

Структуры

Определение 1 Структура – совокупность компонент, называемых полями, которые могут относиться к различным типам данных.

Общий вид:

```
struct <имя> {  
    <тип1> <поле1>;  
    <тип2> <поле2>;  
    .....  
};
```

Примеры: struct coord{

```
    float x,y;
```

```
};
```

```
struct city{
```

```
    char name[20];
```

```
    int kol;
```

```
};
```

Свойства структур

- 1. Описание переменных структурного типа
- `coord a; city b[10];`
- 2. Доступ к полям структуры:
- `a.x a.y b[1].name b[2].name[0]`
- 3. Присваивание структурных переменных
- Переменные одного структурного типа можно присваивать друг другу
- `coord a,b; a=b;`
- `city p,q; p=q;`
- 4. Остальные операции выполняются для каждого поля в отдельности в соответствии с типом этого поля
- Пример задачи 1
- Задан массив точек на плоскости. Определить координаты точки, расположенной наиболее близко к началу координат
- `#include "stdafx.h"`
- `#include "iostream"`
- `#include "string.h"`

Пример программы

- `#include "math.h"`
- `using namespace std;`
- `struct coord{`
- `float x,y;};`
- `coord minrast(coord *s,int n);`
- `void main()`
- `{`
- `coord mas[10];`
- `cout <<"Input mas"<<endl;`
- `for(int i=0;i<10;i++)`
- `cin>>mas[i].x>>mas[i].y;`
- `coord r = minrast(mas,10);`
- `cout<<r.x<<" " <<r.y;`
- `system("pause");`
- `}`

Пример программы

- `coord minrast(coord *s,int n)`
- `{`
- `coord rm;`
- `float minr=1.0e20;`
- `for(int i=0;i<n;i++)`
- `{`
- `float r = sqrt(s[i].x*s[i].x+s[i].y*s[i].y)`
- `if(r<minr)`
- `{`
- `minr=r;`
- `rm=s[i];`
- `}`
- `}`
- `return rm;`
- `}`

Пример программы

- Пример задачи 2
- Задан массив структурных переменных, каждый элемент которого состоит из фамилии студента и его возраста. Составить список студентов, фамилии которых начинаются на заданную букву, а возраст не превосходит заданного значения
- `#include "stdafx.h"`
- `#include "iostream"`
- `#include "string.h"`
- `struct stud{`
- `char name[20];`
- `int age;};`
- `int spis(stud *mas,int n,char c,int ag,char rm[][20]);`
- `void main()`
- `{`
- `stud mas[20];`

Пример программы

- `char rm[20][20];`
- `cout <<“Input mas”<<endl;`
- `for(int i=0;i<10;i++)`
- `cin>>mas[i].name>>mas[i].age;`
- `char c; int ag;`
- `cin>>c>>ag;`
- `int kol = spis(mas,20,c,ag,rm);`
- `for(int i=0;i<kol;i++)`
- `cout<<rm[i]<<endl;`
- `system(“pause”);`
- `}`
- `int spis(stud *mas,int n,char c,int ag,char rm[][20])`
- `{`
- `int k=0;`

Пример программы

- `for(int i=0;i<n;i++)`
- `if((mas[i].age<=ag)&&(mas[i].name[0]==c)`
- `{`
- `strcpy(rm[k],mas[i].name);`
- `k++;`
- `}`
- `return k;`
- `}`

Структуры с битовыми полями

Если у структуры одно или несколько полей имеют ограниченный спектр значений, то можно существенно сэкономить отводимую под структуру память

Пример:

```
struct person {  
    char name[20];  
    unsigned age; //4 байта
```

Структуры с битовыми полями

- `unsigned sex; //4 байта`
- `unsigned child; //4 байта`
- `};`
- Под последние 3 поля – 12 байт (48 бит)

```
struct person {
```

```
    char name[20];
```

```
    unsigned age:7; // 7 бит
```

- `unsigned sex:1;`

- `unsigned child:4;`

- `};`

- Итого под последние 3 поля – 12 бит

- Пример использования – заголовок каждого фрейма mp3 - файла

Структуры с битовыми полями

- struct mp3_frame_hdr {
- unsigned short sync;
- unsigned char priv:1;
- unsigned char pad:1;
- unsigned char freg:2;
- unsigned char rate:4;
- unsigned char mode:2;
- unsigned char mode_exit:2;
- unsigned char copy:1;
- unsigned char orig:1;
- unsigned char emph:2;
- };

Объединение

- **Определение** Объединение – это структура, в которой поля перекрывают друг друга, располагаясь по одному и тому же адресу

- **Общий вид:**

- union <имя> {
- <тип 1> <поле 1>;
- <тип 2> <поле 2>;
-
- };

- Пример:

- struct bytes {
- char lobyte;
- char hibyte;
- };

- union word

- {
- unsigned short w;
- bytes bt;
- };

-

Объединения

- `word d;`
- `d.w= 12;`
- `d.bytes.lobyte` – младший байт
- `d.bytes.hibyte` – старший байт

• Структуры и указатели

- Пример
- `struct coord{`
- `float x,y;};`
- `coord *p;`
- Доступ к каждому полю структуры осуществляется оператором доступа `->`, т.е.
- `p->x` - разименование указателя для поля X
- `p->y` - разименование указателя для поля Y
- Другая форма записи `(*p).x`

Структуры и указатели

- Пример: Дано 6 точек на плоскости, заданных своими координатами. Определить минимальное расстояние между точками
- `#include "stdafx.h"`
- `#include "iostream"`
- `#include "math.h"`
- `using namespace std;`
- `struct coord {`
- `float x,y};`
- `void main()`
- `{`
- `coord *p = new coord[6];`
- `for(int i=0;i<6;i++)`
- `cin>>(p+i)->x>>(p+i)->y;`
- `float min = 1.0e20;`

Структуры и указатели

- `for(int i=0;i<5;i++)`
- `for(int j=i+1;j<6;j++)`
- `{`
- `float r=sqrt(((p+i)->x - (p+j)->x)*((p+i)->x - (p+j)->x)+ ((p+i)->y - (p+j)->y)*((p+i)->y - (p+j)->y));`
- `if(r<min)`
- `min=r;`
- `}`
- `cout << "min = " << min;`
- `delete [] p;`
- `system("pause");`
- `}`