

Содержание

перечисления	3
Определение переменной без имени типа	3

перечисления

Перечисления представляют типы данных, которые содержат набор констант, и каждой константе сопоставлено определенное числовое значение. Перечисление определяется с помощью ключевого слова `enum`:

```
enum название_перечисление { константа1, константа2, ... константаN };
```

Так же как и для структур, для перечислений можно задавать псевдоним при помощи `typedef`

Также можно явным образом присвоить константам числовые значения, если нам не устраивают значения по умолчанию. Можно указать значение для первой константы, тогда все последующие будут получать значение на единицу больше:

```
enum operation {  
    ADD = 5,  
    SUBTRACT, // 6  
    MULTIPLY // 7  
};
```

Можно присвоить всем уникальные значения:

```
enum operation {  
    ADD = 1,  
    SUBTRACT = 2,  
    MULTIPLY = 4  
};
```

Определение переменной без имени типа

Если перечисление не планируется применять нигде, кроме как в одном только месте, то мы можем сразу определить его переменную без объявления имени типа:

```
#include <stdio.h>  
  
struct BinaryOp {  
    int n1;  
    int n2;  
    enum { ADD, SUBTRACT, MULTIPLY } op;  
};  
  
int main (void) {  
    struct BinaryOp mulOp = {.n1 = 6, .n2 = 8, .op = MULTIPLY};  
    printf("Operands: %d and %d. Operation: %d\n", mulOp.n1, mulOp.n2,  
mulOp.op);  
    // Operands: 6 and 8. Operation: 2  
    return 0;  
}
```

}

From:
<https://wiki.radi0.cc/> - radi0wiki

Permanent link:
https://wiki.radi0.cc/c:c_ultimate_guide:enum

Last update: **2025/11/09 12:07**

