

ЛАБОРАТОРНАЯ РАБОТА № 28

ПРОГРАММИРОВАНИЕ С ИСПОЛЬЗОВАНИЕМ СТРУКТУР

Цель работы: изучить особенности работы с пользовательским типом данных – структура.

Краткие теоретические сведения

Структура – это структурированный тип данных, представляющий собой поименованную совокупность разнотипных элементов. Тип *структура* обычно используется при разработке информационных систем, баз данных.

Формат описания структурного типа следующий:

```
struct имя_типа  
    {определения_элементов};
```

В конце обязательно ставится точка с запятой (это оператор). Элементы структуры называются *полями*. Каждому полю должно быть поставлено в соответствие имя и тип.

Например, сведения о выплате студентам стипендии требуется организовать в виде:

```
    |  
struct student {char fam[30];  
                int  kurs;  
                char grup[5];  
                float stip};
```

После этого `student` становится именем структурного типа, который может быть назначен некоторым переменным. В соответствии со стандартом СИ это нужно делать так:

```
struct student stud1, stud2;
```

Здесь `stud1` и `stud2` – переменные структурного типа.

Допускается и другой вариант описания структурных переменных, когда можно вообще не задавать имя типа, а описывать сразу переменные:

```
struct {char fam[30];
```

```

int kurs;
char grup[5];
float stip;
} stud1, stud2, *pst;

```

В этом примере кроме двух переменных структурного типа объявлен указатель pst на такую структуру.

Обращение к полям структурной величины производится с помощью *уточненного* имени следующего формата:

имя_структуры.имя_элемента

Примеры уточненных имен для описанных выше переменных:

```
stud1.fam; stud1.stip;
```

Значения элементов структуры могут определяться вводом, присваиванием, инициализацией. Пример инициализации в описании:

```
student stud1={"Кротов", 3, "ПО313", 350};
```

Пусть в программе определен указатель на структуру

```
student *pst, stud1;
```

Тогда после выполнения оператора присваивания

```
pst=&stud1;
```

к каждому элементу структурной переменной stud1 можно обращаться тремя способами. Например, для поля fam

stud1.fam или (*pst).fam или pst->fam

В последнем варианте используется *знак операции доступа к элементу структуры*: ->. Аналогично можно обращаться и к другим элементам этой переменной.

Допускается использование массивов структур. Например, сведения о 100 студентах могут храниться в массиве, описанном следующим образом:

```
student stud[100];
```

Тогда сведения об отдельных студентах будут обозначаться, например, так: stud[1].fam, stud[5].kurs и т.п. Если нужно взять первую букву фамилии 25-го студента, то следует писать: stud[25].fam[0].

Пример 1. Ввести сведения об N студентах. Определить фамилии студентов, получающих самую высокую стипендию.

```
# include <stdio.h>
```

```

#include <conio.h>
void main()
{ const N=30; int i; long maxs;
  struct student {char fam[15];
                  int kurs;
                  char grup[3];
                  long stip;  };

  student stud[N];
  clrscr();
  for (i=0; i<N; i++)
  { printf("%d-й студент",i);
    printf("\n"Фамилия:"); scanf("%s",&stud[i].fam);
    printf("Курс:"); scanf("%d",&stud[i].kurs);
    printf("Группа:"); scanf("%s",&stud[i].grup);
    printf("Стипендия:"); scanf("%ld",&stud[i].stip); }
  maxs=0;
  for (i=0; i<N; i++)
  if (stud[i].stip>maxs) maxs=stud[i].stip;
  printf("\n Студенты, получающие макс. стипендию %ld  руб.",
  maxs);
  for (i=0; i<N; i++)
  if (stud[i].stip==maxs) printf("\n%s", stud[i].fam);}

```

Порядок выполнения работы

1. Изучить теоретические сведения.
2. Ответить на контрольные вопросы.
3. Выполнить задание.

Контрольные вопросы

1. Какую переменную называют структурной или структурой?
2. В чем состоит преимущество представления информации в виде структуры?
3. Что называют элементом структуры?
4. Как инициализируются структуры?
5. Как выделяется память под структурную переменную?

6. Как осуществляется доступ к отдельным элементам структуры?

7. Какие операции допустимы над структурами?

Задания для выполнения

1. Сформировать массив из **n** структур, содержащих сведения об автомобилях: марка, цвет, номер, фамилия и адрес владельца. Вывести информацию об автомобилях заданной марки, номера которых начинаются с **k** заданных цифр. Предусмотреть случай, что таких сведений нет.

2. Сформировать массив из **n** структур, содержащих сведения о днях рождения друзей: фамилия и имя друга, название месяца, число, год рождения. Вывести информацию о друзьях, родившихся в заданном месяце с указанием исполняющегося количества лет. Предусмотреть случай, что таких сведений нет.

3. Ввести массив из **N** структур, содержащих сведения о лекарствах, хранящихся на аптечном складе. Структура записи: наименование лекарственного препарата, количество, цена, срок хранения (в месяцах). Вычислить, сколько стоят все препараты, хранящиеся на складе, и вывести сведения о препаратах, срок хранения которых более **X** месяцев. Значение **X** ввести с клавиатуры. Предусмотреть случай, что таких препаратов нет.

4. Ввести массив из **n** структур, содержащих сведения о сотрудниках института: фамилия, название отдела, год рождения, стаж работы, должность, оклад. Вычислить средний заработок работающих в отделе **X** (вводить с клавиатуры). Предусмотреть случай, что сведений об этом отделе нет.

5. Сформировать массив из **n** структур, содержащих сведения о книгах по программированию: фамилия автора, название книги, название издательства, год издания. Вывести данные о книгах, изданных после года **X**, в названии которых есть слово "Паскаль". Предусмотреть случай, что таких книг нет.

6. Даны сведения о телефонах абонентов. Каждая запись имеет поля: фамилия абонента, год установки телефона, номер телефона. Определить количество установленных телефонов, начиная с заданного года; по вводимой фамилии абонента выдать номер телефона.

Для заданий 7 – 15 написать программу по обработке массива структур, содержащего следующую информацию о студентах:

- фамилия и инициалы;
- год рождения;
- номер группы;
- оценки за семестр: физика, математика, информатика, химия;
- средний балл.

Организовать ввод исходных данных, средний балл рассчитывать по введенным оценкам.

7. Распечатать анкетные данные студентов, сдавших сессию на 8, 9 и 10.

8. Распечатать анкетные данные студентов-отличников, фамилии которых начинаются с интересующей вас буквы.

9. Распечатать анкетные данные студентов, фамилии которых начинаются с буквы *A* и сдавших математику на 9 и 10.

10. Распечатать анкетные данные студентов интересующей вас группы, имеющих оценку 9 по физике и оценку 10 по высшей математике.

11. Распечатать анкетные данные студентов, не имеющих оценок 4 и 5 по информатике и математике.

12. Вычислить общий средний балл всех студентов и распечатать список студентов со средним баллом выше общего среднего балла.

13. Вычислить общий средний балл всех студентов и распечатать список студентов интересующей вас группы, имеющих средний балл выше общего среднего балла.

14. Распечатать анкетные данные студентов интересующей вас группы, имеющих оценки ниже 4.

15. Распечатать анкетные данные студентов интересующей вас группы, имеющих средний балл выше введенного с клавиатуры.