



9-11 классы

Программирование на C++

Презентация занятия

Условия ветвления. Цикл while.

2 занятие



Минцифры
России



ЦИФРОВАЯ
ЭКОНОМИКА

20.35
УНИВЕРСИТЕТ

Программирование
на C++

Теоретическая часть

Условия ветвления.
Цикл while.

2 занятие



инжинириум®

МГТУ им. Н.Э. Баумана

2019

Тема: Условия ветвления. Цикл while.

Любая программа, написанная на языке программирования – это реализация алгоритма.

Алгоритм — конечная совокупность точно заданных правил решения произвольного класса задач или набор инструкций, описывающих порядок действий исполнителя для решения некоторой задачи.

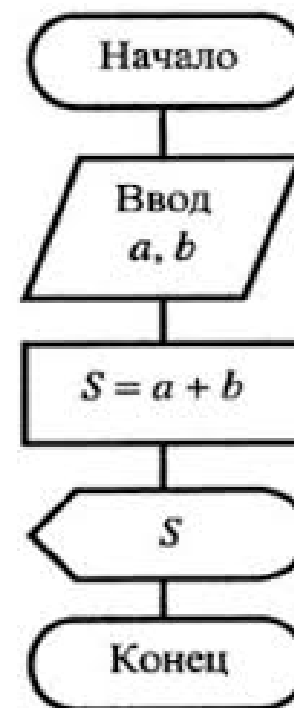
Другими словами, алгоритм – это последовательность команд.

Алгоритм можно реализовать на разных языках программирования, но суть алгоритма остаётся неизменной.



Тема: Условия ветвления. Цикл while.

```
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      int a = 5, b = 10;
7      int s;
8      s = a + b;
9      cout << s << endl;
10     return (0);
11 }
```



Тема: Условия ветвления. Цикл while.

До сегодняшнего для программы, которые мы писали выполнялись построчно (строчка за строчкой), но очень часто бывает необходимо выполнять определённые действия в зависимости от конкретных условий.



Тема: Условия ветвления. Цикл while.

Выполнение решения

Выражение **if** используется для выполнения некоторого кода при соответствии условиям.

```
if (condition)
{
    statements
}
```

Условие определяет, какое выражение будет выполнено. Если условие истинно, то выражение в фигурных скобках будет выполнено. Если условие ложно, то выражение просто игнорируется, и программа продолжает выполнение после тела оператора if.



Тема: Условия ветвления. Цикл while.

```
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      cout << "Output:" << endl;
7      cout << (7 > 4) << endl;
8      if (7 > 4)
9      {
10         cout << "Yes";
11     }
12     return (0);
13 }
```

Оператор if

Используйте операторы отношения для работы с условиями.

Условие, указанное в операторе if, не требует точки с запятой.

Output:

1

Yes



Тема: Условия ветвления. Цикл while.

Операторы отношения

Дополнительные операторы отношения:

Оператор	Описание	Пример
\geq	Больше или равно	$7 \geq 4$ Истина
\leq	Меньше или равно	$7 \leq 4$ Ложь
$==$	Равно	$7 == 4$ Ложь
$!=$	Не равно	$7 != 4$ Истина

Тема: Условия ветвления. Цикл while.

Оператор else

Оператор if может использоваться вместе с оператором else, который выполняется, когда условие ложно.

Синтаксис:

if (condition)

{

 //statements

}

else

{

 //statements

}

Код проверит условие:

- Если оно истинно, то код внутри выражения if будет выполнен.
- Если оно ложно, то будет выполнен код внутри выражения else.





Тема: Условия ветвления. Цикл while.

```
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      int mark = 90;
7      cout << "Output:" << endl;
8      if (mark < 50) {
9          cout << "You failed." << endl;
10     }
11     else {
12         cout << "You passed." << endl;
13     }
14     return (0);
15 }
```

Output:
You passed.





Тема: Условия ветвления. Цикл while.

Вложенный оператор if else

C++ обеспечивает опцией неограниченного использования вложенных операторов if/else. Например:

Output:
You passed.
Perfect!

```
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      int mark = 100;
7      cout << "Output:" << endl;
8      if (mark >= 50)
9      {
10         cout << "You passed." << endl;
11         if (mark == 100)
12         {
13             cout << "Perfect!" << endl;
14         }
15     }
16     else
17     {
18         cout << "You failed." << endl;
19     }
20     return (0);
21 }
```



Тема: Условия ветвления. Цикл while.

Конструкция else if

Помимо проверок условий if и else имеется возможность добавления «промежуточных проверок» else if.

```
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      int mark = 90;
7      cout << "Output:" << endl;
8      if (mark < 50)
9      {
10         cout << "You failed." << endl;
11     }
12     else if (mark > 50)
13     {
14         cout << "You passed." << endl;
15     }
16     else
17     {
18         cout << "You are lucker. There is 50!" << endl;
19     }
20     return (0);
21 }
```

Тема: Условия ветвления. Цикл while.

Логические операторы

Помимо арифметических существует ещё и логические операторы (как и алгебра логики)

Используйте логические операторы для комбинации условных операторов и возвращения true(истина) или false(ложь).

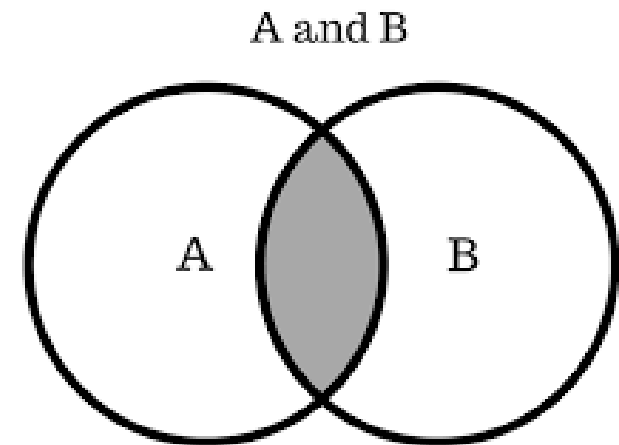
Operator	Name of Operator	Form
&&	AND Operator	y && y
	OR Operator	x y
!	NOT Operator	! x

Тема: Условия ветвления. Цикл while.

Логические операторы

Оператор И работает следующим образом:

Left Operand	Right Operand	Result
false	false	false
false	true	false
true	false	false
true	true	true



В операторе И обе операнды должны быть верны, чтобы результат выражения был также истинным.



Тема: Условия ветвления. Цикл while.

Логические операторы

Выражение в операторе if будет верно, только если оба выражения верны.

```
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      int age = 20;
7      if (age > 16 && age < 60) {
8          cout << "Accepted!" << endl;
9      }
10
11     return 0;
12 }
```

В одиночном операторе if логические операторы могут быть использованы для комбинации множественных условий.

Все выражение будет считаться верным только если все условия верны.

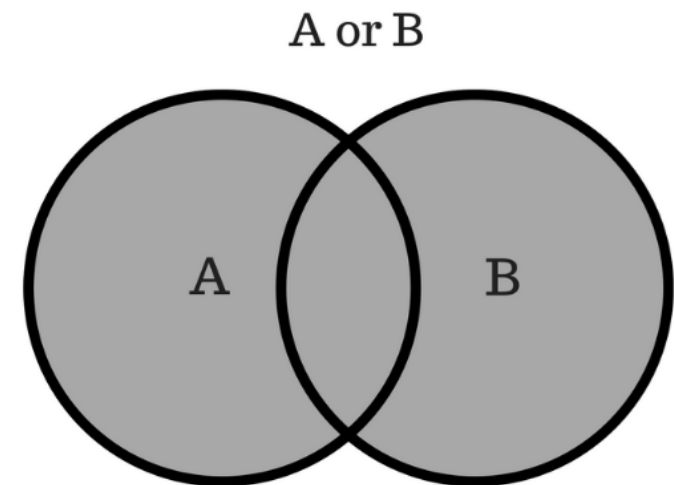


Тема: Условия ветвления. Цикл while.

Оператор ИЛИ

Оператор ИЛИ (`||`) возвращает `true`(верно), если один из операторов `true`(верный).

Left Operand	Right Operand	Result
false	false	false
false	true	true
true	false	true
true	true	true



Тема: Условия ветвления. Цикл while.

Логические операторы

Вы можете комбинировать вместе любое количество логических операторов ИЛИ.

Также, множество операторов ИЛИ и И может быть использовано вместе.

```
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      int age = 20;
7      int grade = 80;
8
9      if (age > 16 && age < 60 && grade > 50) {
10         cout << "Accepted!" << endl;
11     }
12
13     return 0;
14 }
```

```
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      int age = 16;
7      int score = 90;
8      if (age > 20 || score > 50) {
9         cout << "Accepted!" << endl;
10     }
11
12     return 0;
13 }
```

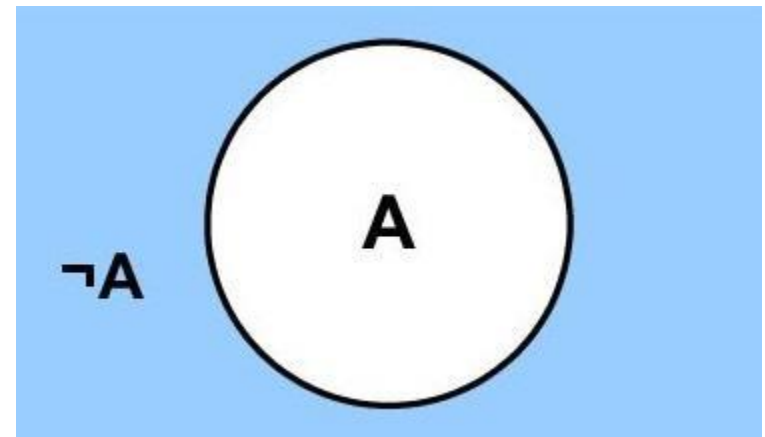
Тема: Условия ветвления. Цикл while.

Логические операторы

Логическое НЕ

Логический оператор НЕ (!) работает только с одним операндом, изменяя его логическое состояние. Поэтому, если состояние верно, оператор НЕ сделает его ложным, и также наоборот.

**Будьте осторожны, используя это,
потому что !false означает true.**



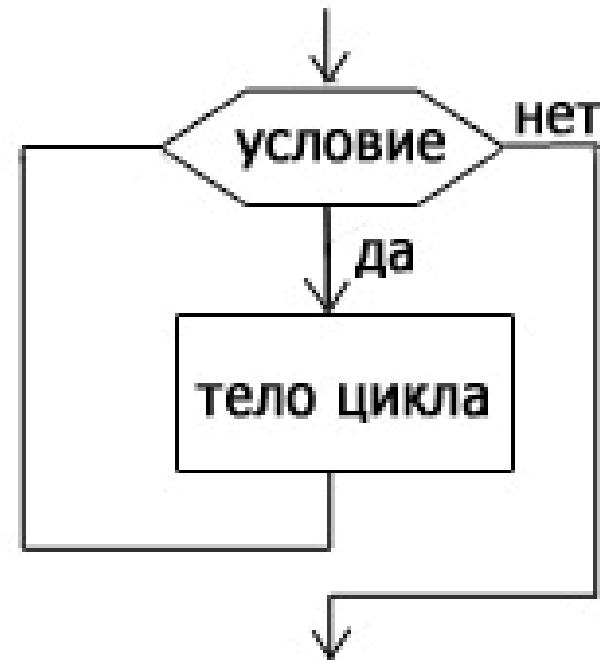
Тема: Условия ветвления. Цикл while.

Циклы

Цикл повторно выполняет набор выражений до выполнения условия. Цикл while повторно выполняет заданное выражение, пока условие истинно.

Если условие истинно, то происходит итерация цикла. В точке, где условие становится ложным, выполнение программы смещается в строку, последующую за циклом.

Телом цикла называется блок выражений внутри фигурных скобок.



```
while (condition)
{
    statements;
}
```



Тема: Условия ветвления. Цикл while.

Цикл while

В приведенном ниже примере объявлена переменная равная 1 (`int num = 1`). Цикл `while` проверяет условие (`num < 6`), и выполняет выражения в теле цикла, которые увеличивают значение `num` на единицу при каждом запуске цикла.

```
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      int num = 1;
7      cout << "Start" << endl;
8      while (num < 6)
9      {
10         cout << "Number: " << num << endl;
11         num++; // num = num + 1
12     }
13     cout << "It has finished" << endl;
14     return (0);
15 }
```

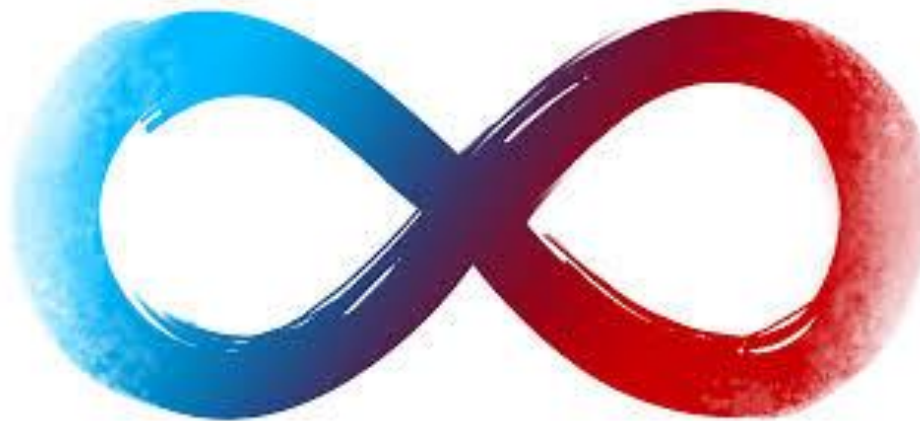
После 5-ой итерации `num` становится равным 6, и условие становится ложным, и цикл останавливается.

```
Start
Number: 1
Number: 2
Number: 3
Number: 4
Number: 5
It has finished
```



Тема: Условия ветвления. Цикл while.

**При отсутствии выражения
завершающего цикл, т.е.
устанавливающего условие
ложным, цикл будет
выполняться бесконечно!**



Программирование
на C++

Практическая часть

Условия ветвления.
Цикл while.

2 занятие



инжинириум®

МГТУ им. Н.Э. Баумана

2019

Тема: Условия ветвления. Цикл while.

Задание 1

«Угадай число»

Напишите программу, которая будет запрашивать положительное целое число (тип данных – int). Затем ваша программа должна вывести на экран: Если пользователь угадал число, то вы должны отправить поздравительное предложение в стандартный вывод

В противном случае – подсказать пользователю «hot» (близко ли он подобрался к загаданному вами числу) или “cold” (либо далеко от него)

Например, загаданное число 50, если пользователь вводит 70.

Программа выдаст «hot», если пользователь введёт 20 -> “cold”



Тема: Условия ветвления. Цикл while.

Задание 2

Ввести с клавиатуры (стандартный входной поток) номер дня недели, в зависимости от номера вывести название «понедельник», «вторник», «среда», «четверг», «пятница», «суббота», «воскресенье». Если номер больше 7 или меньше 1 – вывести «такого дня недели не существует».





Тема: Условия ветвления. Цикл while.

Задание 3

Напишите программу, которая выводит на экран чётный числа (начиная с 0) до 10 (включительно).

*

Начиная с 2



Тема: Условия ветвления. Цикл while.

Задание 4

Сделайте бесконечный цикл и запустите *.exe



Тема: Условия ветвления. Цикл while.

Задание 5

Напишите программу, которая выводит на экран нечётный числа (начиная с 11) до 1 (включительно).

