

Программирование

Лекция 5

Вложенные циклы и двумерные массивы

массивы

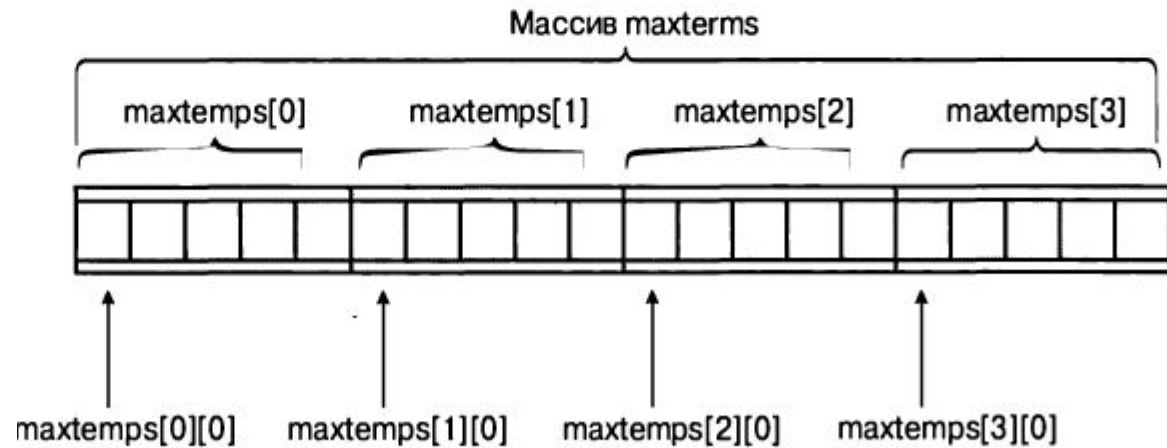
maxterms — массив из четырех элементов

```
int maxtemps[4][5];
```

```
int maxtemps[4][5];
```

Каждый элемент — массив из 5 значений int

**Первый индекс
можно
представлять
как строку
таблицы, а
второй — как ее
столбец**



	0	1	2	3	4	
maxtemps[0]	0	maxtemps[0][0]	maxtemps[0][1]	maxtemps[0][2]	maxtemps[0][3]	maxtemps[0][4]
maxtemps[1]	1	maxtemps[1][0]	maxtemps[1][1]	maxtemps[1][2]	maxtemps[1][3]	maxtemps[1][4]
maxtemps[2]	2	maxtemps[2][0]	maxtemps[2][1]	maxtemps[2][2]	maxtemps[2][3]	maxtemps[2][4]
maxtemps[3]	3	maxtemps[3][0]	maxtemps[3][1]	maxtemps[3][2]	maxtemps[3][3]	maxtemps[3][4]

Вложенные циклы и двумерные

МАССИВЫ

```
for (int row = 0; row < 4; row++)
{
    for (int col = 0; col < 5; ++col)
        cout << maxtemps[row][col] << "\t";
    cout << endl;
}
```

Инициализация двумерного массива

```
// Инициализация одномерного массива
```

```
int btus[5] = {23, 26, 24, 31, 28};
```

```
int maxtemps[4][5] = // двумерный массив
{
    {96, 100, 87, 101, 105}, // значения для maxtemps[0]
    {96, 98, 91, 107, 104}, // значения для maxtemps[1]
    {97, 101, 93, 108, 107}, // значения для maxtemps[2]
    {98, 103, 95, 109, 108} // значения для maxtemps[3]
};
```

МАССИВА

```

#include <iostream>
const int Cities = 5;
const int Years = 4;
int main()
{
    using namespace std;
    const char * cities[Cities] =
    {
        "Gribble City",
        "Gribbletown",
        "New Gribble",
        "San Gribble",
        "Gribble Vista"
    };
    int maxtemps[Years][Cities] = // двумерный массив
    {
        {96, 100, 87, 101, 105}, // значения для maxtemps[0]
        {96, 98, 91, 107, 104}, // значения для maxtemps[1]
        {97, 101, 93, 108, 107}, // значения для maxtemps[2]
        {98, 103, 95, 109, 108} // значения для maxtemps[3]
    };
    cout << "Maximum temperatures for 2008 - 2011\n\n";
        // Максимальные температуры в 2008-2011 гг.
    for (int city = 0; city < Cities; ++city)
    {
        cout << cities[city] << ":\t";
        for (int year = 0; year < Years; ++year)
            cout << maxtemps[year][city] << "\t";
        cout << endl;
    }
    // cin.get();
    return 0;
}

```

Maximum temperatures for 2008 – 2011

Gribble City:	96	96	97	98
Gribbletown:	100	98	101	103
New Gribble:	87	91	93	95
San Gribble:	101	107	108	109
Gribble Vista:	105	104	107	108

Использование двумерного массива

Использование массива массивов char вместо массива указателей для строковых данных:

```
char cities[Cities][25] =           // массив из 5 массивов по 25 символов
{
    "Gribble City",
    "Gribbletown",
    "New Gribble",
    "San Gribble",
    "Gribble Vista"
};
```

Массив указателей более экономичен в отношении используемой памяти!

Использование массива объектов класса string вместо массива указателей для строковых данных:

```
const string cities[Cities] =       // массив из 5 строк
{
    "Gribble City",
    "Gribbletown",
    "New Gribble",
    "San Gribble",
    "Gribble Vista"
};
```

Резюме

- В C++ представлены три варианта циклов: *for*, *while* и *do while*.
- Цикл позволяет повторно выполнять один и тот же набор инструкций до тех пор, пока проверочное условие цикла оценивается как true, и цикл прекращает их выполнение, когда это проверочное условие возвращает false.
- Циклы *for* и *while* являются циклами с проверкой на входе, это означает, что они оценивают проверочное условие перед выполнением операторов, находящихся в теле цикла.
- Цикл *do while* проверяет условие на выходе, т. е. после выполнения операторов, содержащихся в его теле.

Оператор if

```
if (проверочное-условие)  
    оператор
```

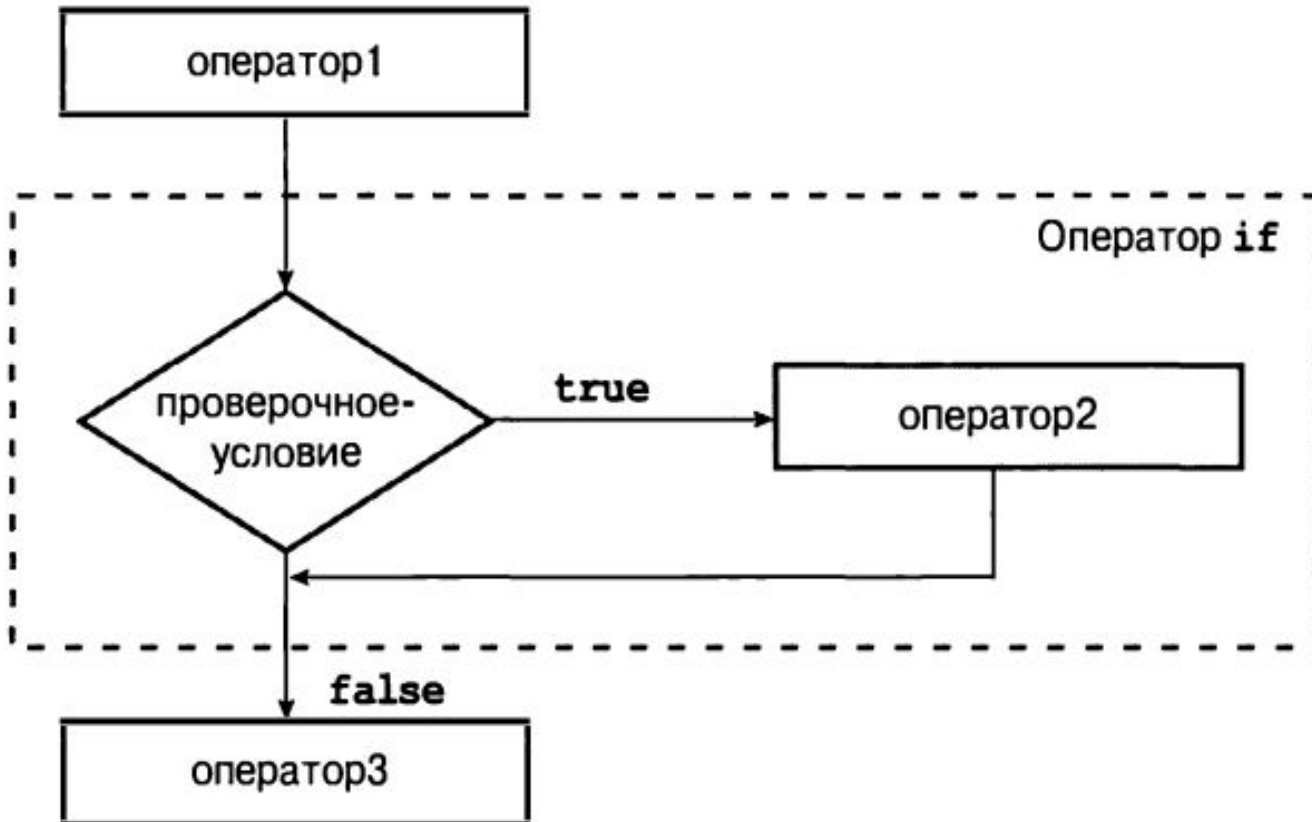
```
#include <iostream>  
int main()  
{  
    using std::cin;    // объявления using  
    using std::cout;  
    char ch;  
    int spaces = 0;  
    int total = 0;  
    cin.get(ch);  
    while (ch != '.') // завершение по окончании предложения  
    {  
        if (ch == ' ') // проверка ch на равенство пробелу  
            ++spaces;  
        ++total; // выполняется на каждом шаге цикла  
        cin.get(ch);  
    }  
    cout << spaces << " spaces, " << total; // вывод количества пробелов  
                                             // и символов в предложении  
    cout << " characters total in sentence\n";  
    return 0;  
}
```

```
The balloonist was an airhead  
with lofty goals.
```

```
6 spaces, 46 characters total in sentence
```

Структура оператора if

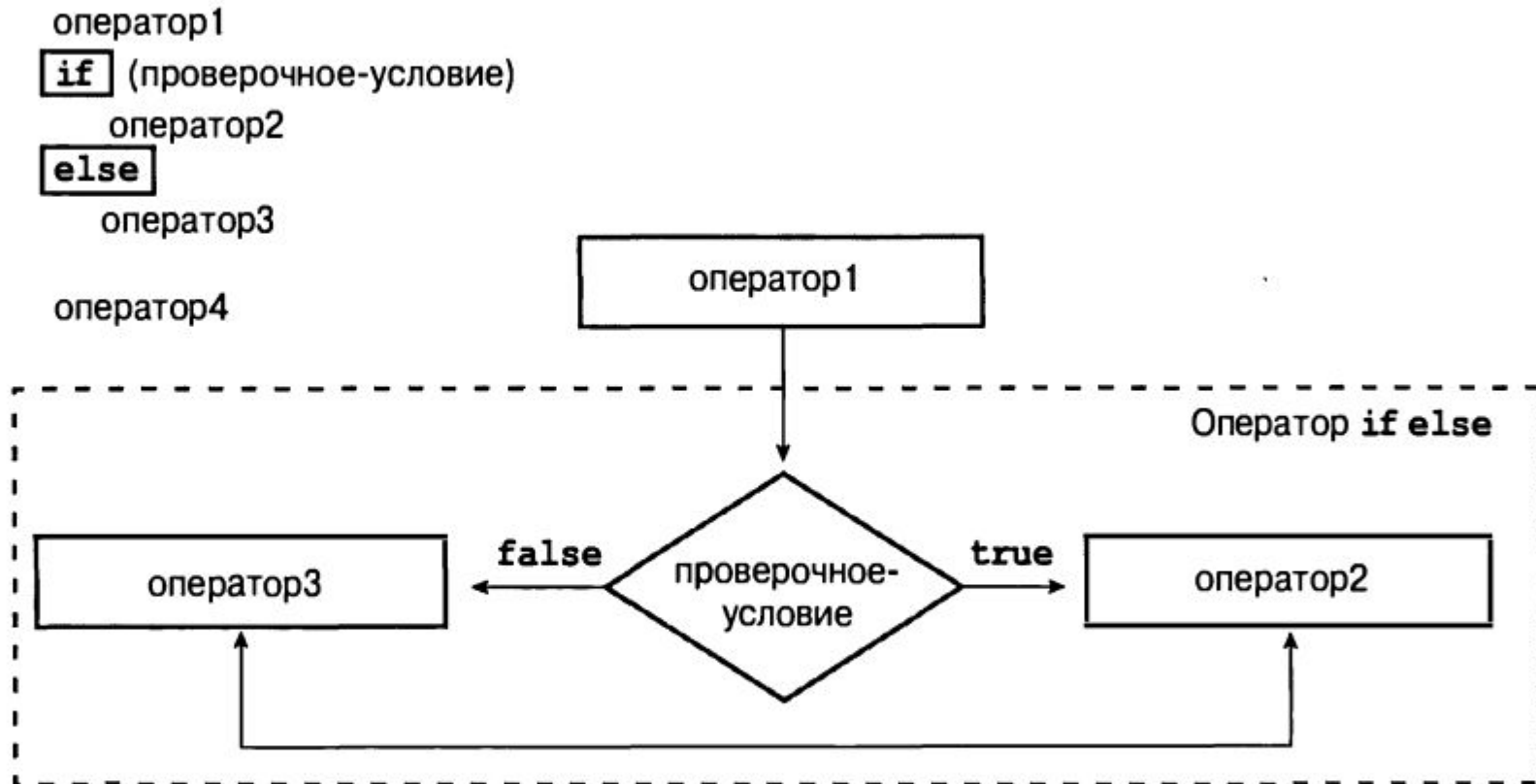
оператор1
if (проверочное-условие)
 оператор2
оператор3



Оператор if else

```
if (проверочное-условие)  
    оператор1  
else  
    оператор2
```

```
if (answer == 1492)  
    cout << "That's right!\n";  
else  
    cout << "You'd better review Chapter 1 again.\n";
```



Форматирование операторов if else

Ошибка при компиляции:

```
if (ch == 'Z')
    zorro++;           // if заканчивается здесь
    cout << "Another Zorro candidate\n";
else
    dull++;           // неверно
cout << "Not a Zorro candidate\n";
```

```
if (ch == 'Z')
{
    zorro++;           // блок, выходящий за пределы if
    cout << "Another Zorro candidate\n";
}
else
{
    dull++;           // блок, выходящий за пределы if
    cout << "Not a Zorro candidate\n";
}

if (ch == 'Z') {
    zorro++;
    cout << "Another Zorro candidate\n";
}
else {
    dull++;
    cout << "Not a Zorro candidate\n";
}
```

Конструкция if else if else

```
#include <iostream>
const int Fave = 27;
int main()
{
    using namespace std;
    int n;
    cout << "Enter a number in the range 1-100 to find ";
    cout << "my favorite number: "; // запрос на ввод числа из диапазона 1-100
    do
    {
        cin >> n;
        if (n < Fave)
            cout << "Too low -- guess again: "; // число слишком мало
        else if (n > Fave)
            cout << "Too high -- guess again: "; // число слишком велико
        else
            cout << Fave << " is right!\n"; // число угадано
    } while (n != Fave);
    return 0;
}
```

```
Enter a number in the range 1-100 to find my favorite number: 50
Too high -- guess again: 25
Too low -- guess again: 37
Too high -- guess again: 31
Too high -- guess again: 28
Too high -- guess again: 27
27 is right!
```

Условные операции и предотвращение ошибок

переменная == значение



значение == переменная

- `if (3 == myNumber)`
- `if (3 = myNumber)`
- `if (myNumber =3)`

Логическая операция "ИЛИ":

||

```
5 == 5 || 5 == 9    // истинно, потому что первое выражение истинно
5 > 3 || 5 > 10     // истинно, потому что первое выражение истинно
5 > 8 || 5 < 10     // истинно, потому что второе выражение истинно
5 < 8 || 5 > 2      // истинно, потому что оба выражения истинны
5 > 8 || 5 < 2      // ложно, потому что оба выражения ложны
```

Поскольку || имеет более низкий приоритет, чем операции сравнения, нет необходимости использовать в этих выражениях скобки!

Значение <code>expr1 expr2</code>		
	<code>expr1 == true</code>	<code>expr1 == false</code>
<code>expr2 == true</code>	true	true
<code>expr2 == false</code>	true	false

ИСПОЛЬЗОВАНИЕ ЛОГИЧЕСКОЙ ОПЕРАЦИИ

«ИЛИ»

```
#include <iostream>
int main()
{
    using namespace std;
    cout << "This program may reformat your hard disk\n"
          "and destroy all your data.\n"
          "Do you wish to continue? <y/n> ";
    char ch;
    cin >> ch;
    if (ch == 'y' || ch == 'Y')           // y или Y
        cout << "You were warned!\a\a\n";
    else if (ch == 'n' || ch == 'N')     // n или N
        cout << "A wise choice ... bye\n";
    else
        cout << "That wasn't a y or n! Apparently you "
              "can't follow\ninstructions, so "
              "I'll trash your disk anyway.\a\a\a\n";
    return 0;
}
```

```
This program may reformat your hard disk
and destroy all your data.
Do you wish to continue? <y/n> N
A wise choice ... bye
```

Логическая операция "И": &&

```
5 == 5 && 4 == 4      // истинно, потому что оба выражения истинны
5 == 3 && 4 == 4      // ложно, потому что первое выражение ложно
5 > 3 && 5 > 10       // ложно, потому что второе выражение ложно
5 > 8 && 5 < 10       // ложно, потому что первое выражение ложно
5 < 8 && 5 > 2        // истинно, потому что оба выражения истинны
5 > 8 && 5 < 2        // ложно, потому что оба выражения ложны
```

Поскольку && имеет более низкий приоритет, чем операции сравнения, нет необходимости использовать в этих выражениях скобки!

Значение <code>expr1 && expr2</code>		
	<code>expr1 == true</code>	<code>expr1 == false</code>
<code>expr2 == true</code>	true	false
<code>expr2 == false</code>	false	false

ИСПОЛЬЗОВАНИЕ ЛОГИЧЕСКОЙ ОПЕРАЦИИ

"И"

```
#include <iostream>
const int ArSize = 6;
int main()
{
    using namespace std;
    float naaq[ArSize];
    cout << "Enter the NAAQs (New Age Awareness Quotients) "
         << "of\nyour neighbors. Program terminates "
         << "when you make\n" << ArSize << " entries "
         << "or enter a negative value.\n";

    int i = 0;
    float temp;
    cout << "First value: ";           // ввод первого значения
    cin >> temp;
    while (i < ArSize && temp >= 0) // два критерия завершения
    {
        naaq[i] = temp;
        ++i;
        if (i < ArSize)           // в массиве еще есть место
        {
            cout << "Next value: ";
            cin >> temp;           // ввод следующего значения
        }
    }
}
```


Установка диапазонов с помощью &&

```
#include <iostream>
const char * qualify[4] =           // массив указателей на строки
{
    "10,000-meter race.\n",         // забег на 10 000 метров
    "mud tug-of-war.\n",           // перетягивание каната в грязи
    "masters canoe jousting.\n",   // состязания мастеров каноэ
    "pie-throwing festival.\n"     // фестиваль по бросанию пирожков
};
int main ()
{
    using namespace std;
    int age;
    cout << "Enter your age in years: ";           // запрос возраста в годах
    cin >> age;
    int index;
    if (age > 17 && age < 35)
        index = 0;
    else if (age >= 35 && age < 50)
        index = 1;
    else if (age >= 50 && age < 65)
        index = 2;
    else
        index = 3;
    cout << "You qualify for the " << qualify[index]; // вывод рекомендованного результата
    return 0;
}
```

```
Enter your age in years: 87
You qualify for the pie-throwing festival.
```

Проверка диапазонов

```
if (age > 17 && age < 35) // Нормально
```

```
if (17 < age < 35) // Не делайте так!
```



```
if ((17 < age) < 35)
```

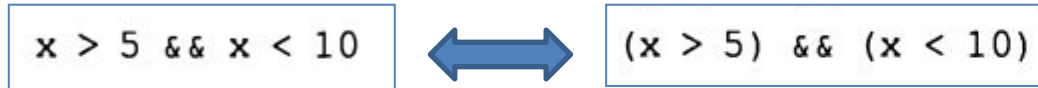
Логическая операция "НЕ": !

```
#include <iostream>
#include <climits>
bool is_int(double);
int main()
{
    using namespace std;
    double num;
    cout << "Yo, dude! Enter an integer value: "; // запрос на ввод целочисленного значения
    cin >> num;
    while (!is_int(num)) // продолжать, пока num не является int
    {
        cout << "Out of range -- please try again: "; // выход за пределы диапазона
        cin >> num;
    }
    int val = int (num); // приведение типа
    cout << "You've entered the integer " << val << "\nBye\n";
    return 0;
}
bool is_int(double x)
{
    if (x <= INT_MAX && x >= INT_MIN)
        return true;
    else
        return false;
}
```

```
Yo, dude! Enter an integer value: 6234128679
Out of range -- please try again: -8000222333
Out of range -- please try again: 99999
You've entered the integer 99999
Bye
```

Порядок выполнения логических операций

логические операции "ИЛИ" и "И" в C++
обладают более низким приоритетом, чем
операции сравнения



операция "НЕ" (!) имеет более высокий
приоритет, чем любая
арифметическая операция и операция
сравнения

```
!(x > 5) // равно false, если x больше 5  
!x > 5  // равно true, если !x больше 5
```

Всегда false

Логическая операция "И" имеет более высокий приоритет, чем
логическая операция "ИЛИ“:

```
age > 30 && age < 45 || weight > 300
```



```
(age > 30 && age < 45) || weight > 300
```

Альтернативные представления

Операция	Альтернативное представление
<code>&&</code>	<code>and</code>
<code> </code>	<code>or</code>
<code>!</code>	<code>not</code>

Операция ?

? – условная операция

выражение1 ? выражение2 : выражение3

5 > 3 ? 10 : 12 // 5 > 3 истинно, поэтому значением всего выражения будет 10
3 == 9? 25 : 18 // 3 == 9 ложно, поэтому значением всего выражения будет 18

```
#include <iostream>
int main()
{
    using namespace std;
    int a, b;
    cout << "Enter two integers: ";           // запрос на ввод двух целых чисел
    cin >> a >> b;
    cout << "The larger of " << a << " and " << b;
    int c = a > b ? a : b;                   // c = a, если a > b, иначе c = b
    cout << " is " << c << endl;           // вывод большего из указанных чисел
    return 0;
}
```

Enter two integers: **25 28**
The larger of 25 and 28 is 28

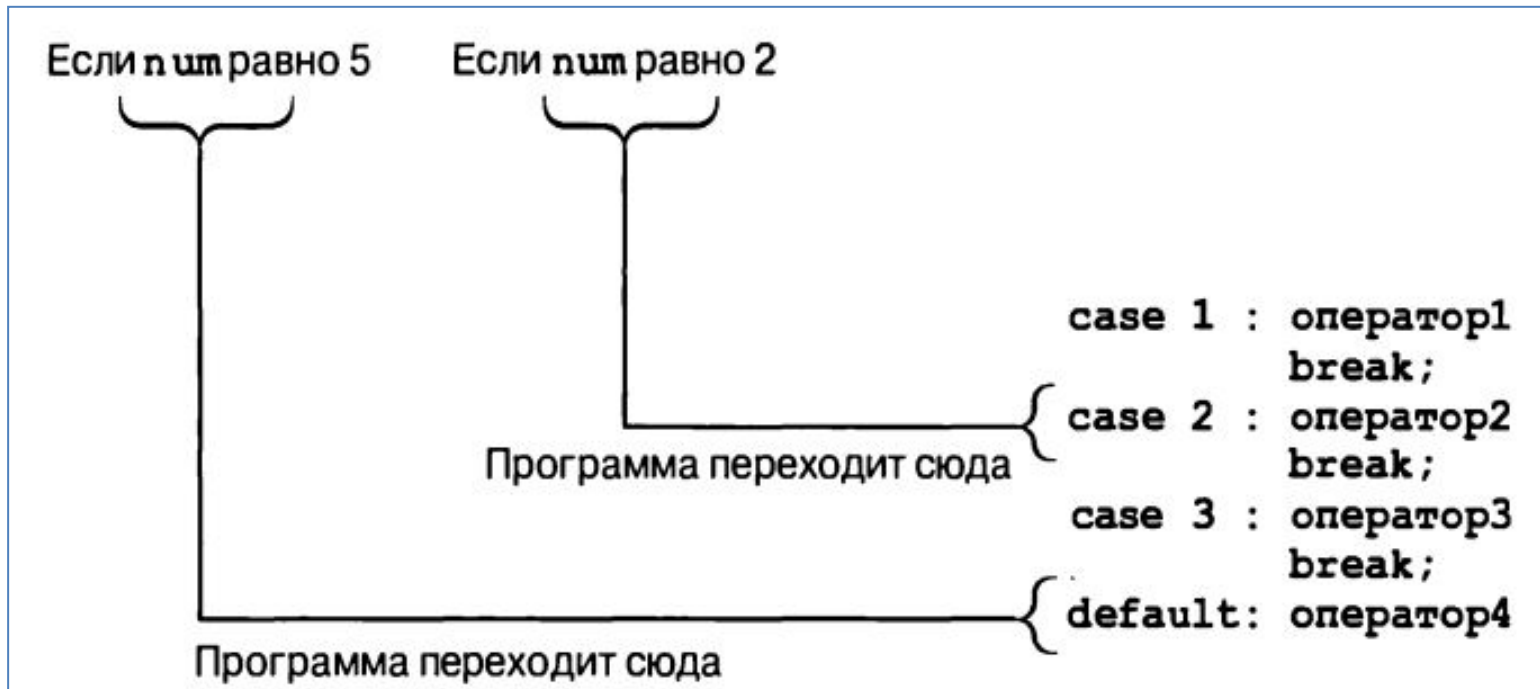
```
int c = a > b ? a : b;
```



```
int c;
if (a > b)
    c = a;
else
    c = b;
```

Оператор switch

```
switch (целочисленное-выражение)
{
    case метка1 : оператор(ы)
    case метка2 : оператор(ы)
    ...
    default : оператор(ы)
}
```



Оператор switch

```
#include <iostream>
using namespace std;
void showmenu(); // прототипы функций
void report();
void comfort();
int main()
{
    showmenu();
    int choice;
    cin >> choice;
    while (choice != 5)
    {
        switch(choice)
        {
            case 1 : cout << "\a\n";
                    break;
            case 2 : report();
                    break;
            case 3 : cout << "The b";
                    break;
            case 4 : comfort();
                    break;
            default : cout << "That's not a choice.\n";
        }
        showmenu();
        cin >> choice;
    }
    cout << "Bye!\n";
    return 0;
}
```

```
void showmenu()
{
    cout << "Please enter 1, 2, 3, 4, or 5:\n"
          "1) alarm 2)      report\n"
          "3) alibi 4)      comfort\n"
          "5) quit\n";
}

void report()
{
    cout << "It's been an excellent week for business.\n"
          "Sales are up 120%. Expenses are down 35%.\n";
}

void comfort()
{
    cout << "Your employees think you are the finest CEO\n"
          "in the industry. The board of directors think\n"
          "you are the finest CEO in the industry.\n";
}
```


Оператор switch

```
char choice;
cin >> choice;
while (choice != 'Q' && choice != 'q')
{
    switch(choice)
    {
        case 'a':
        case 'A': cout << "\a\n";
                break;
        case 'r':
        case 'R': report();
                break;
        case 'l':
        case 'L': cout << "The boss was in all day.\n";
                break;
        case 'c':
        case 'C': comfort();
                break;
        default : cout << "That's not a choice.\n";
    }
    showmenu();
    cin >> choice;
}
```

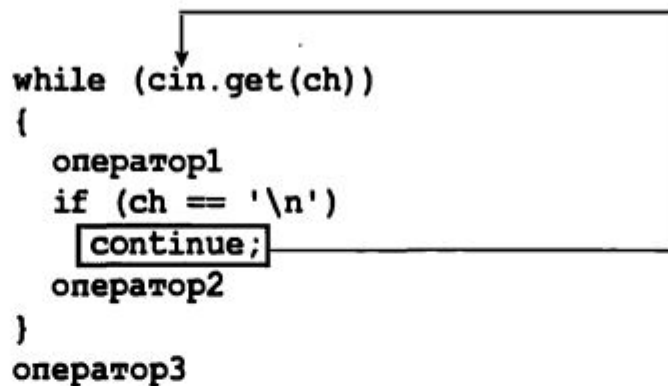
Использование перечислителей в качестве меток

```
#include <iostream>
// создание именованный констант для значений 0 - 6
enum {red, orange, yellow, green, blue, violet, indigo};
int main()
{
    using namespace std;
    cout << "Enter color code (0-6): "; // ввод кода цвета
    int code;
    cin >> code;
    while (code >= red && code <= indigo)
    {
        switch (code)
        {
            case red      : cout << "Her lips were red.\n"; break;
            case orange   : cout << "Her hair was orange.\n"; break;
            case yellow   : cout << "Her shoes were yellow.\n"; break;
            case green    : cout << "Her nails were green.\n"; break;
            case blue     : cout << "Her sweatsuit was blue.\n"; break;
            case violet   : cout << "Her eyes were violet.\n"; break;
            case indigo   : cout << "Her mood was indigo.\n"; break;
        }
        cout << "Enter color code (0-6): ";
        cin >> code;
    }
    cout << "Bye\n";
    return 0;
}
```

```
Enter color code (0-6): 3
Her nails were green.
Enter color code (0-6): 5
Her eyes were violet.
Enter color code (0-6): 2
Her shoes were yellow.
Enter color code (0-6): 8
Bye
```

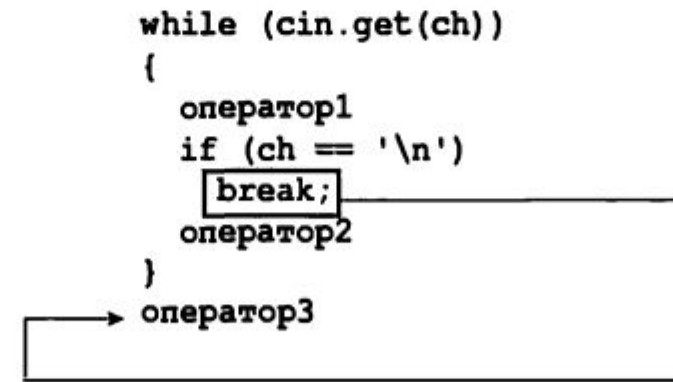
Операторы break и continue

```
while (cin.get(ch))
{
    оператор1
    if (ch == '\n')
        continue;
    оператор2
}
оператор3
```



Оператор **continue** пропускает остаток тела цикла и начинает новую итерацию

```
while (cin.get(ch))
{
    оператор1
    if (ch == '\n')
        break;
    оператор2
}
оператор3
```



Оператор **break** пропускает цикл и переходит на оператор, следующий за циклом

Использование операторов `continue` и `break`

```
#include <iostream>
const int ArSize = 80;
int main()
{
    using namespace std;
    char line[ArSize];
    int spaces = 0;
    cout << "Enter a line of text:\n";           // запрос на ввод строки текста
    cin.get(line, ArSize);
    cout << "Complete line:\n" << line << endl; // вывод полной строки
    cout << "Line through first period:\n";     // вывод строки до первой точки
    for (int i = 0; line[i] != '\0'; i++)
    {
        cout << line[i];                       // отображение символа
        if (line[i] == '.')                   // завершение, если это точка
            break;
        if (line[i] != ' ')                   // пропуск оставшейся части цикла
            continue;
        spaces++;
    }
    cout << "\n" << spaces << " spaces\n";
    cout << "Done.\n";
    return 0;
}
```

```
Enter a line of text:
Let's do lunch today. You can pay!
Complete line:
Let's do lunch today. You can pay!
Line through first period:
Let's do lunch today.
3 spaces
Done.
```

Вопросы для самоконтроля

- В чем состоит разница между циклами с проверкой на входе и циклами с проверкой на выходе? Какой из циклов C++ к какой категории относится?
- Что напечатает следующий фрагмент кода, если использовать его в программе?

```
int i;  
for (i = 0; i < 5; i++)  
    cout << i;  
    cout << endl;
```

- Что напечатает следующий фрагмент кода, если использовать его в программе?

```
int j;  
for (j = 0; j < 11; j += 3)  
    cout << j;  
cout << endl << j << endl;
```

Вопросы для самоконтроля

- Постройте логические выражения для представления перечисленных ниже условий:
- а. `weight` больше или равно 115, но меньше 125.
- б. `ch` равно `q` или `Q`.
- в. `x` — четное, но не равно 26.
- г. `x` — четное, но не кратно 2 6.
- д. `donation` находится в диапазоне 1000-2000 или `guest` равно 1.

- Напишите цикл `for`, который печатает значения 12 48 16 32 64, увеличивая вдвое значение переменной счетчика на каждом шаге.
- Как сделать так, чтобы тело цикла включало более одного оператора?