


***Основы алгоритмизации и
программирования***



Операторы выбора в C++.

Условный оператор.

Синтаксис условного оператора с одной ветвью (неполное ветвление):

```
if (условие)  
/* оператор */ ;
```

В случае нескольких операторов, операторы, которые должны выполняться в случае истинности условия заключаются в операторные скобки { }

```
if (условие)  
{  
/*группа операторов */ ;  
}
```



Лекция 3. Операторы выбора в C++.

В следующей программе для введенного числа проверяется его четность. В случае если число чётное выводится соответствующее сообщение. В противном случае на экран ничего выведено не будет.

```
// Пример 1 использования условного оператора
#include <iostream>;
#include <stdio.h>;
using namespace std;
void main() {
    setlocale(LC_ALL, "Russian");
    int value;
    cout << "Введите число: " << endl;
    cin >> value;
    if (value % 2 == 0)
    cout << "Число " << value << " чётное" << endl;
    getchar();
}
```



Лекция 3. Операторы выбора в C++.

Дано целое положительное число n ($n \leq 1000$). Составить программу, которая, если число чётное, выводит последнюю цифру этого числа.

```
// Пример 2 использования условного оператора
#include <iostream>;
#include <stdio.h>;
using namespace std;
void main() {
    setlocale(LC_ALL, "Russian");
    int value;
    cout << "Введите число: " << endl;
    cin >> value;
    if (value % 2 == 0)
        cout << "Последняя цифра числа " << value << " есть " <<
            value % 10 << endl;
    getchar();
    getchar();
}
```



Лекция 3. Операторы выбора в C++.

Синтаксис условного оператора с двумя ветвями (полное ветвление):

```
if (условие)
{
    /* оператор */ ;
}
else
{
    /* оператор */ ;
}
```

или, в случае нескольких операторов

```
if (условие)
{
    /*группа операторов */ ;
}
else
```

```
▶ {
    /* группа операторов */ ;
```

Лекция 3. Операторы выбора в C++.

В следующей программе для введенного числа проверяется его положительность. В случае если число положительно выводится соответствующее сообщение, если отрицательно — выводится соответствующее ему положительное число.

```
// Пример 3 использования условного оператора
#include <iostream>;
#include <stdio.h>;
using namespace std;
void main() {
    setlocale(LC_ALL, "Russian");
    int value;
    cout << "Введите число: " << endl;
    cin >> value;
    if (value > 0)
    cout << "Число " << value << " положительное" << endl;
    else
        cout << -value;
    getchar();
}
```



Лекция 3. Операторы выбора в C++.

Вложенные условные операторы

```
if (условие) {
    /*группа операторов */ ;
    if (условие) {
        /*группа операторов */ ;
    }
else {
    /*группа операторов */ ;
}
else {
    /*группа операторов */ ;
    if (условие) {
        /*группа операторов */ ;
    }
else {
    /*группа операторов */ ;
}
}
```



Лекция 3. Операторы выбора в C++.

Дано действительное число x . Составить программу, которая находит значение функции

$$f(x) = \begin{cases} |x+3|, & \text{если } x < -3, \\ \sqrt{x-6}, & \text{если } x \geq 6, \\ \frac{1}{x-10}, & \text{если } -3 \leq x < 6. \end{cases}$$

// Пример 4 использования условного оператора

```
#include <iostream>;
#include <stdio.h>;
#include <math.h>;
using namespace std;
void main() {
    setlocale(LC_ALL, "Russian");
    float x;
    cout << "Введите число: " << endl;
    cin >> x;
    if (x<-3)
        cout << abs(x+3) << endl;
    else
        if (x>=6)
            cout << sqrt(x-6) << endl;
        else
            cout << 1/(x-10) << endl;
    getchar();
}
```


Лекция 3. Операторы выбора в C++.

Оператор ? :

Оператор ? : позволяет создавать простые однострочные условные выражения, в которых выполняется одно из двух действий в зависимости от значения условия.

Синтаксис условного оператора ?:

Условие ? выражение1 : выражение2

Если условие истинно, то выполняется выражение1, если ложно, то выражение2.

Примеры

Использование оператора if/else	Использование оператора ? :
<code>If (a>0) a++ else a-- ;</code>	<code>a>0 ? a++ : a-- ;</code>
<code>If (a>0) cout<< a; else cout<< -a;</code>	<code>cout << (a>0 ? a : -a);</code>



Лекция 3. Операторы выбора в C++.

Оператор множественного выбора switch

Синтаксис оператора switch

```
switch (/* переменная или выражение */)
{
    case /* константное выражение1 */:
    {
        /* группа операторов */;
    break;
    }
    case /* константное выражение2 */:
    {
        /* группа операторов */;
        break;
    }
    // . . .
    default:
    {
        /*группа операторов*/;
    }
}
```



Лекция 3. Операторы выбора в C++.

Оператор множественного выбора switch


Написать программу, которая складывает, вычитает, умножает, делит два числа введенных с клавиатуры.

```
// Пример использования оператора выбора в switch
#include <iostream>
#include <stdio.h>;
using namespace std;
void main() {
    setlocale(LC_ALL, "Russian");
    int count; // переменная для выбора в switch
    double a,b; // переменные для хранения операндов
    cout << "Введите первое число: ";
    cin >> a;
    cout << "Введите второе число: ";
    cin >> b;
    cout << "Выберите операцию: 1-сложение; 2-вычитание; 3-умножение;
4-деление: ";
    cin >> count;
```



Написать программу, которая складывает, вычитает, умножает, делит два числа введенных с клавиатуры.

```
switch (count) // начало оператора switch
{
    case 1: { // если count = 1
        cout << a << " + " << b << " = " << a + b << endl; // выполнить сложение
        break;
    }
    case 2: { // если count = 2
        cout << a << " - " << b << " = " << a - b << endl; // выполнить вычитание
        break;
    }
    case 3: { // если count = 3
        cout << a << " * " << b << " = " << a * b << endl; // выполнить умножение
        break;
    }
    case 4: { // если count = 4
        cout << a << " / " << b << " = " << a / b << endl; // выполнить деление
        break;
    }
    default: { // если count равно любому другому значению
        cout << "Неверный ввод" << endl;
    }
    getchar();
    getchar();
}
```



Лекция 3. Операторы выбора в C++.

Операторы сравнения

Оператор	Значение
<	Меньше — возвращает true, если первый операнд меньше второго.
>	Больше — возвращает true, если первый операнд больше второго.
<=	Меньше или равно.
>=	Больше или равно.
==	Равенство. Возвращает true, если два операнда равны.
!=	Неравенство. Возвращает true, если два операнда неравны.



Лекция 3. Операторы выбора в C++.

Логические операторы

Логический оператор	Смысловое обозначение	Описание	Пример составного условия
&&	И	Составное условие истинно, если истинны оба простых условия	<code>x==2 && y>3</code>
	ИЛИ	Составное условие истинно, если истинно, хотя бы одно из простых условий	<code>x==2 y>3</code>
!	НЕ	Условие истинно, если условие, стоящее после символа ! ложно.	<code>!(x==2)</code>



Лекция 3. Операторы выбора в C++.

В языке программирования C операции с высшими приоритетами вычисляются первыми. Наивысшим приоритетом является приоритет равный 1. Операции с одинаковым приоритетом выполняются слева направо в порядке следования.

Сводная таблица операторов с учетом их приоритетов.

Приоритет	Оператор	Описание
1	++	Префиксный инкремент
	--	Префиксный декремент
	()	Вызов функции или подвыражение
	[]	Выделение элемента массива
	->	Указатель структуры
	.	Член структуры
2	!	Логическое отрицание
	~	Поразрядное логическое НЕ (двоичная инверсия)
	-	Унарный минус (изменение знака)
	+	Унарный плюс
	(type)	Преобразование к типу
	*	Разыменование указателя
	&	Определение адреса переменной
	sizeof	Определение размера в байтах

Сводная таблица операторов с учетом их приоритетов.

Приоритет	Оператор	Описание
3	*	Умножение
	/	Деление
	%	Остаток от деления
4	+	Сложение
	-	Вычитание
5	>>	Поразрядный сдвиг вправо
	<<	Поразрядный сдвиг влево
6	<	Меньше
	>	Больше
	<=	Меньше или равно
	>=	Больше или равно
7	==	Равно
	!=	Не равно
8	&	Поразрядное логическое И
9	^	Поразрядное исключающее ИЛИ
10		Поразрядное логическое ИЛИ

Сводная таблица операторов с учетом их приоритетов.

Приоритет	Оператор	Описание
11	&&	Логическое И
12		Логическое ИЛИ
13	?:	Оператор условия
14	=	Присваивание
	+=	Составное сложение
	-=	Составное вычитание
	*=	Составное умножение
	/=	Составное деление
	%=	Составное определение остатка от деления
	>>=	Составной поразрядный сдвиг вправо
	<<=	Составной поразрядный сдвиг влево
	&=	Составное поразрядное логическое И
	^=	Составное поразрядное исключающее ИЛИ
	=	Составное поразрядное логическое ИЛИ
15	,	Операция "запятая"
	++	Постфиксный инкремент
	--	Постфиксный декремент

Лекция 3. Операторы выбора в C++.

Задания для самостоятельной работы

13. Определите, что будет выведено на экран в результате работы следующей программы.

```
#include <iostream>;
#include <stdio.h>;
using namespace std;
void main() {
    int a=17;
    bool f = a<18 && a>5;
    if (f)
        cout << (a%10 < 5 ? a/10*2 : a/5+3) << endl;
    else
        cout << a << endl;
    getchar();
}
```

Наберите текст программы и проверьте правильность ответа. Запишите условие задачи, соответствующее данной программе.



Лекция 3. Операторы выбора в C++.

Задания для самостоятельной работы

14. Дано целое число x . Составьте программу, которая выясняет, верно ли, что $0 \leq x < 1$. На экран вывести <<Да>> или <<Нет>>.
15. Даны действительные числа x , y , z . Составьте программу, которая находит наибольшее из них.
16. Даны три числа a , b , c . Составьте программу, которая определяет, имеется ли среди них хотя бы одна пара равных между собой чисел.
17. Дано действительное число x . Составьте программу, которая вычисляет значение функции

$$f(x) = \begin{cases} |\sin(2x)|, & \text{если } x < 0, \\ e^{\sqrt{x-2}}, & \text{если } x \geq 2, \\ \ln(x+10), & \text{если } 0 \leq x < 2. \end{cases}$$



Лекция 3. Операторы выбора в C++.

Задания для самостоятельной работы

18. Заданы площади круга R и квадрата S . Составьте программу, которая определяет, поместится ли квадрат в круге (квадрат поместится в круге, если диагональ квадрата меньше или равна диаметру окружности).
 19. Дано целое трёхзначное положительное число n . Составьте программу, которая определяет, входит ли в него цифра 6.
 20. Дано целое трёхзначное положительное число n . Составьте программу, которая определяет, кратно ли трем произведение его цифр.
 21. Дано целое четырёхзначное положительное число n . Составьте программу, которая определяет, является ли оно палиндромом (<<перевёртышем>>) с учётом четырёх цифр. Например, числа 1221, 0660, 4554 будем считать палиндромами. (Палиндромом называется число, десятичная запись которого читается одинаково слева направо и справа налево.)
-



Лекция 3. Операторы выбора в C++.

Задания для самостоятельной работы

22. Поле шахматной доски определяется парой чисел (a, b) , каждое от 1 до 8, первое число задает номер столбца, второе – номер строки. Заданы две клетки. Определите, может ли шахматная ладья попасть с первой клетки на вторую за один ход. Входные данные: 4 целых числа от 1 до 8 каждое, первые два задают начальную клетку, вторые два задают конечную клетку. Начальные и конечные клетки не совпадают. Программа должна вывести YES, если из первой клетки ходом ладьи можно попасть во вторую или NO в противном случае.
 23. Билет на одну поездку в метро стоит 15 рублей, билет на 10 поездок стоит 125 рублей, билет на 60 поездок стоит 440 рублей. Пассажир планирует совершить n поездок. Определите, сколько билетов каждого вида он должен приобрести, чтобы суммарное количество оплаченных поездок было не меньше n , а общая стоимость приобретенных билетов – минимальна. Входные данные: количество поездок n . Выходные данные – три целых числа, равные необходимому количеству билетов на 1, на 10, на 60 поездок. Например, при вводе 129 программа должна вывести 0 1 2 (2 билета на 60 поездок и 1 билет на 10 поездок).
-



Лекция 3. Операторы выбора в C++.

Задания для самостоятельной работы

24. По введенному номеру дня недели вывести его название, используя оператор множественного выбора. Нумерация дней недели начинается с 1 – понедельник, 2 - вторник и т. д.
 25. Мастям игральных карт присвоены следующие порядковые номера: пики — 1, трефы — 2, бубны — 3, червы — 4; а достоинству карт: валету — 11, даме — 12, королю — 13, тузу — 14 (порядковые номера остальных достоинств соответствуют их названию: шестерке — 6, семерке — 7 и т.д.) Составьте программу, которая по заданному номеру масти (от 1 до 4) и достоинству карты (от 6 до 14) определяет полное название (масть и достоинство) соответствующей карты, например, «Король бубен», «Десятка пик» и т.п.
-

