

12. Типы языка Си. Преобразование типов.

План ответа:

1. понятие «тип данных»;
2. классификация типов языка Си;
3. простые типы:
 - целочисленные,
 - вещественные,
 - символьные,
 - перечисляемый тип,
 - логический тип (с99);
4. оператор typedef;
5. операция sizeof;
6. неявное и явное преобразование типов.

???

???

1. понятие «тип данных»;

Тип данных определяет:

внутреннее представление данных в памяти;
множество значений, которые могут принимать величины этого типа;
операции и функции, которые можно применять к величинам этого типа.

а.

2. классификация типов языка Си;

Простые (скалярные) типы

целый (int);
вещественный (float и др.);
символьный (char);
перечисляемый тип;
логический тип (с99);
void;
указатели.

Составные (структурированные) типы

массивы;
структуры;
объединения.

3. простые типы:

3.1 целочисленные

В языке Си существует несколько типов целых чисел.

Они различаются объемом памяти, отводимым под переменную (диапазоном);

возможностью присваивания положительных и отрицательных чисел.

Для создания нужного целого типа используется ключевое слово int и модификаторы типа:

signed, unsigned, short, long

Только следующие шесть комбинаций «создают» разные типы

short int; unsigned short int; int; unsigned int; long int;

t; unsigned long int;

Стандарт C99 добавляет еще два целых типа: long long int; unsigned long long int;

d long long int;

Согласно стандарту тип int знаковый.

Стандарт определяет минимальный диапазон значений (limits.h) и относительные размеры типов

sizeof(short int) <= sizeof(int) <= sizeof(long int)

<= sizeof(long long int)

Для указания типа константы используются «суффиксы» L (l), U (u),

LL (ll).

2016ull // unsigned long long int

3.2 вещественные

Язык Си предоставляет три вещественных типа:

float; double; long double

Стандарт Си не оговаривает точность указанных типов.

Константы, определяющие характеристики вещественных типов, могут

быть найдены в заголовочном файле float.h.

```
Для указания типа константы используются «суффиксы» F (f), L (l).  
57.0l // long double
```

3.3 символьные

Для работы с символами предназначен тип char. (Под набором символов, как правило, понимается набор символов ASCII.)
Переменной типа char может быть присвоено значение любого ASCII символа.

Под значение типа char отводится один байт.
Язык Си интерпретирует символьный тип как «маленькое целое».
Стандарт не определяет «знаковость» этого типа.
Для ввода и вывода значений символьного типа используется спецификатор %c.

Стандартные функции для обработки отдельных символов объявляются в заголовочном файле ctype.h.

3.4 перечисляемый тип

Тип разработан для переменных, которые принимают небольшое количество значений.

Значения этого типа перечисляются программистом.
enum color { RED, GREEN, BLUE };

3.5 логический тип

Стандарт C99 добавил логический тип _Bool.
_Bool flag;
Переменные типа _Bool могут принимать только значения 0 и 1.
flag = 5; printf("%d", flag); // 1

Стандарт c99 предоставляет заголовочный файл stdbool.h, который облегчает использование «нового» логического типа.

```
#include <stdbool.h>  
bool flag; flag = true;
```

4. оператор typedef;

Позволяет определять имена новых типов.
typedef тип имя;

“+” улучшает читаемость.
“+” облегчает внесение изменений.

```
typedef unsigned int quantity_t;  
quantity_t n_pens, n_copybooks;
```

5. операция sizeof;

Операция sizeof возвращает размер переменной или типа в байтах . sizeof(выражение);

6. неявное и явное преобразование типов.

Неявные
int i = 3.541 + 3; // предупреждение компилятора (?)

Явные
int i = 0;
i = (int) 3.541 + 3;
printf("%d", i);

Неявные преобразования типа

6.1 Операнды в арифметическом или логическом выражении имеют разный тип (обычное арифметическое преобразование: все операнды приводятся к наибольшему типу из встретившихся).

6.2 Присваиваются друг другу выражения разных типов (результатирующим является тип выражения, которому присваивается значение).

6.3 Передача функции аргумента, тип которого отличается от типа соответствующего формального параметра (тип фактического аргумента приводится к типу параметра).

6.4 Возврат из функции значения, тип которого не совпадает с типом возвращаемого результата, заданным в объявлении (тип фактически возвращаемого значения приводится к объявленному).

Обычное арифметическое преобразование Общие правила:

Типы всегда приводятся к тому типу, который способен обеспечить наибольший диапазон значений при наибольшей точности.

Любое арифметическое выражение, включающее в себя целые типы, меньшие чем `int`, перед вычислением всегда преобразуется в `int`.

Дополнения

`char`, `signed char`, `unsigned char`, `short int` преобразуются в `int`.
`unsigned short int` преобразуется в `int`, если этот тип достаточен для представления всего диапазона значений `unsigned short int`. В противном случае преобразуется в `unsigned int`.

Преобразование `unsigned int` в `long` происходит только, если тип `long` способен вместить весь диапазон значений `unsigned int`. В противном случае преобразуется в `unsigned long`.