

7. ОБЪЯВЛЕНИЯ И ОПРЕДЕЛЕНИЯ

Любое имя, за исключением имён меток, должно быть объявлено в программе:

```
int i, j;  
double d=7.3; extern int m;  
typedef unsigned int size_t; int add  
(int a, int b){ return a + b; } void  
func (char*, int);
```

После этих объявлений компилятор знает, что *i*, *j*, *m* – имена переменных типа *int*, *d* – имя переменного типа *double*, *size_t* – имя типа, а *add* и *func* – имена функций (о функциях см. раздел 16).

В объявлении с именем может сопоставляться не только некоторый тип. С именем может быть сопоставлен некоторый элемент, идентификатором которого оно является. Например, при объявлении *int i, j*; для переменных *i*, *j* отводится память, с этими именами сопоставляются участки памяти по 2 байта, которые можно использовать для хранения значений переменных. С именем *size_t* сопоставлено имя конкретного типа (*unsigned int*), синонимом которого является теперь *size_t*, а с именем *add* сопоставлен код функции. Все такие объявления называют определениями. Однако не все объявления являются определениями. Из всех вышеприведённых объявлений два не являются определениями: *extern int m*; *void func (char *, int)*;

В первом из этих объявлений сообщается только, что переменная *m* имеет тип *int*, память же для неё должна выделяться где-то в другом месте, о чём говорит ключевое слово *extern*.

Второе объявление говорит, что *func* есть имя функции с двумя аргументами, первый из которых – указатель на *char*, а второй – *int*, а сама функция не возвращает никакого значения. Такое объявление называют **прототипом** функции. Сама же функция *func* должна быть определена вместе со своим телом где-то в другом месте.

В программе для каждого имени должно присутствовать одно и только одно определение, в то время как объявлений, не являющихся определениями, может быть как угодно много.