

Операторы

Операторы

Оператор (statement) - установленное синтаксисом языка базовое действие в программе.

В C++ любое выражение, которое заканчивается символом *точка с запятой*, является оператором.

Оператор присваивания:

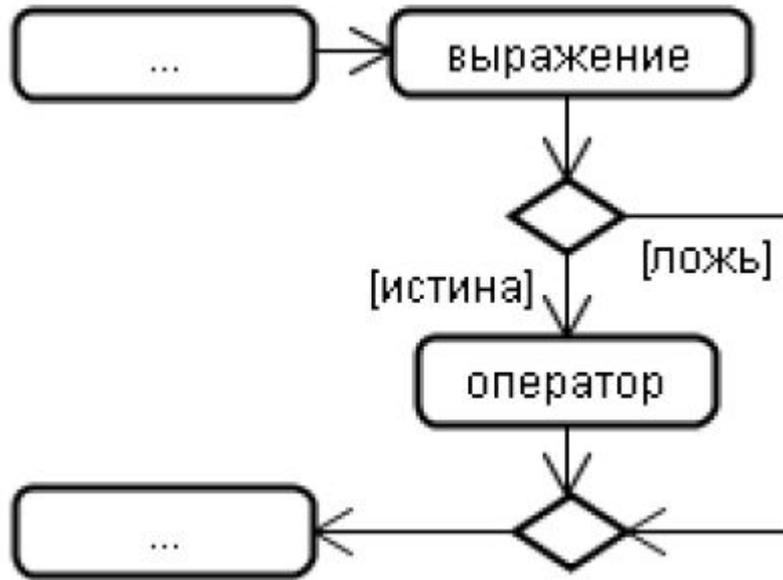
Идентификатор = Выражение ;

```
int a = 10;  
int b = 20;  
a = b;  
int c = a*2 + b;
```

Составной оператор (блок) состоит из двух или большего числа операторов любого типа, заключенных в фигурные скобки.

```
{  
    int d = 10;  
    c += d %3;  
}
```

Условный оператор



**if (выражение)
оператор**

```
if ( a == x )  
    temp = a ;
```

```
if ( ! c ) {  
    temp = x ;  
    x = y ;  
    y = temp ;  
}
```

Условный оператор

```
int a, b, c ;
cin >> a ; cin >> b ;      // a = 2, b = 3

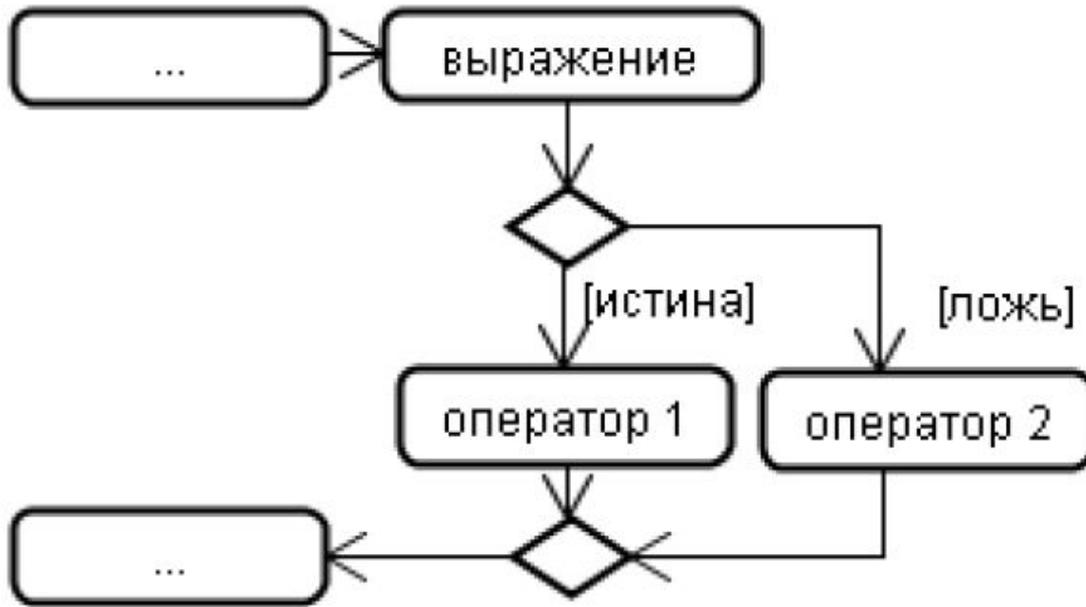
if ( a < b )
    cout << "a is less than b\n" ;
if ( a == b )
    cout << "you won't see this\n" ;

c = a - b ; // c = 1
if ( c >= 0 )
    cout << "c is non-negative\n" ;

if ( c < 0 )
    cout << "c is negative\n" ;

c = b - a ; // c = 1
if ( c >= 0 )
    cout << "c is non-negative\n" ;
if ( c < 0 )
    cout << "c is negative\n" ;
```

Условный оператор



```
if (выражение)  
    оператор 1  
else  
    оператор 2
```

```
if ( a == x )  
    temp = a ;  
else  
    temp = -a ;  
if ( ! k && ( x >= M && x <= N ) ) {  
    ++k  
    x *= k ;  
}  
else  
    x /= k ;
```

Условный оператор

```
#include <iostream>
using namespace std;

int main()
{
    int number;
    cout << "Enter an integer: ";
    cin >> number;

    if ( number >= 0)
    {
        cout << "You entered a positive integer: " << number << endl;
    }

    else
    {
        cout << "You entered a negative integer: " << number << endl;
    }

    cout << "This line is always printed.";
    return 0;
}
```

Условный оператор

Test expression is true

```
int test = 5;

if (test < 10)
{
  // codes
}

// codes after if
```



Test expression is false

```
int test = 5;

if (test > 10)
{
  // codes
}

// codes after if
```



Условный оператор

Test expression is true

```
int test = 5;

if (test < 10)
{
    // codes
}
else
{
    // codes
}
// codes after if...else
```



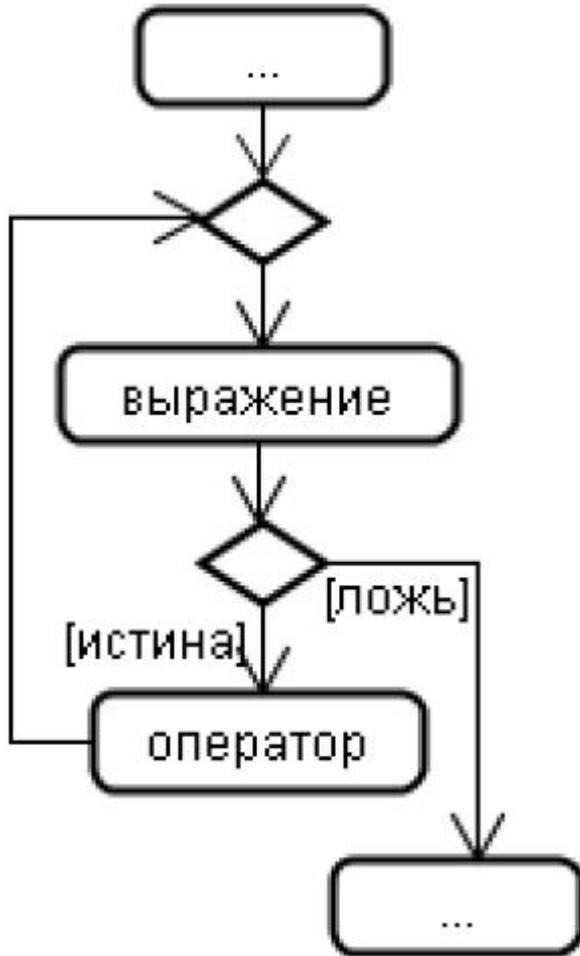
Test expression is false

```
int test = 5;

if (test > 10)
{
    // codes
}
else
{
    // codes
}
// codes after if...else
```



Цикл while



**while (выражение)
оператор**

```
static const int N (10) ;  
int x (1), sum (0) ;
```

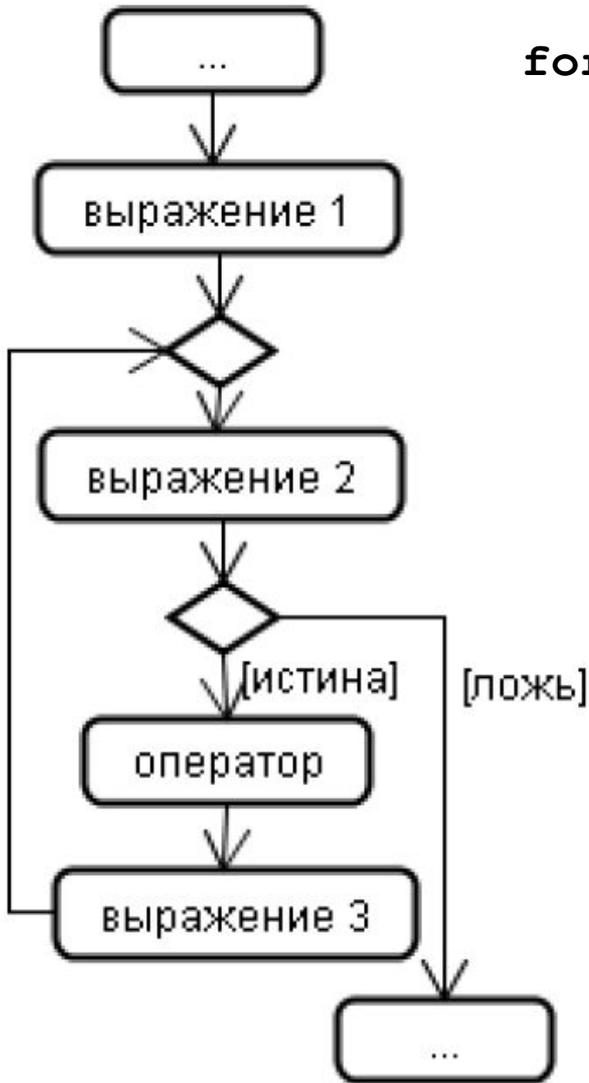
```
while ( x < N + 1 )  
{  
    sum += x ;  
    ++x ;  
}
```

```
cout << sum << endl ;
```

Цикл for

**for (выражение 1 ; выражение 2 ; выражение 3)
оператор**

```
static const int N (10) ;  
int sum ;  
for ( int x ( 1 ), sum = 0 ; x < N +  
1 ; ++x )  
    sum += x ;  
  
cout << sum << endl ;
```



Цикл do-while



do

оператор

while (выражение) ;

```
static const int N (10) ;  
int x (1), sum (0) ;
```

do

{

```
    sum += x ;
```

```
    ++x ;
```

```
} while ( x < N + 1 ) ;
```

```
cout << sum << endl ;
```

Примеры

```
if (x == 1)
    if (y == 1) cout << "x равно 1 и y равно 1";
    else cout << "x не равно 1";
```

```
if (x==1)
    {if (y==1) cout << "x равно 1 и y равно 1";}
else cout << "x не равно 1";
```

Примеры

```
#include <iostream>
using namespace std;

int main()
{
    int number, i = 1, factorial = 1;

    cout << "Enter a positive integer: ";
    cin >> number;

    while ( i <= number) {
        factorial *= i;          //factorial = factorial * i;
        ++i;
    }

    cout<<"Factorial of "<< number <<" = "<< factorial;
    return 0;
}
```

Примеры

```
#include <iostream>
using namespace std;

int main()
{
    int i, n, factorial = 1;

    cout << "Enter a positive integer: ";
    cin >> n;

    for (i = 1; i <= n; ++i) {
        factorial *= i;    // factorial = factorial * i;
    }

    cout<< "Factorial of "<<n<<" = "<<factorial;
    return 0;
}
```

Примеры

```
#include <iostream>
using namespace std;

int main()
{
    float number, sum = 0.0;

    do {
        cout<<"Enter a number: ";
        cin>>number;
        sum += number;
    }
    while(number != 0.0);

    cout<<"Total sum = "<<sum;

    return 0;
}
```

Оператор continue

`continue ;`

Оператор направляет поток управления внутри цикла. Он заставляет прерваться текущую итерацию цикла и немедленно начинать следующую. Оператор `continue` может использоваться **только внутри тела операторов `while`, `for` и `do while`**.

```
while (test expression) {  
    statement/s  
    if (test expression) {  
        continue;  
    }  
    statement/s  
}
```

```
do {  
    statement/s  
    if (test expression) {  
        continue;  
    }  
    statement/s  
} while (test expression);
```

```
for (initial expression; test expression; update expression) {  
    statement/s  
    if (test expression) {  
        continue;  
    }  
    statements/  
}
```

Оператор continue

```
int x, sum (0) ;
bool flag (true) ;

while ( flag ) {
    cout << "Enter x -> " ;
    cin >> x ;
    if ( x > 0 ) {
        sum += x ;
        continue ;
    }
    flag = false ;
}
cout << sum << endl ;
```

Оператор continue

```
#include <iostream>
using namespace std;

int main()
{
    for (int i = 1; i <= 10; ++i)
    {
        if ( i == 6 || i == 9)
        {
            continue;
        }
        cout << i << "\t";
    }
    return 0;
}
```

Оператор break

break ;

Оператор вызывает немедленный выход из цикла оператора while, for или do while, а также из оператора-переключателя switch. Управление передается оператору, непосредственно следующему за заканчиваемым оператором цикла или переключателем. Типичное использование оператора break состоит в том, чтобы завершить выполнение цикла досрочно.

```
while (test expression) {  
    statement/s  
    if (test expression) {  
        break;  
    }  
    statement/s  
}
```

```
do {  
    statement/s  
    if (test expression) {  
        break;  
    }  
    statement/s  
} while (test expression);
```

```
for (initial expression; test expression; update expression) {  
    statement/s  
    if (test expression) {  
        break;  
    }  
    statements/  
}
```

Оператор break

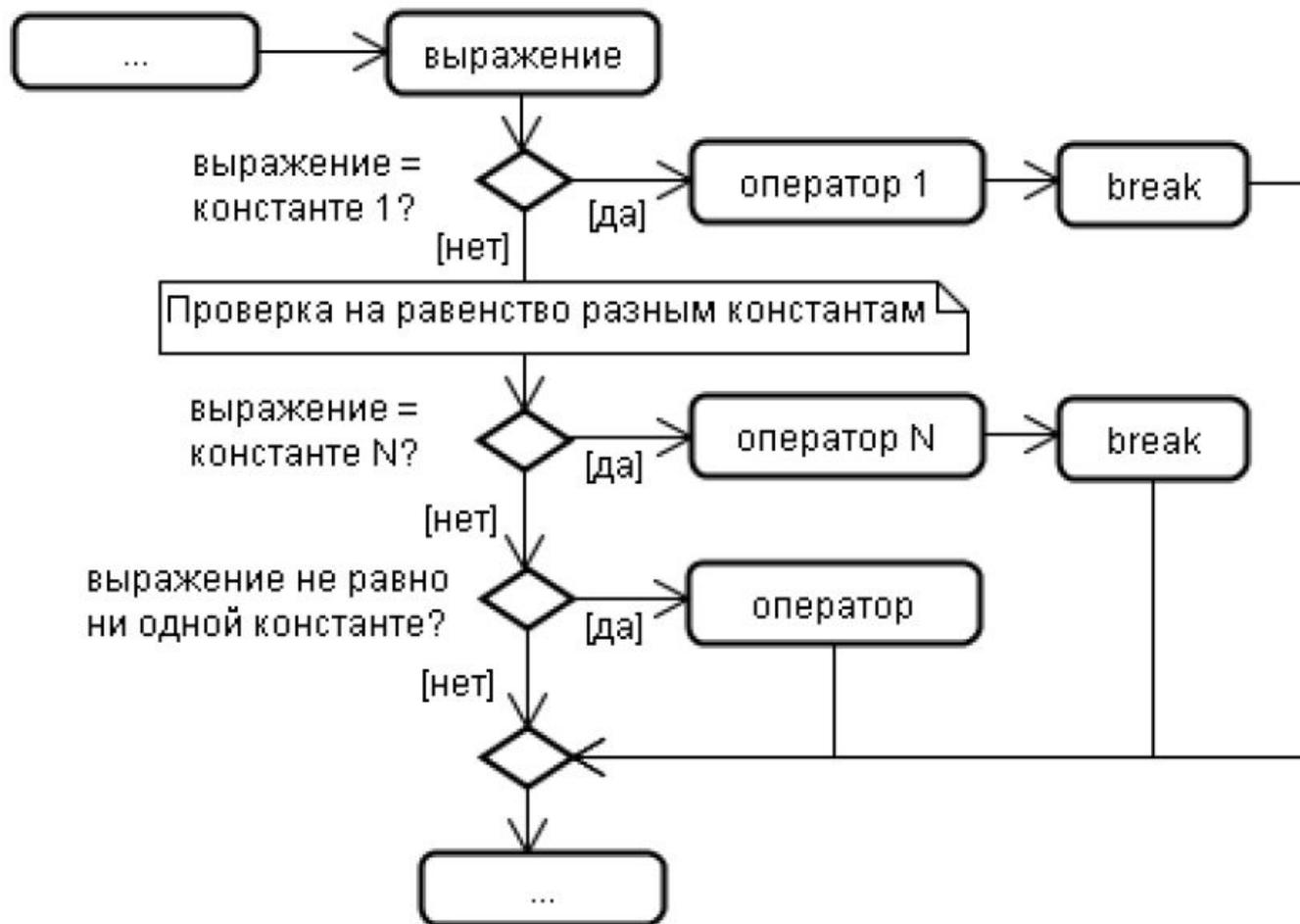
```
#include <iostream>
using namespace std;
int main() {
    float number, sum = 0.0;

    // test expression is always true
    while (true)
    {
        cout << "Enter a number: ";
        cin >> number;

        if (number != 0.0)
        {
            sum += number;
        }
        else
        {
            // terminates the loop if number equals 0.0
            break;
        }
    }
    cout << "Sum = " << sum;

    return 0;
}
```

Оператор switch



Оператор switch

```
switch (выражение)  
{  
  case константа 1 :  
    оператор 1  
  case константа 2 :  
    оператор 2  
  ...  
  case константа N :  
    оператор N  
  default :  
    оператор  
}
```

```
int x ; char choice ;  
cout << "Enter an integer -> " ;  
cin >> x ;  
cout << "Enter 1...3 -> " ;  
cin >> choice ;  
switch ( choice ) {  
  case '1' :  
    cout << x << endl ;  
    break ;  
  case '2' :  
    cout << x * x << endl ;  
    break ;  
  case '3' :  
    cout << x * x * x << endl ;  
    break ;  
  default :  
    cout << "Error of input\n" ;  
}
```

Оператор switch

```
char c ( ' ' ); int counterABC (0); int counterOTHERS (0);
cout << "Enter symbols -> ";
while ( c != '.' ) {
    cin >> c;
    switch ( c ) {
        case 'A':
        case 'a':
        case 'B':
        case 'b':
        case 'C':
        case 'c':
            ++counterABC;
            break;
        case '.':
            break;
        default:
            ++counterOTHERS;
    }
}
cout << "ABC or abc = " << counterABC << endl;
cout << "Other symbols = " << counterOTHERS << endl;
```

Оператор switch

```
void task1() {
    // ...
}

int main () {
    char choice = 0;
    bool isExit = false;
    do {
        cout << "1. f(x)" << endl;
        cout << "2. f(x1), f(x2)" <<
endl;
        cout << "3. f(x+d)" << endl;
        cout << "4. f(x)=0" << endl;
        cout << "5. Exit" << endl;
        cout << "Enter [1-5]: ";
        cin >> choice;

        switch (choice) {
            case '1':
                task1();
                break;
            case '2':
                task2();
                break;
            case '3':
                task3();
                break;
            case '4':
                task4();
                break;
            case '5':
                isExit = true;
                break;
            default:
                cout << "Please
                    enter [1-5]" << endl;
                break;
        }
    }
    while (!isExit);
    return 0;
}
```

Тернарный оператор

выражение 1 ? выражение 2 : выражение 3 ;

`k = (j < 0) ? (-j) : (j) ;`

`max = (a > b) ? a : b ;`

Оператор sizeof

sizeof (выражение) ;

```
cout << sizeof ( double ) << endl ;
```

```
int j ;
```

```
cout << sizeof ( j ) ;
```

Примеры

```
double y (double a, double b, double c, double x) {
    return a * x * x + b * x + c;
}
double f (double a, double b, double c, double x, double M) {
    if (x > M) {
        x = M;
    }
    else if (x < -M) {
        x = -M;
    }
    return y(a, b, c, x);
}
int main() {
    double a, b, c, M;
    cout << "Enter a: ";
    cin >> a;
    // ...

    // ...
    return 0;
}
```

Примеры

```
int choice = 0;
while (choice != 5) {
    cout << "1. Task 1" << endl;
    cout << "2. Task 2" << endl;
    cout << "3. Task 3" << endl;
    cout << "4. Task 4" << endl;
    cout << "5. Exit" << endl;
    cin >> choice;

    // ...
}
```

Примеры

```
switch (choice) {
    case 1:
        task1(a, b, c, M);
        break;
    case 2:
        task2();
        break;
    // ...
    case 5:
        break;
    default:
        cout << "Enter once again" << endl;
}
```

```
void task1 (double a, double b, double c, double M) {
    cout << "Enter x: " << endl;
    double x;
    cin >> x;
    cout << "f(x)= " << f (a, b, c, x, M) << endl;
}
```

```
// ...
```