

Структура, инициализация.

Массив структур и его инициализация.

Динамический массив структур.

Задание структуры

Структура – это комбинированный тип данных.

Один элемент такого типа состоит из группы полей разного типа и произвольного количества.

Формат описания структуры

```
struct имя_типа {  
    список_полей_с_описанием_типа_и_имени};
```

Например,

```
struct abit {char fio[25], town[15]; int age};
```

Использование и инициализация:

```
struct abit {char fio[25], town[15]; int age;};
```

```
void main()
```

```
{ abit a = {"Андреев ИИ", "Москва", 17};  
  printf("%s %s %d", a.fio, a.town, a.age);  
}
```

Видим, что к полям структуры обращаемся через **точку**(имя_переменной.имя_поля).

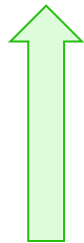
Массив определяется как обычно в статической или динамической области.

Каждый элемент массива инициализируется отдельно в { } или вводится по **ПОЛЯМ**.

Пример 1.

Задан список фамилий учеников
нескольких классов школы и их рост.
Определить в каком классе учится самый
высокий ученик.

```
struct schoolb{ char fio[20], class[4];  
                int height ;};
```



Новый тип данных

Пример 2

Массив структур можно задать и в *динамической* области.

Задача. Заданы 2 списка сдачи двух предметов абитуриентами ФПМК. Не все абитуриенты сдавали оба экзамена. Сформировать упорядоченный по убыванию суммарного балла список абитуриентов, сдавших оба экзамена.

```
struct abit { char fio[25]; int ball;};  
void main()  
{ abit *l1, *l2, *l3, t; int i, n, m, k = 0, j;  
  scanf("%d%d", &n, &m);  
  l1 = new abit[n];  
  l2 = new abit[m];  
  l3 = new abit[n<m?n:m];  
  print("Вводите 1 список");  
  for(i = 0; i<n; i++)  
  { scanf("%s", l1[i].fio);  
    scanf("%d",&l1[i].ball);  
    fflush(stdin);  
  }  
  //аналогично вводим 2-ой список
```

Условное выражение



```
for(i = 0; i<n; i++) // ищем
{ for(j = 0; j<m&&strcmp(l1[i].fio, l2[j].fio); j++);
  if (j<m)
  { strcpy(l3[k].fio, l1[i].fio); // заносим
    l3[k++].ball = l1[i].ball + l2[j].ball;}
}
```

```
for(i = 0; i<k-1; i++) // сортируем
```

```
  for(j = 0; j<k-i-1; j++)
```

```
    if ( l3[j].ball<l3[j+1].ball )
```

Целиком над структурами определена только операция присвоения '=' !

```
      { t = l3[j]; l3[j] = l3[j+1]; l3[j+1] = t;}
    
```

```
printf("\n Сдали оба экзамена абитуриенты:");
```

```
for(i = 0; i<k; i++)
```

```
  printf("\n %20s %3d",l3[i].fio, l3[i].ball);
```

```
delete [] l1; delete [] l2; delete [] l3;
```

```
}
```

Пример 3

Поле структуры может быть и другая известная структура.

Задача. Задан список студентов:

фамилия, группа, дата рождения.

Вывести список студентов, день рождения которых в мае.

```
struct date{ int day; char month[10]; int year; };
```

```
struct student { char fio[20]; int gr; date bd;};
```



```
struct date{ int day; char month[10]; int year; };
```

```
struct student { char fio[20]; int gr; date bd;};
```

```
void main()
```

```
{ student fpmk[20] = {  
    { "Андреев А", 1155, {10, "март", 1997} },  
    { "Алексеева В", 1165, {14, "май", 1998} },  
    { "Мазура С", 1172, {25, "июль", 1997} }  
};
```

```
int i, n=3;
```

```
printf("\n Родились в мае:");
```

```
for(i = 0; i<n; i++)
```

```
if( strcmp(fpmk[i]. bd.month, "май") == 0)
```

```
printf("\n%s %d", fpmk[i].fio, fpmk[i].bd.day);
```

```
}
```

Пример 4

Поля структуры – указатели.

Задача. Ввести список абитуриентов:

фамилия город, откуда приехал.

Использовать динамический массив.

Вывести фамилии томичей.

```
struct abit { char *fio, *town;};
```

```
struct abit { char *fio, *town;};
```

```
void main()  
{ abit *tgu; char fam[20], town[30];  
  int n = 15, i;  
  tgu = new abit [n];  
  puts("Вводите фамилию, enter, город, enter");  
  for(i = 0; i<n; i++)  
  { scanf("%s", fam); scanf("%s", town);  
    tgu[i].fio = new char[strlen(fam)+1];  
    strcpy(tgu[i].fio, fam);  
    tgu[i].town = new char[strlen(town)+1];  
    strcpy(tgu[i].town, town);  
  }  
}
```

Томичи

```
for (i = 0; i<n; i++)
    if(strcmp(tgu[i].town, "Tomsk")==0)
        puts(tgu[i].fio);
// освободим память!
for( i = 0; i<n; i++)
{ delete [ ] tgu[i].fio; delete [ ] tgu[i].town;}
delete [ ] tgu;
}
```

Задачи.

Дополнение к задаче 2.

2. Заданы 2 списка сдачи двух предметов абитуриентами ФПМК. Не все абитуриенты сдавали оба экзамена.

Сформировать упорядоченный по убыванию суммарного балла список абитуриентов, сдавших оба экзамена.

Найти проходной балл, если на факультет набирается p человек. Возможно и полупроходной.

3. Задан список результатов сдачи сессии студентами:

фио, группа, оц1, оц2, оц3.

**Сформировать список студентов,
назначенных**

**на стипендию по результатам сессии в виде
ФИО группа стипендия.**

Правила назначения на стипендию:

все 5 – 10000, без троек – 7000.

**Использовать в обоих списках вложенную
структуру**

{ФИО, группа}