

Лабораторная работа №12.

Динамические массивы структур

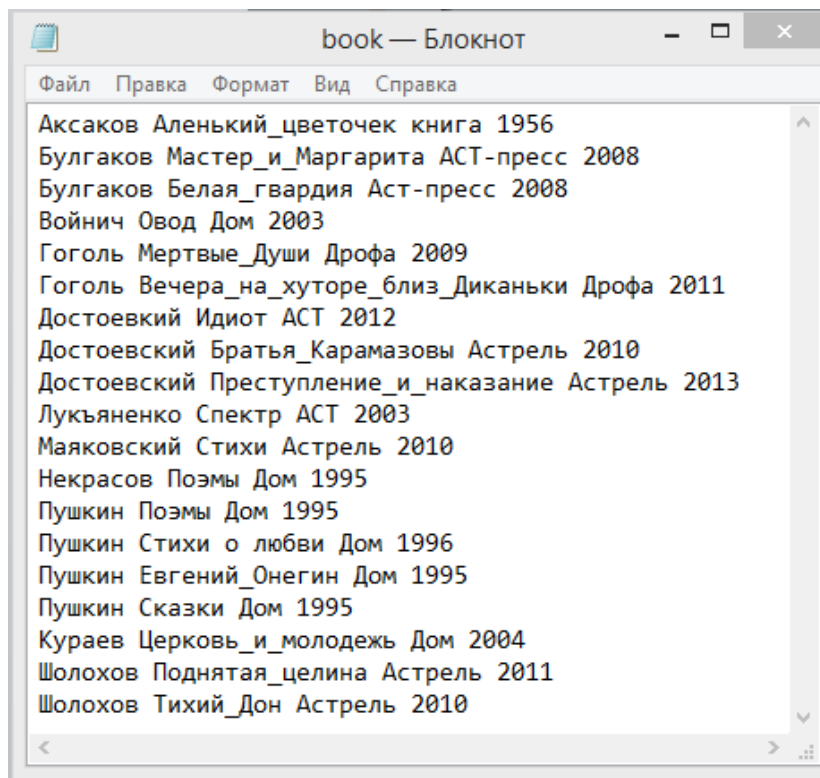
(2 часа)

Цель работы:

Знакомство и получение навыков работы с динамическими массивами структур на языке C++.

2. Условия задания:

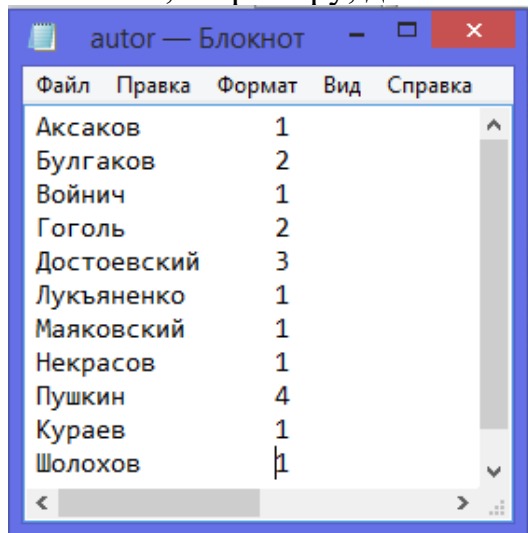
В файле находятся данные со списком книг



Строка 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.

Файл, к примеру, должен выглядеть так:



Поскольку мы не знаем заранее, сколько записей в файле, то не можем описать *статический массив* структур.

Размер структуры будет определяться в программе чтения данных из файла. Для работы с массивами структур, размеры которых заранее неизвестны используются *указатели при передаче структур в функции* и *указатели на функции*, где формируется размер массива структуры.

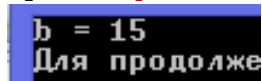
Хотя функция - это не переменная, она по-прежнему имеет физическое положение в памяти, которое может быть присвоено указателю. Адрес, присвоенный указателю, является входной точкой в функцию. Указатель может использоваться вместо имени функции. Он также позволяет передавать функции как обычные аргументы в другие функции.

Чтобы понять, как работают указатели на функции, необходимо понять, как компилируются и вызываются функции в С. По мере компиляции функций исходный код преобразуется в объектный и устанавливается точка входа. При вызове функции происходит обращение к данной точке входа. Поэтому указатель на функцию может использоваться для вызова функции.

Адрес функции получается при использовании имени функции без каких-либо скобок или аргументов. (Очень похоже на массивы, где адрес получается с использованием имени массива без индексов.)

Примеры применения указателей с простыми переменными

```
int a,b;  
int *aa; // указатель на переменные целого типа  
a=15;  
aa=&a; // указатель получает адрес переменной a  
b=*aa; // значение для b берется по адресу переменной a  
cout << "b = "<<b <<endl;  
system("pause");
```



Применение указателей на структуры и функции рассмотрите на примере текста предложенной программы.

3. Решение задания (словесный алгоритм):

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.

Особенностью работы с файлами является введение имени файла через **консоль**.

4. Текст программы

```
#include <conio.h>//для _getch()
#include <iostream>
#include <string>
#include <stdlib.h>
#include <fstream>
#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 *perrech(book x[], int k, int &k2); //прототип функции типа autor - формирование списка авторов с количеством их книг
void outputperrech(autor y[], int k2); //прототип функции - вывод полученного списка на консоль
void outputperrechfile(autor y[], int k2); //прототип функции - вывод полученного списка в файл

// 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(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(autor 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(autor 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;
}

void 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");
}

```

Варианты 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 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.