**Перечисляемый тип**

*В*си выделен отдельный тип перечисление (enum), задающий набор всех возможных целочисленных значений переменной этого типа. Синтаксис перечисления

[?](https://learnc.info/c/enums.html)

|  |
| --- |
| **enum <имя> {**  **<имя поля 1>,**  **<имя поля 2>,**  **...**  **<имя поля N>**  **};  //здесь стоит ;!** |

Например

[?](https://learnc.info/c/enums.html)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16 | #include <conio.h>  #include <stdio.h>    enum Gender {      MALE,      FEMALE  };    void main() {      enum Gender a, b;      a = MALE;      b = FEMALE;      printf("a = %d\n", a);      printf("b = %d\n", b);      getch();  } |

В этой программе объявлено перечисление с именем Gender. Переменная типа enum Gender может принимать теперь только два значения – это MALE И FEMALE.

По умолчанию, первое поле структуры принимает численное значение 0, следующее 1, следующее 2 и т.д. Можно задать нулевое значение явно:

[?](https://learnc.info/c/enums.html)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25 | #include <conio.h>  #include <stdio.h>    enum Token {      SYMBOL,         //0      NUMBER,         //1      EXPRESSION = 0, //0      OPERATOR,       //1      UNDEFINED       //2  };    void main() {      enum Token a, b, c, d, e;      a = SYMBOL;      b = NUMBER;      c = EXPRESSION;      d = OPERATOR;      e = UNDEFINED;      printf("a = %d\n", a);      printf("b = %d\n", b);      printf("c = %d\n", c);      printf("d = %d\n", d);      printf("e = %d\n", e);      getch();  } |

Будут выведены значения 0 1 0 1 2. То есть, значение SYMBOL равно значению EXPRESSION, а NUMBER равно OPERATOR. Если мы изменим программу и напишем

[?](https://learnc.info/c/enums.html)

|  |  |
| --- | --- |
| 4  5  6  7  8  9  10 | enum Token {      SYMBOL,             //0      NUMBER,             //1      EXPRESSION = 10,    //10      OPERATOR,           //11      UNDEFINED           //12  }; |

То SYMBOL будет равно значению 0, NUMBER равно 1, EXPRESSION равно 10, OPERATOR равно 11, UNDEFINED равно 12.

Принято писать имена полей перечисления, как и константы, заглавными буквами. Так как поля перечисления целого типа, то они могут быть использованы в операторе switch.

Заметьте, что мы не можем присвоить переменной типа Token просто численное значение. Переменная является сущностью типа Token и принимает только значения полей перечисления. Тем не менее, переменной числу можно присвоить значение поля перечисления.

Обычно перечисления используются в качестве набора именованных констант. Часто поступают следующим образом - создают массив строк, ассоциированных с полями перечисления. Например

[?](https://learnc.info/c/enums.html)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23 | #include <conio.h>  #include <stdio.h>  #include <stdlib.h>    static char \*ErrorNames[] = {      "Index Out Of Bounds",      "Stack Overflow",      "Stack Underflow",      "Out of Memory"  };    enum Errors {      INDEX\_OUT\_OF\_BOUNDS = 1,      STACK\_OVERFLOW,      STACK\_UNDERFLOW,      OUT\_OF\_MEMORY  };    void main() {      //ошибка случилась      printf(ErrorNames[INDEX\_OUT\_OF\_BOUNDS-1]);      exit(INDEX\_OUT\_OF\_BOUNDS);  } |