

Лабораторно упражнение N 12

Сложни типове данни от динамичен тип. Структури в език C.

I. Теоретична обосновка

Типовете данни в програмните езици могат да бъдат класифицирани като:

- **прости** - типове данни, които се състоят от един елемент (компонент);
- **структурни** - типове данни, които се състоят от повече от един елемент.

Типичен пример за данни от структуриран (сложен) тип са масивите. В език C като структуриран тип данни могат да бъдат класифицирани: структурите (`struct`) и обединенията (`union`). Тези типове данни могат да се използват за реализация на списъци и дървовидни структури.

1. Структури (*struct*)

Структурата е сложен тип данни, състоящ се от повече от един разклонени елемента. Структурата има еквивалентен тип *Record (запис)* в програмен език Pascal.

1.1. Деклариране на структура

Чрез декларирането на структура се създава нов тип данни, който е дефиниран от програмиста. Декларацията на структурата има следният общ вид:

```
struct име  
{ тип име1; //описание на първи елемент  
тип име2; //описание на втори елемент  
...  
};
```

Структурата представлява поредица от елементи, които в общия случай са от различен тип. *Име* е валиден идентификатор, служещ за дефиниране на структурата. Името се предшества от ключова дума *struct*. В блока се описват елементите на структурата, определени чрез техните *имена (идентификатори)* и техният *тип*.

Пример:

```
struct dot // структура dot , описваща точка  
{ int x; // с координатите: x,y и цвят:color  
int y ;  
unsigned char color;  
};
```

Чрез декларацията на структура не се дефинира променлива, а се определя нов тип т.е. явява се описание на шаблон за този тип. След като бъде декларирана дадена структура, в последствие, тя може да се използва като всеки един от типовете за дефиниране на променливи от този тип.

1.2. Операции със структури

Структурите могат да участват като операнди в следните оператори:

- прилагане на адресна операция **&**;
- обръщение към елемент от структура (операции **.** и **->**);
- инициализиране на структури.

Адресен оператор извличане на адрес (**&**) може да се приложи към променлива – структура, така както при обикновените променливи.

Пример:

```
struct dot
{ int x;
  int y;
} current, *pcurrent;
pcurrent = &current; //променлива pcurrent сочи ( съдържа адреса на )
// структура current
```

Обръщането към елемент от структура може да се извърши чрез една от следните оператори :

- оператор **.** (*извличане на елемент от структура, чрез променлива структура*) се прилага върху променлива-структура в следния вид:

променлива_структура.елемент

Пример :

```
current.x = 10;
current.y = 20;
```

- оператор **->** (*извличане на елемент от структура чрез указател*) се прилага към указател от тип структура в следния вид:

указател към структура -> елемент

Пример:

```
pcurrent=&current;
pcurrent->x=10; //задава стойности 10,10 за координатите x и y на текущата
pcurrent->y=10; // точка
```

Променлива структура се инициализира като всяка променлива - след знак за присвояване във фигурни скоби се записват поредица от начални стойности за всеки елемент от структурата.

Пример :

```
struct dot current = {100, 28};
```

1.3. Съвместно използване на структури и масиви

Масивите и структурите могат да се използват съвместно. Възможно е да се дефинира масив от структури или масив да е елемент на структура. В случай, че се дефинира масив от структури, дефиницията има следния общ вид:

```
struct име_структура име масив [граница];
```

II. Задачи за изпълнение

1. Да се декларира структура, описваща точка в равнината. Да се създаде конзолно приложение, с което се дефинират две точки, въвеждат се стойности за координатите и се извеждат. Да се допълни функционалността на приложението като се въведе масив от точки, въведат се и се извеждат стойностите на координатите.

2. Да се декларира структура, описваща правоъгълник. Да се дефинира функция, която намира лице на правоъгълник. Да се създаде приложение, с което се дефинира масив от правоъгълници, въвеждат се стойности за страните и се намират лицата на фигурите.

3. Да се декларира структура, описваща комплексно число. Да се дефинират функции, които намират сбора и разликата на две комплексни числа. Да се създаде демонстрационна програма, която симулира работата на калкулатор за комплексни числа.

Относно въпроси по темата на адрес: ln_zh_st@yahoo.com