

Структуры в С++

Для работы с группой различных по типу данных, в языке программирования С++ применяют тип данных структура

Структуры в C++

Понятие структуры мы рассмотрим на примере ведомости по результатам сессии:

N	Фамилия	Количество экзаменов	Средний балл
1.	Сурин	5	3,2
2.	Куличёв	3	4,5
3.	Яшин	4	4,0

Каждая строка в этой таблице состоит из отдельных элементов - данных разного типа:

- а) порядковый номер - целый тип;*
- б) Фамилия - строковый тип;*
- в) количество экзаменов- целый тип;*
- г) средняя оценка – вещественный тип;*

Зачем нужны структуры?

Студент

- порядковый номер
- фамилия
- количество экзаменов
- средний балл

целое число

символьная строка

целое число

дробное число

Как хранить данные?

```
int id[n]  
string fam[n];  
int kol_ekz[n];  
double sr_ball[n];
```

неудобно работать
(сортировать и т.д.), ошибки

Эти разнотипные данные можно объединить в одну группу и считать структурой

Структура – это составной тип данных, содержащий набор элементов разных типов. Составляющие структуру элементы называются ее полями. В структуре каждое поле имеет собственное имя. Чтобы описать структуру, необходимо указать имя структуры и имена объектов, составляющих структуру и их типы.

Для объявления структуры используется ключевое слово **struct**.

Тип структуры вводится описанием следующего вида:

```
struct [имя_struct] {тип 1 имя_поля 1;  
                    тип 2 имя_поля 2;
```

.....

```
тип n имя_поля n;};
```

где **имя_struct** - имя структуры, тип 1, тип 2, ..., тип n - любые predetermined типы ;
имя_поля 1, ... , имя_поля n - идентификаторы полей, удовлетворяющие правилам задания идентификаторов.

Объявление и определение структур

Пример описания структуры:

```
struct Student{ int id;  
                string  fam;  
                int  kol_ekz;  
                double sr_ball;  
                };
```

Мы определили структуру с именем **Student**. Она содержит 4 переменные: **id** типа `int`, **fam** типа `string`, **kol_ekz** типа `int` и **sr_ball** типа `double`. Эти переменные, которые являются частью структуры, называются **полями структуры**.

Чтобы использовать структуру **Student**, нам нужно просто объявить переменную типа `Student`, например: **Student person;**

// имя структуры (Student) принято начинать с заглавной буквы, а имя переменной (person) - со строчной, но это необязательно.

Обращение к полям структур

Когда мы объявляем переменную структуры, например, **Student person**, то **person** ссылается на всю структуру. Для того, чтобы получить доступ к отдельным её полям (элементам), используется **оператор выбора поля** (.).

В примере используется оператор выбора полей для инициализации каждого поля структуры:

```
person.id = 1;      // присваиваем значение полю id структуры person  
person.fam = "Surin"; // присваиваем значение полю fam структуры person  
person.kol_ekz = 4; // присваиваем значение полю kol_ekz структуры person  
person.sr_ball = 4.17; // присваиваем значение полю sr_ball структуры person
```

Задача

В классе n учеников. Введите фамилии и оценки по 5 предметам для каждого ученика, найдите средний балл и выведите на экран фамилию и средний балл лучшего ученика.

```
#include <iostream>
#include <windows.h>
using namespace std;
int main()
{
    SetConsoleCP(1251); // установка кодовой страницы win-ср 1251 в поток ввода
    SetConsoleOutputCP(1251); // установка кодовой страницы win-ср 1251 в поток вывода
    struct Student
    {
        string fam;
        int p1, p2, p3, p4, p5;
        float sr;
    };
    int n, i;
```

Задача

В классе n учеников. Введите фамилии и оценки по 5 предметам для каждого ученика, найдите средний балл и выведите на экран фамилию и средний балл лучшего ученика.

продолжение программы:

```
cin>>n;
Student a[n];
float m_b=0;
for(i = 0;i < n; i++)
{
    cout << "Введите фамилию" << endl;
    cin >> a[i].fam;
    cout << "Введите 5 оценок" << endl;
    cin >> a[i].p1 >> a[i].p2 >> a[i].p3 >> a[i].p4 >> a[i].p5;
    a[i].sr = (a[i].p1+ a[i].p2 + a[i].p3 + a[i].p4 + a[i].p5) / 5.;
}
for(i = 0; i < n; i++)
    if(a[i].sr >= m_b) m_b = a[i].sr;
for(i = 0; i < n; i++)
    if (a[i].sr == m_b)
        cout << a[i].fam << " средний балл= « << a[i].sr << endl;
return 0;
}
```

Запись структуры в файл

```
struct base
{
    string name;
    string work;
    int year;
};
base a;

ofstream f;
f.open("out.txt");

f<<a.name<<" "<<a.work<<" "<<a.year<<endl;

f.close();
```

Задача: записать данные типа структура в файл

продолжение программы

```
#include <fstream>
#include <string>
#include <windows.h>
using namespace std;
struct base
{
    string name;
    string work;
    int year;
};
int main()
{SetConsoleCP(1251);
 SetConsoleOutputCP(1251);
 int n; //кол-во экземпляров структуры//
 cin>>n;
 base b[n];
```

```
    for (int i=0;i<n;++i)
    {
        cout << "Enter name : " << endl;
        cin >> b[i].name;
        cout << "Enter work : " << endl;
        cin >> b[i].work;
        cout << "Enter year : " << endl;
        cin >> b[i].year;
    }
    ofstream outfile;
    outfile.open("out.txt");
    for (int i = 0; i < n; ++i)
        outfile << b[i].name << " " << b[i].work << " " << b[i].year << endl;
    outfile.close();
    return 0;
}
```

Чтение структуры из файла

```
struct base
{
    string name;
    string work;
    int year;
};
base a;

ifstream f;
f.open("out.txt");

f>>a.name>>a.work>>a.year;

f.close();
```

Задачи для самостоятельной работы

1. В файл записать данные типа структура о результатах сдачи экзамена. Структура содержит фамилию, имя и количество баллов:
<Фамилия> <Имя> <Количество баллов>
Вывести в другой файл фамилии и имена тех учеников, которые получили больше 80 баллов.
2. В предыдущей задаче добавить к полученному списку нумерацию, сократить имя до одной буквы и поставить перед фамилией:
 1. **П. Иванов**
 2. **И. Петров**
 3. **...**

Задачи для самостоятельной работы

3. В файле записаны данные о результатах сдачи экзамена. Каждая строка содержит фамилию, имя и количество баллов, разделенные пробелами:

<Фамилия> <Имя> <Количество баллов>

Вывести в другой файл данные учеников, которые получили больше 80 баллов. Список должен быть отсортирован по убыванию балла. Формат выходных данных:

П. Иванов 98

И. Петров 96

...