноз. Получить в файле g список пациентов старше X лет с диагнозом Y.

9. Дан файл f , содержащий сведения об экспортируемых товарах: указывается наименование товара, страна, импортирующая товар, и объем поставляемой партии в штуках. Найти страны, в которые экспортируется данный товар, и общий объем его экспорта.
10. Дан файл, содержащий данные об автомобилях. Выполнить следующую обработку данных:
 - а) Вывести автомобили, произведенные в 1999 или 2001 году.
 - б) Переписать в новый файл номера, марки и стоимости автомобилей заданного цвета.
11. Задан файл, содержащий фамилию, имя, отчество, группу и оценки студентов за последнюю сессию.
 - а) Вывести фамилии и инициалы студентов, сдавших сессию без троек.
 - б) Переписать в новый файл фамилии и названия групп тех студентов, которые имеют хотя бы одну задолженность.
12. Дан файл, содержащий различные даты (год, месяц, число). Определить самую позднюю дату.
13. Дан файл, содержащий информацию о багаже пассажиров (номер квитанции, количество вещей, общий вес багажа).
 - а) Найти число пассажиров, имеющих более двух вещей и число пассажиров, количество вещей которых превосходит среднее количество вещей.
 - б) Выяснить, имеется ли пассажир, багаж которого превышает багаж каждого из остальных пассажиров и по числу вещей, и по весу.
14. Для файла, содержащего данные об автомобилях, выполнить следующую обработку данных:
 - а) Подсчитать среднюю стоимость автомобилей и самый дорогой автомобиль каждого цвета.
 - б) Определить самую популярную марку автомобиля.
15. Задан файл с полями: номер видеокассеты, название фильма, фамилия режиссера, год выпуска фильма.
 - а) Подсчитать, сколько фильмов каждого года выпуска хранится в файле.
 - б) Определить, какой режиссер чаще других встречается в файле.
 - в) Вывести все фильмы в порядке убывания номера кассеты.
16. Задан файл с полями: название музея, страна, город, количество экспонатов. Определить в какой стране больше всего музеев и в каком музее этой страны больше всего экспонатов.

Контрольные вопросы

1. Что такое структура?
2. Что такое поля структуры?
3. Какого типа могут быть поля структуры?
4. Правила обращения к полям структуры.
5. Какой заголовочный файл необходимо подключить для работы с файлами?
6. Какие методы используют для открытия и закрытия файлов?
7. Как проверить достижение конца файла?

```

Co string title;
   string author;
#include "stdafx.h"
#include <fstream>
#include <iostream>
#include <iomanip>
#include <string.h>
#include <conio.h>
#include <windows.h>
using namespace std;
struct book
{
    string title;
    string author;
    float price;
    int count;
};
// сортировка массива структур по полю «количество экземпляров»
void sort(book arr[], int size)
{
    for (int i=size-1; i > 0; --i)
        for (int j=0; j < i; ++j)
            if (arr[j].count>arr[j+1].count)
            {
                book tmp = arr [j];

```

```

        arr[j] = arr[j+1];
        arr[j+1] = tmp;
    }
}
// поиск самой дорогой книги
void max_price(book arr[], int size)
{
    float max=arr[0].price;
    string max_title=arr[0].title;
    string max_author=arr[0].author;
    for (int j=0; j < size; ++j)
        if (max<arr[j].price)
        {
            max=arr[j].price;
            max_title=arr[j].title;
            max_author=arr[j].author;
        }
    cout<<"\nСАМАЯ ДОРОГАЯ КНИГА - ";
    cout<<max_title;
    cout<<" "<<max_author<<" ЕЕ ЦЕНА " <<max<<"\n";
}
int main()
{
    SetConsoleCP(1251);
    SetConsoleOutputCP(1251);
    ifstream fin("c:\\my_books.txt"); //исходный файл
    ofstream bookf; //файл для записи отсортированного массива
    bookf.open("c:\\BOOKINFO1.txt");
    if(!fin.is_open())
    {
        cout<<"Файл не найден";
        return 1;
    }
    book b[10]; //массив из 10 структур book
    int N=0; //первое значение в файле - размер массива структур
    fin>>N;
    for(int i = 0; i < N; i++)
    {
        fin.ignore(); //пропустить символ '\n' (значения в файле разделены переводом строки)
        getline(fin,b[i].title,'\n'); //прочитать строку с названием книги
        getline(fin,b[i].author,'\n'); //прочитать строку с именем автора
        fin>>b[i].price; // прочитайте строку с ценой книги
        fin>>b[i].count; // прочитайте строку с числом экземпляров книги
    }
    cout<<"ИСХОДНЫЙ МАССИВ\n\n";
    cout<<setw(25)<<left<<"title"<<setw(10)<<left<<"author"<<"\t"<<"price"<<"\t"<<"count"<<"\n\n";
    //вывод массива на экран
    for(int i = 0; i < N; i++)
        cout<<setw(25)<<left<<b[i].title<<setw(10)<<left<<b[i].author<<"\t"<<b[i].price<<"\t"<<b[i].count<<"\n";
    sort(b, N);
    cout<<"\nМАССИВ ОТСОРТИРОВАННЫЙ ПО ПОЛЮ count\n\n";
    //вывод массива на экран и запись в файл
    for(int i = 0; i < N; i++)
    {
        cout<<setw(25)<<left<<b[i].title<<setw(10)<<left<<b[i].author<<"\t"<<b[i].price<<"\t"<<b[i].count<<"\n";
        bookf<<b[i].title<<"\n"<<b[i].author<<"\n"<<b[i].price<<"\n"<<b[i].count<<"\n";
    }
    max_price(b, N);
    fin.close(); //закрыть файл
    bookf.close();
    getch();
    return 0;
}

```

```

#include "stdafx.h"
#include <fstream>
#include <iostream>
#include <iomanip>
#include <string.h>
#include <conio.h>
#include <windows.h>
using namespace std;
struct book
{
    char title[100];
    char author[100];
    float price;
    int count;
};
// сортировка массива структур по полю «имя автора»
void sort(book arr[], int size)
{
    for (int i=size-1; i > 0; --i)
        for (int j=0; j < i; ++j)
            if(strcmp(arr[j].author,arr[j+1].author)<0)
            {
                book tmp = arr [j];
                arr [j] = arr [j+1];
                arr [j+1] = tmp;
            }
}
// подсчет количества книг каждого автора и поиск автора с максимальным количеством книг
void kol_books(book arr[], int size)
{
    char auth[100], max_auth[100];
    int i=0, max_c=0;
    cout<<"\n"<<setw(25)<<left<<"АВТОР"<<setw(25)<<left<<"КОЛИЧЕСТВО КНИГ"<<"\n\n";
    while ( i < size) // проход по массиву
    {
        strcpy(auth,arr[i].author);
        int c=0;
        while (strcmp(arr[i].author,auth)==0)
        {
            c+=arr[i].count; // подсчет количества книг автора auth
            ++i;
        }
        cout<<setw(25)<<left<<auth<<setw(25)<<left<<c<<"\n";
        //поиск автора с максимальным количеством книг
        if (c>max_c)
        {
            max_c=c;
            strcpy(max_auth,auth);
        }
    }
    cout<<"\n"<<"АВТОР С МАКСИМАЛЬНЫМ КОЛИЧЕСТВОМ КНИГ ("<<max_c<<")
"<<max_auth<<"\n\n";
}

// поиск самой дорогой книги
void max_price(book arr[], int size)
{
    float max=arr[0].price;
    char max_title[100];
    strcpy(max_title,arr[0].title);
    char max_author[100];
    strcpy(max_author,arr[0].author);
    for (int j=0; j < size; ++j)
        if (max<arr[j].price)

```



```

        {
            max=arr[j].price;
            strcpy(max_title,arr[j].title);
            strcpy(max_author, arr[j].author);
        }
    cout<<"\nСАМАЯ ДОРОГАЯ КНИГА - "<<max_title<<" "<<max_author<<" ЕЕ ЦЕНА "<<max<<"\n";
}
int main()
{
    SetConsoleCP(1251);
    SetConsoleOutputCP(1251);
    ifstream fin("c:\\my_books.txt"); //исходный файл
    ofstream bookf;                  //файл для записи отсортированного массива
    bookf.open("c:\\BOOKINFO.txt");
    if(!fin.is_open())
    {
        cout<<"Файл не найден";
        return 1;
    }
    book b[10];    //массив из 10 структур book
    int N=0;       //первое значение в файле - размер массива структур
    fin>>N;
    for(int i = 0; i < N; i++)
    {
        fin.ignore(); //пропустить символ '\n' (значения в файле разделены переводом
        строки)
        fin.getline(b[i].title,100);    //прочитать строку с названием книги
        fin.getline(b[i].author,100);   //прочитать строку с именем автора
        fin>>b[i].price;                 // прочитайте строку с ценой книги
        fin>>b[i].count;                 // прочитайте строку с числом экземпляров
        книги
    }
    cout<<"ИСХОДНЫЙ МАССИВ\n\n";
    cout<<setw(25)<<left<<"title"<<setw(10)<<left<<"author"<<"\t"<<"price"<<"\t"<<"count"<<"\n\n";
    //вывод массива на экран
    for(int i = 0; i < N; i++)
    cout<<setw(25)<<left<<b[i].title<<setw(10)<<left<<b[i].author<<"\t"<<b[i].price<<"\t"<<b[i].count<<"\n";
    sort(b, N);
    cout<<"\nМАССИВ ОТСОРТИРОВАННЫЙ ПО ПОЛЮ count\n\n";
    //вывод массива на экран и запись в файл
    for(int i = 0; i < N; i++)
    {
        cout<<setw(25)<<left<<b[i].title<<setw(10)<<left<<b[i].author<<"\t"<<b[i].price<<"\t"<<b[i].count<<"\n";
        bookf<<setw(25)<<left<<b[i].title<<setw(10)<<left<<b[i].author<<"\t"<<b[i].price<<"\t"<<b[i].count<<"\n";
    }
    kol_books(b, N);
    max_price(b, N);
    fin.close(); //закрывать файл
    bookf.close();
    getch();
    return 0;
}

```