

# Идентификаторы в языке Си. Области видимости идентификаторов. Связывание. Примеры.

**Идентификатор** – последовательность строчных ("a – z" ) и прописных ("A – Z" ) букв латинского алфавита, а также цифр ("0– 9") и знака подчеркивания ("\_").

Идентификатор начинается либо с буквы, либо со знака нижнего подчеркивания.

**Идентификатор** отображает способ, которым программа, написанная на Си, указывает на объект, хранящийся в аппаратной памяти, а также регламентирует действия, которые могут быть выполнены с объектом.

Пример:

```
int vals[10]; const int valConst[10];
int *pt = &vals; const *int ptConst = &vals;
```

**Область видимости (ОВ)** - описывает участок или участки программ, где можно обращаться к идентификатору. Например:

1. В пределах блока
2. В пределах функции
3. В пределах прототипа функции
4. В пределах файла

В пределах блока:

```
{
    double val = 10;
    ...
    return val;
}
```

Или:

```
double blocky(double cval )
{
    double val =0;
    int i;
```

```

    for (i = 0; i < 10; i++){
        double q = cval *i;val *= q;
    }
    return val;
}

```

В пределах функции, применяется только к меткам операции goto:

```

void someFunc(int a, int b){
    if (a>b)
        goto end;
    printf ("%d \n", b);
end: printf ("%d \n", a);
}

```

В пределах прототипа функции, применяется к переменным, используемым в прототипах:

```

void someFunc(int a, int b);
void someArrFunc(int n, int m, arr[n][m])

```

В пределах файлов

1. глобальная (в других файлах помимо текущего):

```

int units =0;
int main (void ){
    /* code */
    return 0;
}

```

2. только текущий файл (Идентификатор объявляется с ключевым словом `static` вне функций и виден только в пределах текущего файла.)

```

static int units = 0;
int main (void ){
    /* code */
    return 0;
}

```

**Связывание** - описывает видимость идентификаторов программы.

В Си переменная имеет одно из следующих связываний:

1. внешнее
2. внутреннее
3. отсутствует

### Отсутствие связывания:

Переменные с ОВ в пределах блока, функции или прототипа функции. Они являются закрытыми для блока, функции или прототипа, в котором определены.

```
{// void someFunc (int closed);  
    int closed = 1;  
    ...  
}
```

### Внутреннее связывание:

Переменные с ОВ в пределах файла. Для внутреннего связывания применяется ключевое слово `static`, которое явно указывает свойство связывания и запрещает к использованию переменную в другом файле.

```
static char ch = '\n' ;  
static void printSome (void ){  
    `puts ("Some" );  
}
```

### Внешнее связывание:

Переменные видны в других файлах программы.

```
int units = 0;  
int main (void){  
    /* code */  
    return 0;  
}
```