

16. ПЕРЕЧИСЛЕНИЯ

Перечисление – это тип данных, который удобно использовать в случае применения в программе переменных и констант, принимающих значения из сравнительно небольшого множества целых чисел, причём таких, что обычно обращение к ним разумнее осуществлять по имени. Может быть и так, что их значение само по себе не важно.

Примером такого множества констант могут служить названия цветов, названия дней недели или месяцев года, названия шахматных фигур или символические имена арифметических операций языка C++.

Задание типа **перечисление** начинается ключевым словом **enum**, после которого идёт **имя** типа (иногда называемое **тегом**), за которым в фигурных скобках следует **список членов перечисления** – перечислителей:

```
enum chess {king, queen, rook, bishop, knight, p}; enum  
month {Jan, Feb, Mar, Apr, May, Jun,  
        Jul, Aug, Sep, Oct, Nov, Dec};  
enum operator_CPP{plus = '+', minus = '-', mult = '*', div  
= '/', rem = '%'};
```

Члены перечислений являются константами типов unsigned char или int, в зависимости от их значений и режима компиляции. При использовании перечислителя в выражении его тип всегда преобразуется в int.

Если перечислителям явно не присвоено никакого значения, как в chess и month, то первый из них получает значение 0, второй – 1, и т. д. Вообще, любой перечислитель по умолчанию имеет значение, на 1 превышающее значение предыдущего, если умолчание не отменяется явной инициализацией.

Все члены перечисления operator_CPP получают явно указанные значения, равные ASCII – кодам символов '+', '-', '*', '/', '%'.

Значения перечислителей вычисляются на стадии компиляции, поэтому при задании их значений можно использовать значения всех ранее определенных констант. При этом допускается, чтобы несколько перечислителей имели одно и то же значение: const int TWO = 2;

```
enum {first, second = first, next = first + TWO, last  
= next * next + 1} dummy;
```

Отметим, что в последнем случае не вводится тег перечисления, а сразу описывается переменная dummy, которая в соответствии с

шаблоном перечисления может принимать одно из четырех значений: first, second, next, last.

Следовательно, возможны присваивания

```
dummy = first; dummy  
= last;
```

В общем случае переменной-перечислению можно присваивать значения, задаваемые перечислителями. Так, если сделать описания month m1, m2; operator_CPP op1, op2, op3; enum colour { red, green } c[10]; то переменным op1, op2, op3 можно присваивать значения plus, minus, ...; m1, m2 – Jan, Feb и т. д., а элементам массива c[0]...c[9] – значения red и green.

Перечислитель может появиться везде, где разрешено появление значения типа int. Обратное неверно без явного преобразования типа.

```
int i = dummy;                // i равно 5  
month current_month = Jan;  
current_month = 0;            // Ошибка!  
current_month = ( month )0;    // Теперь верно!  
Feb = 1;                      // Ошибка: Feb – константа!
```

Пример:

```
# include < iostream > const  
int NUMDAYS = 7;  
enum DAYS { mon, tue, wen, thur, fri, sat, sun } day1, day2, day3;  
DAYS day_before(DAYS), day_after(DAYS);  
void print_day(DAYS); void  
main(){ setlocale(0, "rus"); day1 =  
sun; day2 = day_after(day1); day3 =  
day_before(day1); cout << "Если  
сегодня "; print_day(day1); cout <<  
", то завтра будет ";  
print_day(day2);  
    cout << ", \на вчершний день - ";  
    print_day(day3); cout << ".\n";  
    system("pause");}  
DAYS day_after(DAYS day){ return  
((DAYS)((day + 1) % NUMDAYS));  
}  
DAYS day_before(DAYS day){ int prev =  
(day - 1) % NUMDAYS;
```

```

        return (DAYS)((prev < 0) ? (NUMDAYS - 1) : prev);
    } void print_day(DAYS day){
    int day_i = day; const char *
    days[] ={ "Понедельник",
              "Вторник" ,
              "Среда",
              "Четверг",
              "Пятница",
              "Суббота",
              "Воскресенье"
    };
    if(day_i < 0 || day_i > NUMDAYS)
        cout << " Ошибка!\n"; else
        cout << days[day_i];
    }

```

Результат выполнения:

Если сегодня воскресенье, то завтра будет
Понедельник, а вчерашний день –
Суббота.