

### 13. СИНОНИМ ИМЕНИ ТИПА

Синоним имени типа строится при помощи ключевого слова **typedef**. Выражение, в котором присутствует это ключевое слово, является описанием некоторого имени. Наличие слова **typedef** говорит о том, что объявляемый идентификатор становится не именем объекта некоторого типа, а синонимом имени этого типа.

```
int INTEGER;           // INTEGER – имя переменной типа int
typedef int INT;        // INT – синоним типа int typedef
unsigned size_t; typedef char string [255]; typedef void (*FPTR)
(int);
```

Последние две строки определяют **string** как синоним типа «массив из 255 символов», а **FPTR** – синоним типа «указатель на функцию, имеющую один аргумент типа **int** и не возвращающую никакого результата».

После объявления с помощью **typedef** новое имя становится полноценным именем типа:

```
string array;           // array – массив из 255 символов
FPTR func_pointer; // void (*func_pointer)(int); typedef
string STRING;
```

Использование **typedef** может упростить понимание сложных имен типов. Так, тип **int (\*(void))[ ]** можно построить так:

```
typedef int intarray[]; // тип «массив целых» typedef intarray *
ptrtointarray; //тип указателя на массив целых typedef
ptrtointarray TYPE (void);
```

Теперь имя **TYPE** можно использовать, например, в прототипах:  
**double fun (int, int, TYPE);**