

Глава 9. Использование структур.

Если в программе используются разнотипные данные, имеющие некоторое общее назначение, то имеет смысл объединять их в структуры.

Рассмотрим для примера объект «файл». У него имеется имя, тип (расширение), размер. Кроме того, чтобы найти его на диске необходимо знать имя диска, и папки, в которой он находится. Эту информацию можно объединить в структуру следующим образом:

```
struct File
{
    char name[80]; // Имя
    char type[16]; // Тип
    char disk; // буква диска
    char dir[100]; // директория
    long size; // размер файла
};
```

Рассмотрим программу, выполняющую основные операции с данными структурного типа. Структуры в файл целесообразно писать в бинарном режиме, т.к. в этом случае не требуется подробное перечисление элементов структуры. Однако, следует учесть, что такой файл нельзя будет прочитать текстовым редактором. Кроме того, в составе структур не должно быть элементов с динамическим выделением памяти (иначе в файл будут записываться указатели на них, а не сами данные). Такая запись слегка избыточна, т.к. в файл будут записываться структуры целиком, включая неиспользованные участки текстовых массивов. Но с ней можно смириться, если записей не слишком много.

Рассмотрим пример программы, обрабатывающей данные о файлах.

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
struct File
{
    char name[80]; // Имя файла
    char type[16]; // Тип файла
    char disk; // буква диска
    char dir[100]; // директория
    long size; // размер файла
};
const int MAX_SIZE=100; // максимальное к-во обрабатываемых структур

void Input(File* pf) // функция ввода структуры, pf - адрес куда вводить
{
    printf("Имя: ");
    fgets(pf->name, 80, stdin); // ввод строки с именем файла
    pf->name[strlen(pf->name)-1]='\0'; // удаление из строки символа '\n'
    printf("Тип: ");
    fgets(pf->type, 16, stdin); // ввод строки с названием типа файла
    pf->type[strlen(pf->type)-1]='\0'; // удаление из строки символа '\n'
    printf("Диск: ");
    pf->disk=getchar(); // ввод символа - имени диска
    getchar(); // удаление из входного потока символа '\n'
    printf("Директория: ");
    fgets(pf->dir, 100, stdin); // ввод строки с директорией
    pf->dir[strlen(pf->dir)-1]='\0'; // удаление из строки символа '\n'
    printf("Размер: ");
    scanf("%ld",&pf->size); // ввод размера файла
    getchar(); // удаление из входного потока символа '\n'
}

void Print(File f) // функция вывода структуры, f - выводимая структура
{
    printf("\nИмя: %s\n", f.name);
    printf("Тип: %s\n", f.type);
    printf("Диск: %c\n", f.disk);
    printf("Директория: %s\n", f.dir);
    printf("Размер: %d байт\n", f.size);
}

void Write(File f[],int n) // функция записи в файл всех структур (n шт.)
{
```

```

FILE* store;
store=fopen("files.dat","wb"); // открытие на запись в бинарном режиме
if(store!=NULL)
{
    fwrite((void*)f,sizeof(File),n,store);
    fclose(store);
    printf("Записано %d структур\n",n);
}
else
    puts("Файл для записи не открылся.");
}
int Read(File* f) // функция чтения из файла всех структур
{
    FILE* store;
    store=fopen("files.dat","rb"); // открытие на чтение в бинарном режиме
    if(store!=NULL)
    {
        int n=0;
        while( n < MAX_SIZE &&
            fread((void*) (f+n),sizeof(File),1,store)==1 )
            // читаем структуры по одной, пока не достигнем конца файла
            n++;
        printf("Прочитано %d структур\n",n);
        return n;
    }
    else
    {
        puts("Файл для чтения не открылся.");
        return 0;
    }
}
void FullName(char* fname,File f) // функция формирующая полное имя файла
{
    sprintf(fname,"%c:\\%s\\%s.%s",f.disk,f.dir,f.name,f.type);
}
int main()
{
    system("chcp 1251");

    File fl[MAX_SIZE]; // объявление массива структур
    int n=0; // счетчик количества заполненных структур
    int action; // переменная выбора пункта меню
    do
    {
        puts("\nВыберите действие\n"
            "1 - Ввод с клавиатуры\n"
            "2 - Вывод на экран\n"
            "3 - Запись\n"
            "4 - Чтение\n"
            "5 - Полные имена\n"
            "0 - Выход из программы\n"
            );
        scanf("%d",&action);
        getchar(); // удаление из потока символа "Enter"
        switch(action)
        {
            case 1: // Ввод с клавиатуры одной структуры
                Input(&fl[n]); // указываем адрес, куда вводить
                n++;
                break;
            case 2: // Вывод на экран всех структур
                for(int i=0;i<n;i++)
                    Print(fl[i]); // указываем, что выводить
                break;
            case 3: // запись всех структур на диск

```

```

        Write(f1,n);
        break;
    case 4: // чтение всех структур с диска
        n=Read(f1); // n - прочитанное к-во
        break;
    case 5: // обработка
        for(int i=0;i<n;i++)
        {
            char FileFullName[100];
            FullName(FileFullName,f1[i]);
            // вывод полного имени и размера
            printf("%s %ld\n",FileFullName,f1[i].size);
        }
        break;
    }
}while(action!=0); // завершаем цикл при вводе нуля
return 0;
}

```

Функция *Input()* вводит данные очередной структуры с клавиатуры, функция *Print()* выводит содержимое структуры на экран, функция *write()* выводит массив структур в бинарный файл, функция *Read()* читает массив структур из бинарного файла и возвращает количество прочитанных структур, функция обработки *FullName()* из полей структуры формирует полное имя, включающее букву диска, имя папки и имя файла.

В основной программе имеется цикл, позволяющий выбрать необходимое действие с массивом структур.

Варианты задач для решения

В задании необходимо написать программу, которая содержит функции ввода и вывода информации структурного типа с клавиатуры и из файла в бинарном режиме (аналогично приведённому образцу) и функции, выполняющие обработку данных структуры в соответствии с решаемым вариантом.

В функции *main()* необходимо реализовать консольное меню, аналогичное приведённому выше образцу, позволяющее считывать данные из файла в массив структур, записывать массив в файл, добавлять данные с клавиатуры, выводить содержимое всего массива, выполнять указанные действия с этим массива.

1. Вариант

Задан список структур Books: имя и фамилия писателя, название произведения, год издания.
 Написать программу, которая
 -по заданному автору выводит все его произведения
 -выводит список всех авторов, произведения которых начинаются с заданной буквы.

2. Вариант

Задан список пользователей User: фамилия, имя, код пользователя, дата регистрации.
 Написать программу, которая выводит
 -всех пользователей фамилии которых начинаются с заданной буквы
 -всех тезок (если они есть).

3. Вариант

Автомагазин продает автомобили. О каждой машине хранится информация о марке автомобиля, фирме изготовителе, годе выпуска, объеме двигателя, типе кузова, стоимости.
 Написать программу, которая выводит список
 - всех автомобилей указанной фирмы и имеющих заданный тип кузова
 - автомобилей стоимостью ниже заданной выпущенных после указанного года.

4. Вариант

Задан список структур Mountains для представления информации по горным вершинам: название территории, название вершины и ее высота.
 Написать программу, которая выводит
 -список всех вершин расположенных на заданной территории
 -названия всех вершин имеющих высоту выше среднего значения.

5. Вариант

Задан список структур Hotels описывающий гостиничный номер: название гостиницы, номер, комфортность (люкс, полулюкс, стандарт, эконом), количество человек, стоимость.

Написать программу, которая выводит информацию

- о номерах, в названии гостиницы которых есть по 2 буквы 'a'

- о номерах, в которых стоимость люкса не более заданной.

6. Вариант

Задан список структур Student, содержащую фамилию и инициалы, номер группы и успеваемость (массив из пяти элементов).

Написать программу, которая выводит

- фамилии и номера групп студентов, имеющих хотя бы одну неудовлетворительную оценку;

- фамилии и успеваемость студентов из заданной группы.

7. Вариант

Задан список структур Sklad (вид товара; наименование товара; количество; стоимость; процент торговой надбавки).

Написать программу, которая выводит

- суммарное количество товаров каждого вида

- список товаров заданного вида стоимостью не выше заданной.

8. Вариант

Задан список структур Saity: URL, E-mail, ответственный, дата внесения, количество обращений.

Написать программу, которая выводит

- список сайтов с датой внесения не позже указанной и количеством обращений меньших заданного

- количество сайтов зарегистрированных за каждым из ответственных.

9. Вариант

Задан список структур Journals: код, название, периодичность, цена, дата начала издания. Написать программу, которая выводит

- суммарное количество журналов с периодичностью 1 месяц

- список журналов стоимостью не выше заданной и датой начала издания не ранее указанной.

10. Вариант

Задан список структур Books: ISSN, название, автор, к-во страниц, год издания, издательство. Написать программу, которая выводит

- количество книг у каждого из авторов.

- список книг с количеством страниц не менее заданного и годом издания не ранее указанного.

11. Вариант

Задан список структур Films: код, название, режиссер, дата выхода, кассовые сборы. Вывести

- суммарное количество фильмов каждого из режиссеров

- список фильмов с кассовыми сборами не менее заданного вышедших за указанный год.

12. Вариант

Задан список структур Songs: группа, солист, название, дата, рейтинг.

Написать программу, которая выводит

- количество песен у каждого из солистов

- список песен за указанный год с рейтингом не ниже заданного.

13. Вариант

Задан список структур Oborudovanie: Инв. номер, наименование, дата приобретения, стоимость, аудитория.

Написать программу, которая выводит

- суммарное количество оборудования по каждому из наименований

- список оборудования стоимостью не выше заданной в указанной аудитории.

14. Вариант

Задан список структур Materialy: артикул, наименование, дата приобретения, дата списания, стоимость.

Написать программу, которая выводит

- суммарное количество материалов каждого наименования

- список материалов со стоимостью не выше заданной и списанных за указанный год.

| |
|---|
| <p>15. Вариант Задан список структур Товар: наименование, артикул, штрих код, производитель, единица измерения, цена. Написать программу, которая выводит - суммарное количество товаров каждого производителя - список товаров данного производителя стоимостью не выше заданной.</p> |
| <p>16. Вариант Задан список структур Pокupateli: код покупателя, фамилия, адрес, телефон, сумма покупки. Написать программу, которая выводит - общую сумму покупок по каждому покупателю - список покупателей делающих покупки стоимостью не ниже заданной.</p> |
| <p>17. Вариант Задан список структур Postavshiki: код поставщика, наименование, адрес, телефон, вид товара. Написать программу, которая выводит - суммарное количество товаров каждого поставщика - список поставщиков заданного вида товара.</p> |
| <p>18. Вариант Задан список структур Zаkazy: код заказа, дата заказа, код заказчика, артикул товара, количество товара. Написать программу, которая выводит - суммарное количество заказанных предметов по каждому из артикулов - список заказов за указанный месяц по указанному артикулу.</p> |
| <p>19. Вариант Задан список структур Prodazhi: дата покупки, код-заказа, наименование товара, количество, вид оплаты(нал/карта). Написать программу, которая выводит - суммарное количество покупок каждого из товаров за указанный год - список покупок с количеством не ниже заданного оплаченных наличными.</p> |
| <p>20. Вариант Задан список структур Postavki: дата поставки, артикул, количество, цена-поставки. Написать программу, которая выводит - суммарное количество поставок по каждому из артикулов - список поставок с ценой не ниже заданной за указанный год.</p> |
| <p>21. Вариант Задан список структур Fishing: вид рыбы, дата ловли, вес, длина, занятое место. Написать программу, которая выводит - суммарное количество выловленной рыбы по каждому из месяцев - список выловленной рыбы весом не ниже заданного для заданного вида рыбы.</p> |
| <p>22. Вариант Задан список структур Turnir: дата, код участника, фамилия, к-во выигрышей, проигрышей, ничьих, место. Написать программу, которая выводит - суммарное количество выигрышей - список участников занимавших первые места в заданном году.</p> |
| <p>23. Вариант Задан список структур Futbol: код команды, название, к-во забитых мячей, пропущенных мячей, место. Написать программу, которая выводит - суммарное количество забитых мячей - список команд занимавших призовые места.</p> |
| <p>24. Вариант Задан список структур Electrichestvo: код абонента, адрес, дата, показание пред, показание тек, уплачено. Написать программу, которая выводит - суммарное количество потраченной энергии по каждому из абонентов - список должников на указанную дату.</p> |

25. Вариант

Задан список структур Kwartplata: лицевой счет, фамилия, адрес, телефон, дата оплаты, сумма. Написать программу, которая выводит

- суммарную суммы оплаты по каждому абоненту
- список абонентов заплативших в указанном году сумму не менее заданной.

26. Вариант

Задан список структур Account: номер счета, пароль, фамилия, сумма, процент, срок хранения. Написать программу, которая выводит

- общую сумму на счетах у каждого из клиентов
- список счетов имеющих срок хранения не менее заданного под указанный процент.

27. Вариант

Задан список структур User: Фамилия, логин, пароль, дата, суммарный трафик. Написать программу, которая выводит

- общий трафик каждого пользователя за указанный месяц.
- список пользователей, у которых трафик за указанную дату был не менее заданного.

28. Вариант

Задан список структур Kafedra: код, название, зав-кафедрой, аудитория, телефон.
Написать программу, которая выводит

- общее количество аудиторий закрепленных за каждой кафедрой
- список кафедр пользующихся указанным телефоном.

29. Вариант

Задан список структур Univer: факультет, декан, телефон, количество кафедр, количество учебных групп.
Написать программу, которая выводит

- суммарное количество учебных групп
- список факультетов имеющих количество кафедр не более заданного и количество групп не менее заданного.

30. Вариант

Задан список структур Person: фамилия, имя, отчество, дата рождения, семейное положение, адрес проживания.
Написать программу, которая выводит

- суммарное количество записей по каждому из семейных положений.
- список тех, кто родился не ранее указанной даты и имеет указанное имя.