

Список ключевых слов C++

asm	else	new	this
auto	enum	operator	throw
bool	explicit	private	true
break	export	protected	try
case	extern	public	typedef
catch	false	register	typeid
char	float	reinterpret_cast	typename
class	for	return	union
const	friend	short	unsigned
const_cast	goto	signed	using
continue	if	sizeof	virtual
default	inline	static	void
delete	int	static_cast	volatile
do	long	struct	wchar_t
double	mutable	switch	while
dynamic_cast	namespace	template	

Константы в языке C++

Константа	Формат	Примеры
Целая	Десятичный: последовательность десятичных цифр, начинающаяся не с нуля, если это не число нуль	8, 0, 199226
	Восьмеричный: нуль, за которым следуют восьмеричные цифры (0,1,2,3,4,5,6,7)	01, 020, 07155
	Шестнадцатеричный: 0x или 0X, за которым следуют шестнадцатеричные цифры (0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F)	0xA, 0x1B8, 0X00FF
Вещественная	Десятичный: [цифры].[цифры] ²	5.7, .001, 35.
	Экспоненциальный: [цифры][.][цифры][Ee][+ -][цифры] ³	0.2E6, .11e-3, 5E10
Символьная	Один или два символа, заключенных в апострофы	'A', 'ю', '*', 'db', '\0', '\n', '\012', '\x07\x07'
Строковая	Последовательность символов, заключенная в кавычки	"Здесь был Вася", "Значение r=\0xF5\n"

Диапазоны значений простых типов данных для IBM PC

Тип	Диапазон значений	Размер (байт)
bool	true и false	1
signed char	-128 ... 127	1
unsigned char	0 ... 255	1
signed short int	-32 768 ... 32 767	2
unsigned short int	0 ... 65 535	2
signed long int	-2 147 483 648 ... 2 147 483 647	4
unsigned long int	0 ... 4 294 967 295	4
float	3.4e-38 ... 3.4e+38	4
double	1.7e-308 ... 1.7e+308	8
long double	3.4e-4932 ... 3.4e+4932	10

Пример программы, использующей функции ввода/вывода в стиле C:

```
#include <stdio.h>
int main(){
    int i;
    printf("Введите целое число\n");
    scanf("%d", &i);
    printf("Вы ввели число %d, спасибо!", i);
    return 0;
```

Управляющие последовательности в языке C++

Изображение	Шестнадцатеричный код	Наименование
\a	7	Звуковой сигнал
\b	8	Возврат на шаг
\f	C	Перевод страницы (формата)
\n	A	Перевод строки
\r	D	Возврат каретки
\t	9	Горизонтальная табуляция
\v	8	Вертикальная табуляция
\\	5C	Обратная косая черта
\'	27	Апостроф
\"	22	Кавычка
\?	3F	Вопросительный знак
\oddd	-	Восьмеричный код символа
\oxddd	ddd	Шестнадцатеричный код символа

та же программа с использованием библиотеки классов C++:

```
#include <iostream.h>
int main(){
    int i;
    cout << "Введите целое число\n";
    cin >> i;
    cout << "Вы ввели число " << i << ", спасибо!";
    return 0;
```

Основные операции языка C++

Операция	Краткое описание
Унарные операции	
++	увеличение на 1
--	уменьшение на 1
sizeof	размер
~	поразрядное отрицание
!	логическое отрицание
-	арифметическое отрицание (унарный минус)
+	унарный плюс
&	взятие адреса
*	разадресация
new	выделение памяти
delete	освобождение памяти
(type)	преобразование типа
Бинарные и тернарная операции	
*	умножение
/	деление
%	остаток от деления
+	сложение
-	вычитание
<<	сдвиг влево
>>	сдвиг вправо
<	меньше
<=	меньше или равно
>	больше
>=	больше или равно
==	равно
!=	не равно
&	поразрядная конъюнкция (И)
^	поразрядное исключающее ИЛИ
	поразрядная дизъюнкция (ИЛИ)
&&	логическое И
	логическое ИЛИ

Операция	Краткое описание
? :	условная операция (тернарная)
=	присваивание
*=	умножение с присваиванием
/=	деление с присваиванием
%=	остаток от деления с присваиванием
+=	сложение с присваиванием
-=	вычитание с присваиванием
<<=	сдвиг влево с присваиванием
>>=	сдвиг вправо с присваиванием
&=	поразрядное И с присваиванием
=	поразрядное ИЛИ с присваиванием
^=	поразрядное исключающее ИЛИ с присваиванием
.	последовательное вычисление

Операции увеличения и уменьшения на 1 (++ и --). Эти операции, называемые также инкрементом и декрементом, имеют две формы записи — *префиксную*, когда операция записывается перед операндом, и *постфиксную*. В префиксной форме сначала изменяется операнд, а затем его значение становится результирующим значением выражения, а в постфиксной форме значением выражения является исходное значение операнда, после чего он изменяется.

```
#include <stdio.h>
int main(){
    int x = 3, y = 3;
    printf("Значение префиксного выражения: %d\n", ++x);
    printf("Значение постфиксного выражения: %d\n", y++);
    printf("Значение x после приращения: %d\n", x);
    printf("Значение y после приращения: %d\n", y);
    return 0;
}
```

Результат работы программы:

```
Значение префиксного выражения: 4
Значение постфиксного выражения: 3
Значение x после приращения: 4
Значение y после приращения: 4
```

Спецификаторы формата функции printf()

Код	Формат
%c	Символ
%d	Десятичное целое число со знаком
%i	Десятичное целое число со знаком
%e	Научный формат (строчная буква e)
%E	Научный формат (прописная буква E)
%f	Десятичное число с плавающей точкой
%g	В зависимости от того, какой формат короче, применяется либо %e, либо %f
%G	В зависимости от того, какой формат короче, применяется либо %E, либо %F
%o	Восьмеричное число без знака
%s	Строка символов
%u	Десятичное целое число без знака
%x	Шестнадцатеричное число без знака (строчные буквы)
%X	Шестнадцатеричное число без знака (прописные буквы)
%p	Указатель
%n	Указатель на целочисленную переменную. Спецификатор вызывает присвоение этой целочисленной переменной количества символов, выведенных перед ним
%%	Знак %

Операторы ветвления

Условный оператор if

```

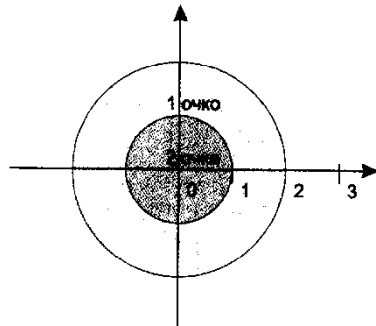
if (a<0) b = 1; // 1
if (a<b && (a>d || a==0)) b++; else {b * = a; a = 0;} // 2
if (a<b) {if (a<c) m = a; else m = c;} else {if (b<c) m = b; else m = c;} // 3
if (a++) b++; // 4
if (b>a) max = b; else max = a; // 5

```

```

#include <iostream.h>
int main(){
    float x, y, temp; int kol;
    cout << "Введите координаты выстрела\n";
    cin >> x >> y;
    temp = x * x + y * y;
    kol = 0;
    if (temp < 4 ) kol = 1;
    if (temp < 1 ) kol = 2;
    cout << "\n Очков: " << kol;
    return 0;
}

```

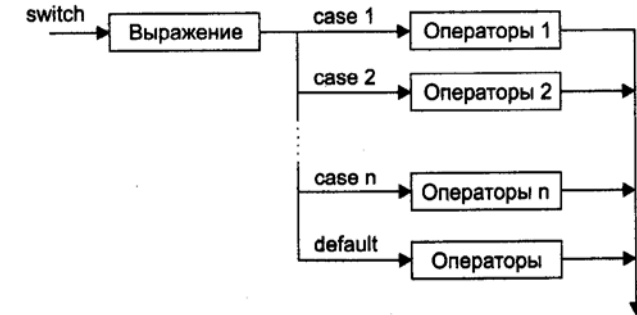


Оператор switch

```

switch ( выражение ){
    case константное_выражение_1: [список_операторов_1]
    case константное_выражение_2: [список_операторов_2]
    ...
    case константное_выражение_n: [список_операторов_n]
    [default: операторы ]
}

```



```

#include <iostream.h>
int main(){
    int a, b, res;
    char op;
    cout << "\nВведите 1й операнд : "; cin >> a;
    cout << "\nВведите знак операции : "; cin >> op;
    cout << "\nВведите 2й операнд : "; cin >> b;
    bool f = true;
    switch (op){
        case '+': res = a + b; break;
        case '-': res = a - b; break;
        case '*': res = a * b; break;
        case '/': res = a / b; break;
        default : cout << "\nНеизвестная операция"; f = false;
    }
    if (f) cout << "\nРезультат : " << res;
    return 0;
}

```

Операторы цикла

Цикл с предусловием (while)

Пример (программа печатает таблицу значений функции $y=x^2+1$ во введенном диапазоне):

```
#include <stdio.h>
int main(){
    float Xn, Xk, Dx;
    printf("Введите диапазон и шаг изменения аргумента: ");
    scanf("%f%f%f", &Xn, &Xk, &Dx);
    printf("| X | Y |\n"); // шапка таблицы
    float X = Xn; // установка параметра цикла
    while (X <= Xk){ // проверка условия продолжения
        printf("| %5.2f | %5.2f |\n", X, X*X + 1); // тело цикла
        X += Dx; // модификация параметра
    }
    return 0;
}
```

Пример (программа находит все делители целого положительного числа):

```
#include <iostream.h>
int main(){
    int num;
    cout << "\nВведите число : "; cin >> num;
    int half = num / 2; // половина числа
    int div = 2; // кандидат на делитель
    while (div <= half){
        if (!(num % div))cout << div << "\n";
        div++;
    }
    return 0;
}
```

Цикл с постусловием (do while)

Пример. Программа вычисляет квадратный корень вещественного аргумента X с заданной точностью Eps по итерационной формуле:

$$y_n = \frac{1}{2} (y_{n-1} + X/y_{n-1}).$$

где y_{n-1} — предыдущее приближение к корню (в начале вычислений выбирается произвольно), y_n — последующее приближение. Процесс вычислений прекращается, когда приближения станут отличаться друг от друга по абсолютной величине менее, чем на величину заданной точности. Для вычисления абсолютной величины используется стандартная функция `fabs()`, объявление которой находится в заголовочном файле `<math.h>`.

```
#include <stdio.h>
#include <math.h>
int main(){
    double X, Eps; // аргумент и точность
    double Yp, Y = 1; // предыдущее и последующее приближение
    printf("Введите аргумент и точность: ");
    scanf("%lf%lf", &X, &Eps);
    do{
        Yp = Y;
        Y = (Yp + X/Yp)/2;
    }while (fabs(Y - Yp) >= Eps);
    printf("\nКорень из %lf равен %lf", X, Y);
    return 0;
}
```

Цикл с параметром (for)

Пример (оператор, вычисляющий сумму чисел от 1 до 100):

```
for (int i = 1, s = 0; i <= 100; i++) s += i;
```

Пример (программа печатает таблицу значений функции $y=x^2+1$ во введенном диапазоне):

```
#include <stdio.h>
int main(){
    float Xn, Xk, Dx, X;
    printf("Введите диапазон и шаг изменения аргумента: ");
    scanf("%f%f%f", &Xn, &Xk, &Dx);
    printf("| X | Y |\n");
    for (X = Xn; X <= Xk; X += Dx)
        printf("| %5.2f | %5.2f |\n", X, X*X + 1);
    return 0;
}
```

Пример (программа находит все делители целого положительного числа):

```
#include <iostream.h>
int main(){
    int num, half, div;
    cout << "\nВведите число : "; cin >> num;
    for (half = num / 2, div = 2; div <= half; div++)
        if (!(num % div))cout << div << "\n";
    return 0;
}
```