ПЗ-02: **Работа со структурами**

**Теория** – см. лекцию.

***NB*:** Для использования структуры необходимо:

1. установить шаблон для структуры

2. объявить переменную, соответствующую этому шаблону

3. осуществить доступ к компонентам структуры.

**Шаблон структуры**

* ***Шаблон****– это схема, описывающая содержание структуры.*

Установка структурного шаблона телефонный справочник:

struct sprav {

char fio[20];

long num;

};

Данный шаблон описывает структуру с именем типа структуры sprav, состоящую из двух компонентов: строки fio и целой переменной num типа long.

Имя типа структуры sprav необязательно и используется для ссылки на эту структуру.

Компоненты структуры – данные любого типа, включая и другие структуры.

Имя внутри структуры может быть таким же, как имя объекта вне структуры.

Если шаблон описан внутри функции – он доступен только этой функции, если шаблон описан вне функции – он доступен любой функции программы.

Установка шаблона не вызывает никаких действий в программе.

**Структурные переменные**

Объявление структурных переменных приводит к выделению памяти для компонент структуры, куда можно записать данные или откуда можно прочитать их. Для объявления структурных переменных имеются несколько способов:

|  |  |
| --- | --- |
| 1) Установить структурный шаблон:  struct sprav {  char fio[20];  long num;  };  Объявить простую переменную, массив структур, указатель на структуру: struct sprav tel1, tel2[5], \*tel3; | 2) Объявить переменные одновременно с установкой шаблона (если на данную структуру вы больше не ссылаетесь):  struct {  char fio[20];  long num;  } tel1, tel2[3], \*tel3; |
| 3) Ввести новый тип данных (TEL)-структура определенного вида:  typedef struct {  char fio[20];  long num;  } TEL; | 3) Объявить переменные нового типа:  TEL tel1, tel2[6], \*tel3;  Если программа достаточно объемна, представляется более удобным третий способ. |

**Инициализация структуры.** Инициализировать можно только **внешние** или **статические** структуры:

static struct {

char fio[20];

long num;

} tel[2]={

"Иванов Ф.А.", 456756,

"Петров В.П.", 632345

};

============================================================================================

1. **Доступ к компонентам структуры**

|  |  |
| --- | --- |
| // **Обращение к элементам структуры через имя переменной**  #include <stdio.h>  #include <conio.h>  int main(void)  {  struct{  char fio[20]; // фамилия  long num; // телефон (long – 4 байта)  } tel1, tel2;  \*tel2.fio = "0";  tel2.num = 0;  puts("Enter your name: ");  gets(tel1.fio);  puts("Enter the tlf number: ");  scanf("%ld",&tel1.num);  tel2 = tel1; /\* так можно, но нельзя сравнивать структуры \*/  puts("Input :");  printf("Name: %s Number: %ld\n",tel2.fio,tel2.num);  return 0;  } // Почему во втором примере (справа) такой результат?.. |  |

/\* **Динамическое выделение памяти для структуры** \*/

/\* **Обращение к элементам структуры через указатель** \*/

#include <stdio.h>

#include <conio.h>

struct sprav

{

char fio[20];

long num;

};

int main()

{

struct sprav \*tel1, \*tel2;

/\* Выделение памяти для структуры \*/

tel1=(struct sprav \*)malloc(sizeof(struct sprav));

tel2=(struct sprav \*)malloc(sizeof(struct sprav));

puts("Enter your name: ");

gets(tel1->fio);

puts("Enter the tlf number: ");

scanf("%ld",&tel1->num);

\*tel2= \*tel1;

puts("Input: ");

printf("Name: %s Number: %ld\n",(\*tel2).fio,(\*tel2).num);

// printf("Name: %s Number: %ld\n",tel2->fio,tel2->num); // или так

free(tel1);

free(tel2);

return 0;

}

============================================================================================

1. **Массив структур**

|  |  |
| --- | --- |
| /\* Массив структур. Обращение к элементам структуры через \*/  /\* имя элемента массива \*/  #include <stdio.h>  #include <conio.h>  #include <string.h>  #define SPRAV struct sprav  int main()  {  int const N=3;  SPRAV  {  char fio[20];  long num;  };  SPRAV tel[N]; /\* массив структур - 3 элементов \*/  char fio\_tek[20];  int i;  /\* ввод данных в массив структур \*/  for(i=0; i<N; i++)  {  puts("Enter your name: ");  gets(tel[i].fio);  puts("Enter the tlf number: ");  scanf("%ld",&tel[i].num);  getchar();  }  puts("Choosing a phone by last name ");  gets(fio\_tek);  /\* поиск структуры по фамилии абонента \*/  for(i=0; i<N; i++)  if(!strcmp(fio\_tek,tel[i].fio)) break;  if(i!=5) /\* цикл закончен по break \*/  printf("Subscriber number %s is equal to %ld\n",fio\_tek, \  tel[i].num);  else /\* цикл выполнился полностью \*/  puts("Subscriber not found"); // Абонент не найден  return 0;  } |  |

/\* **Массив структур. Память выделяется динамически**. \*/

/\* **Обращение к элементам структуры через указатель** \*/

#include <stdio.h>

#include <conio.h>

#include <string.h>

int const N=3;

typedef struct

{

char fio[20];

long num;

} TEL;

int main()

{

TEL \*tel;

char fio\_tek[20];

int i;

/\* Выделение памяти для массива - N элементов \*/

tel=(TEL \*)malloc(sizeof(TEL)\*N);

for(i=0; i<N; i++)

{

puts("Enter your name: ");

gets((tel+i)->fio);

puts("Enter the tlf number: ");

scanf("%ld",&(tel+i)->num);

getchar();

}

puts("hoosing a phone by last name ");

gets(fio\_tek);

for(i=0; i<N; i++,tel++)

if(!strcmp(fio\_tek,tel->fio)) break;

if(i!=N)

printf("Subscriber number %s is equal to %ld\n",fio\_tek, \

tel->num);

else

puts("Subscriber not found"); // Абонент не найден

return 0;

}

============================================================================================

1. **Передача структуры в функцию**

Непосредственный доступ к компонентам структуры – плохой стиль программирования.

Все операции, которые разрешены применительно к структуре, должны быть при этом реализованы в виде отдельных функций.

***Не все компиляторы языка Си позволяют передавать структуры в функцию по значению***, поэтому в примерах передача структуры идет через указатель.

|  |  |
| --- | --- |
| /\* Передача структуры в функцию через указатель на структуру \*/  /\* Определение комплексного числа через структуру и действия \*/  /\* над комплексными числами ( ввод, вывод, вычисление суммы) \*/  #include <stdio.h>  typedef struct { float a; /\* действительная часть \*/  float b; /\* мнимая часть \*/  } COMPLEX;  void vvod(COMPLEX \*,float,float);  void sum(COMPLEX \*,COMPLEX \*,COMPLEX \*);  void out(COMPLEX \*);  void main(void)  {  COMPLEX x,y,z;  vvod(&x,2.5,6.7);  vvod(&y,6.89,8.45);  puts("Numbers are entered: ");  out(&x);  out(&y);  sum(&x,&y,&z);  puts("The sum of complex numbers is equal to: ");  out(&z);  }  /\* Вывод комплексного числа \*/  void out( COMPLEX \*p)  {  printf("(%.2f,%.2f)\n", (\*p).a,(\*p).b);  return;  } | /\* Вычисление суммы двух комплексных чисел \*/  void sum(COMPLEX \*p1,COMPLEX \*p2,COMPLEX \*p3)  {  (\*p3).a=(\*p1).a+(\*p2).a;  (\*p3).b=(\*p1).b+(\*p2).b;  return;  }  /\* Ввод значений для элементов структуры \*/  void vvod(COMPLEX \*p,float a, float b)  {  p->a=a;  p->b=b;  return;  } |

============================================================================================

1. **Вложенные структуры**

*Структура, являющаяся компонентом другой структуры, называется* ***вложенной****.*

|  |  |
| --- | --- |
| #include <stdio.h>  #include <conio.h>  typedef struct Model  {  int id;  struct  {  int id;  char \*name;  } make;  char \*name;  unsigned year;  } Model;  int main()  {  Model m = {10, {10, "Acura"}, "CL", 2012};  printf("Model name = %s\n", m.name);  printf("Make name = %s\n", m.make.name);  printf("year = %d\n", m.year);  getch();  return 0;  } | Структура сама может являться полем структуры.  **Пример**: структура Model – модель автомобиля, имеет:   * название * номер * год выпуска * поле Make (которое в свою очередь хранит):   + номер марки   + и её название.     Вложенные структуры инициализируются как многомерные массивы. |

==============================================================================================

**Задачи:**

**Описать структуру с именем STUDENT, содержащую следующие поля:**

* фамилия и инициалы;
* номер группы;
* успеваемость (массив из пяти элементов).

|  |  |
| --- | --- |
| 1. *Написать программу, выполняющую следующие действия*:  * ввод с клавиатуры данных в массив, состоящий из десяти структур типа STUDENT; * вывод на дисплей фамилий и номеров групп для всех студентов, включенных в массив, если средний балл студента больше 4,0; * если таких студентов нет, вывести соответствующее сообщение. | 1. *Написать программу, выполняющую следующие действия*:  * ввод с клавиатуры данных в массив, состоящий из десяти структур типа STUDENT; * вывод на дисплей фамилий и номеров групп для всех студентов, имеющих оценки 4 и 5; * если таких студентов нет, вывести соответствующее сообщение. |
| 1. *Написать программу, выполняющую следующие действия*:  * ввод с клавиатуры данных в массив, состоящий из десяти структур типа STUDENT; * вывод на дисплей фамилий и номеров групп для всех студентов, имеющих хотя бы одну оценку 2;   если таких студентов нет, вывести соответствующее сообщение. | |

**Описать структуру с именем AEROFLOT, содержащую следующие поля:**

* название пункта назначения рейса;
* номер рейса;
* тип самолета.

1. *Написать программу, выполняющую следующие действия:*

* ввод с клавиатуры данных в массив, состоящий из семи элементов типа AEROFLOT;
* вывод на экран номеров рейсов и типов самолетов, вылетающих в пункт назначения, название которого совпало с названием, введенным с клавиатуры;
* если таких рейсов нет, выдать на дисплей соответствующее сообщение.

1. *Написать программу, выполняющую следующие действия:*

* ввод с клавиатуры данных в массив, состоящий из семи элементов типа AEROFLOT;   
  записи должны быть размещены в алфавитном порядке по названиям пунктов назначения;
* вывод на экран пунктов назначения и номеров рейсов, обслуживаемых самолетом, тип которого введен с клавиатуры;
* если таких рейсов нет, выдать на дисплей соответствующее сообщение.

**Самостоятельно** изучить и реализовать (*будет доп. плюсом*) в одном из примеров чтение/запись структур в/из файл(а).