

Структуры. Массивы структур.

C/C++



Для чего нужен тип структура?

- При решении задач обработки большого количества **однотипных** данных используют массивы
- Чаще встречаются комбинации из данных разных типов: например, сведения о сотрудниках (фамилия, имя, год рождения, стаж работы)
- Для обработки **разнотипных** данных в языке C\C++ имеется комбинированный тип данных — **структура**



Что такое структура?

Структура — это тип данных, состоящий из фиксированного числа компонент одного или различных типов, называемых полями структуры.



В языке C\C++ структура:

- Стандартный тип **struct**
- Включает ряд компонент, называемых **ПОЛЯМИ**
- В разных полях могут храниться данные разного типа
- Структура имеет **ИМЯ**; поля структуры также имеют имена



Объявление структуры в программе:

```
struct Date
{
    unsigned int Year;
    char Month[12];
    unsigned int Day;
    char DayWeek[10];
}
```

//поля структуры имеют разные
ТИПЫ

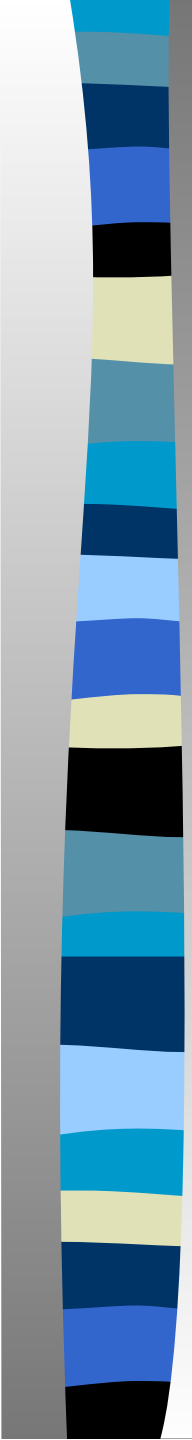


Переменные типа структура

После введения типа структура можно задать переменные или массивы этого типа

```
Date D1 = new Date();
```

```
Date *ArDate = new Date[10];
```



Обращение к полям структуры:

- ❑ Осуществляется указанием имени переменной и имени поля через точку
- ❑ Для указателей используется операция выбора ->

```
D1.Year=2015;
```

```
D1.Month="September\0";
```

```
D1.Day=29;
```

```
D1.DayWeek="Monday\0"
```

Ввод и вывод структуры:

Составные имена с указанием полей
можно использовать в операторах
ввода-вывода:

```
cin>>D1.Year>>D1.Month>>D1.Day;  
cout<<D1.Year<<" "<<D1.Month<<" "<<  
D1.Day<<endl;
```

Нельзя использовать в операторах
ввода-вывода переменную-структуру
целиком:

```
cout<<D1;           // ошибка !!!
```




Массивы структур:

Для хранения и обработки большого количества структур удобно использовать **массивы структур**.

В этом случае сначала объявляется **структура**, например, `Person`, а затем **массив**, состоящий из некоторого количества структур типа `Person`.



Пример использования массива структур

Введем структуру Person для описания сведений о сотруднике: фамилия, имя, пол, год рождения, стаж, должность.

Затем объявим массив, состоящий из 5 структур типа Person.

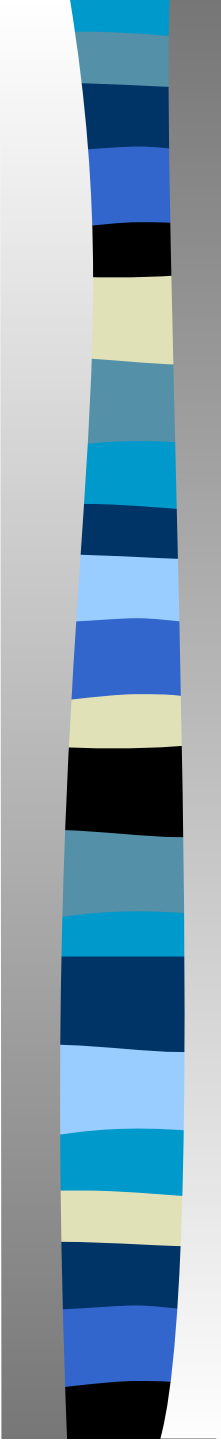
Из этого массива можно будет получить выборочные сведения.



Описание структуры и массива

структур:

```
#include <stdio.h>
void main()
{
    struct Person
    { char Name[20];
      char Profession[15];
      unsigned int Year;
      unsigned int Stag;
      char Pol;
    } A[5];
    int i, k;
```



```
for (i=0; i<5; i++)
{ printf("Введите фамилию:\n");
  gets_s(A[i].Name);
  _flushall();
  printf("Введите должность:\n");
  gets_s(A[i].Profession);
  _flushall();
  printf("Введите год рождения:\n");
  scanf("%u",&A[i].Year);
  printf("Введите стаж:\n");
  scanf("%u", &A[i].Stag);
  printf("Введите пол (м/ж):\n");
  A[i].Pol = getch();
  printf("%c",A[i].Pol);
  printf("\n-----\n"); }
```

***Ввод данных
в массив***

Вывод массива на экран:

```
printf("\n фамилия\t должность\t год\t  
стаж\t пол\n");  
    // вывод строки-заголовка  
printf("\n-----\n");  
for (i=0; i<5; i++)  
{   printf("%s\t %s\t %u\t %u\t %c\n",  
    A[i].Name, A[i].Profession,    A[i].Year,  
A[i].Stag, A[i].Pol);  
}
```



Выбор сотрудников со стажем менее 5 лет

```
for (i=0; i<5; i++)  
{  
    if (A[i].Stag<5)  
        printf("%s\t %s\t %u\t %u\t %c\n",  
            A[i].Name, A[i].Profession, A[i].Year,  
            A[i].Stag, A[i].Pol);  
}
```

Поиск мужчин и женщин, достигших пенсионного возраста

```
for (i=0; i<5; i++)  
{  
    if  (A[i].Pol=='ж' && 2019-A[i].Year>=55 ||  
        A[i].Pol=='м' && 2019-A[i].Year>=60)  
  
        printf("%s\t %s\t %u\t %u\t %c\n",  
            A[i].Name, A[i].Profession,  A[i].Year,  
            A[i].Stag, A[i].Pol);  
}
```

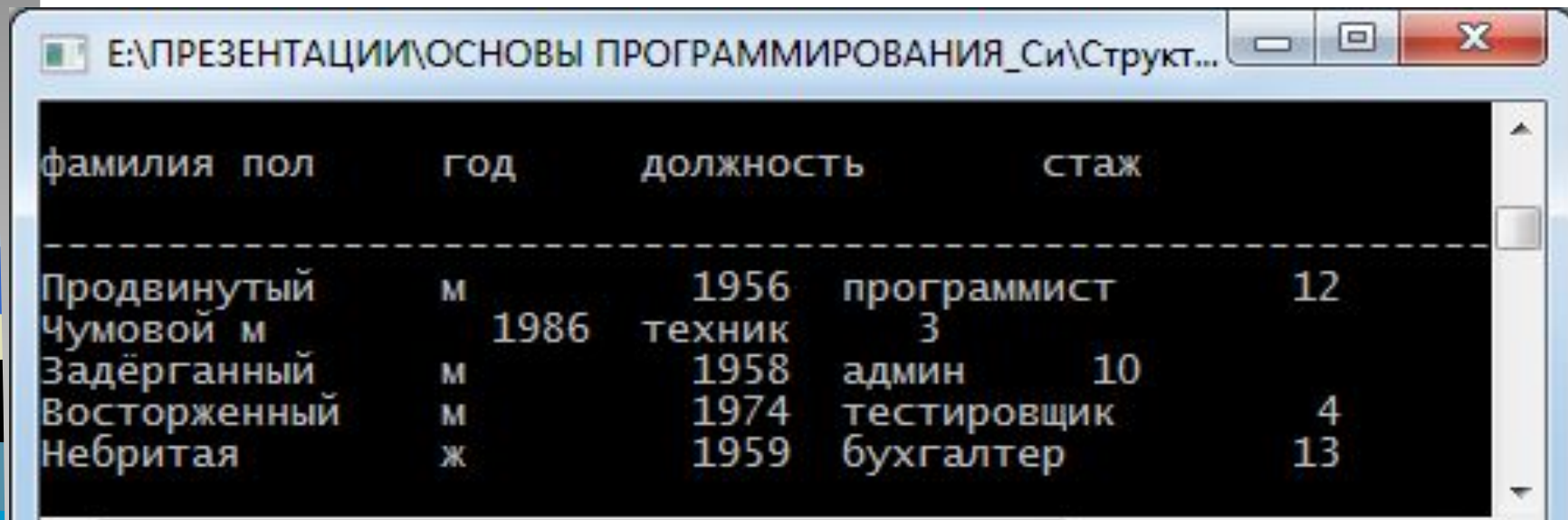
Вычисление среднего возраста мужчин

```
int S = 0; k = 0;
float Sr;
for (i=0; i<5; i++)
{
    if (A[i].Pol == 'М')
    {
        S += 2019-A[i].Year;
        k++;
    }
}
Sr = (float) S/k;
printf("Средний возраст мужчин %0.1f", Sr);
```


Ввод данных в массив:

```
E:\ПРЕЗЕНТАЦИИ\ОСНОВЫ ПРОГРАММИРОВАНИЯ...
Введите фамилию:
Продвинутый
Введите должность:
программист
Введите год рождения:
1956
Введите стаж:
12
Введите пол (м/ж):
м
-----
Введите фамилию:
Чумовой
Введите должность:
техник
Введите год рождения:
1986
Введите стаж:
3
Введите пол (м/ж):
м
-----
Введите фамилию:
Задёрганный
Введите должность:
админ
Введите год рождения:
1958
Введите стаж:
10
Введите пол (м/ж):
м
-----
```

Вывод массива на экран



The screenshot shows a Windows command prompt window with the title bar text "E:\ПРЕЗЕНТАЦИИ\ОСНОВЫ ПРОГРАММИРОВАНИЯ_Си\Структ...". The window contains a table of data printed in a monospaced font. The table has five columns: "фамилия", "пол", "год", "должность", and "стаж". The data is as follows:

фамилия	пол	год	должность	стаж
Продвинутый	м	1956	программист	12
Чумовой	м	1986	техник	3
Задёрганный	м	1958	админ	10
Восторженный	м	1974	тестировщик	4
Небритая	ж	1959	бухгалтер	13

Выборка данных

Сотрудники со стажем менее 5 лет:

фамилия	пол	год	должность	стаж
Чумовой	м	1986	техник	3
Восторженный	м	1974	тестировщик	4

Сотрудники пенсионного возраста

фамилия	пол	год	должность	стаж
Продвинутый	м	1956	программист	12
Небритая	ж	1959	бухгалтер	13



Задача 2

Опишите структуру сведений о товарах:
наименование, производитель, цена,
количество на складе.

Объявите массив, состоящий из N
структур.

Из этого массива получить выборочные
сведения:

- О товарах, которых нет на складе
- О товарах с максимальной ценой



Начать программу можно так:

```
#include <stdio.h>
void main()
{
    struct Товар
    {   char Name[80];
        char Brend[40];
        float Prace;
        unsigned int Kol;
    }
    int i, k, N;
    printf("Введите количество товаров: ");
    scanf("%i", &N);
    Товар *T=new Товар[N];
```



Задача 3

Составить программу обработки данных о машинах (марка машины, год выпуска, вид машины: грузовая или легковая, для легковых машин указать количество посадочных мест, для грузовых – грузоподъемность).

Найти в массиве грузовые и легковые машины определенной марки.



Пример объявления классов с наследованием

```
class Car
{   char Brend[20];
    unsigned int Year;  }
    // фиксированная часть (базовый класс)

class Sedan : Car
    {   unsigned int Passagir;  }
class Truck : Car
    {   float Tonn;  }
    // вариантная часть (дочерние классы)
```

Пример работы с классами

```
с:\ ?????????  
Марка: ауди  
Год выпуска: 1990  
Вид: легковая  
Количество мест: 4  
-----  
Марка: мерседес  
Год выпуска: 2010  
Вид: легковая  
Количество мест: 6  
-----  
Марка: мерседес  
Год выпуска: 2008  
Вид: грузовая  
Грузоподъемность: 12  
-----  
Марка: порше  
Год выпуска: 1998  
Вид: легковая  
Количество мест: 4  
-----  
Марка: мерседес  
Год выпуска: 2009  
Вид: грузовая  
Грузоподъемность: 9.5_
```


Результат запроса

C:\ ?????????				
Введите нужную марку: мерседес				
Марка	Год	Вид	Количество	
мерседес	2010	легковая	Посадочных мест:	6
мерседес	2008	грузовая	Грузоподъёмность:	12.0
мерседес	2009	грузовая	Грузоподъёмность:	9.5
Всего машин этой марки: 3				



Выводы:

- Структура является значимым, а не ссылочным типом данных, то есть экземпляр структуры хранит значения своих элементов, а не ссылки на них
- Структура не может участвовать в иерархиях наследования
- В программе разрешено напрямую обращаться к полям экземпляра структуры, присваивать им значения

Домашнее задание

Выучить теорию по теме «Структуры».

Задача

Составить программу для работы с массивом записей о студентах (фамилия, год рождения, группа, средний балл).

Вывести сведения о студентах определенной группы, средний балл которых выше 4.
