

# ПРОГРАММИРОВАНИЕ. Практика

## Тема 2. Основы C++



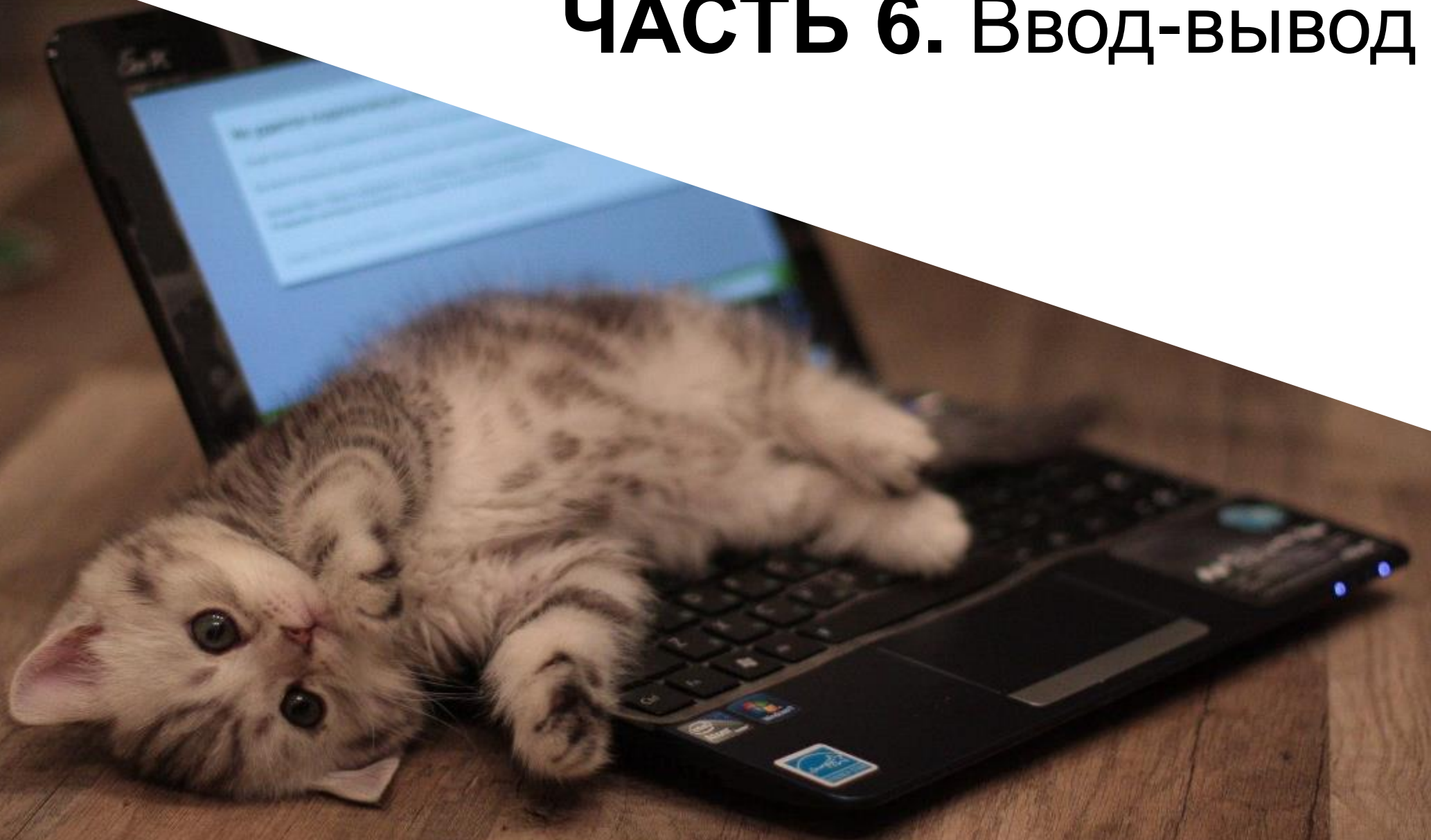
**ФЕДОТЕНКО**

**Мария Александровна**

 [ma.fedotenko@mpgu.su](mailto:ma.fedotenko@mpgu.su)

 [marusyafm](https://vk.com/marusyafm)

# ЧАСТЬ 6. Ввод-вывод



# Оператор вывода cout

Общий вид:

**cout << \*что вывести в консоль\*;**

```
1 ▾ #include <iostream>
2   using namespace std;
3
4 ▾ int main() {
5     cout << "Текст внутри кавычек";
6 }
```

✕ Output

Текст внутри кавычек

Process Finished.

>>>

# Оператор вывода cout



Для вывода с новой строки используется оператор **endl**

```
4 ▾ int main() {  
5     cout << "Текст внутри кавычек";  
6     cout << "Еще текст внутри кавычек";  
7 }
```

✕ Output

Текст внутри кавычекЕще текст внутри кавычек

Process Finished.

>>>

```
4 ▾ int main() {  
5     cout << "Текст внутри кавычек" << endl;  
6     cout << "Еще текст внутри кавычек";  
7 }
```

✕ Output

Текст внутри кавычек  
Еще текст внутри кавычек

Process Finished.

>>>

# Оператор вывода cout



Выводить можно как текст, так и значения переменных

```
4 ▾ int main() {  
5     cout << "Просто текст внутри кавычек" << endl;  
6  
7     int someNumber = 12;  
8     cout << someNumber << endl;  
9  
10    cout << "Значение переменной someNumber: " << someNumber;  
11 }
```

## ✕ Output

Просто текст внутри кавычек

12

Значение переменной someNumber: 12

Process Finished.

>>>



# Оператор ввода cin



Общий вид:

`cin >> *куда записать введенные данные*;`

Ввод осуществляется только с сохранением в переменную. То есть сначала эту переменную нужно объявить.

```
4 ▾ int main() {  
5     int someNumber;  
6     cin >> someNumber;  
7     // Посмотрим, что получилось:  
8     cout << someNumber;  
9 }
```

× Output

>>>125

125

Process Finished.

>>>

# Оператор ввода cin



Пользователь должен всегда понимать, что и для чего он должен вводить.

Поэтому перед вводом нужно выводить сопровождающий текст.

```
4 ▾ int main() {  
5     int someNumber, anoherNumber;  
6     cout << "Введите целое число: " << endl;  
7     cin >> someNumber;  
8     cout << "Введите второе целое число: " << endl;  
9     cin >> anoherNumber;  
10  
11     cout << "Сумма введенных чисел: " << someNumber+anoherNumber;  
12 }
```

## ✕ Output

Введите целое число:

>>>5

Введите второе целое число:

>>>27

Сумма введенных чисел: 32

Process Finished.

>>>

# Комбинирование ввода-вывода



```
4 ▾ int main() {  
5     string name;  
6     cout << "Введите ваше имя: " << endl;  
7     cin >> name;  
8     cout << "Здрвствуй, " << name << "! Теперь ты без 5 минут программист!";  
9 }
```

## ✕ Output

Введите ваше имя:

>>>Мария

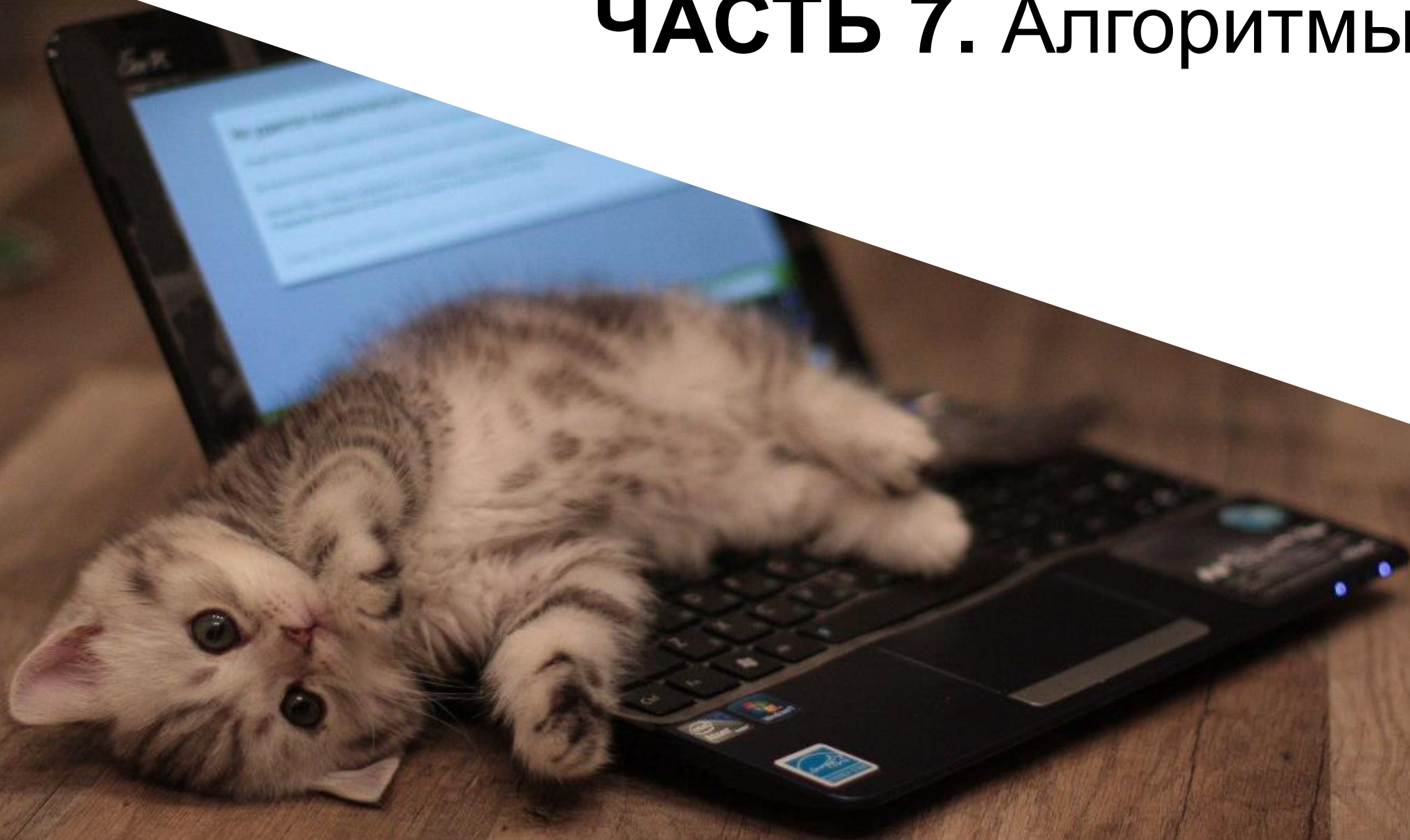
Здрвствуй, Мария! Теперь ты без 5 минут программист!

Process Finished.

>>> █

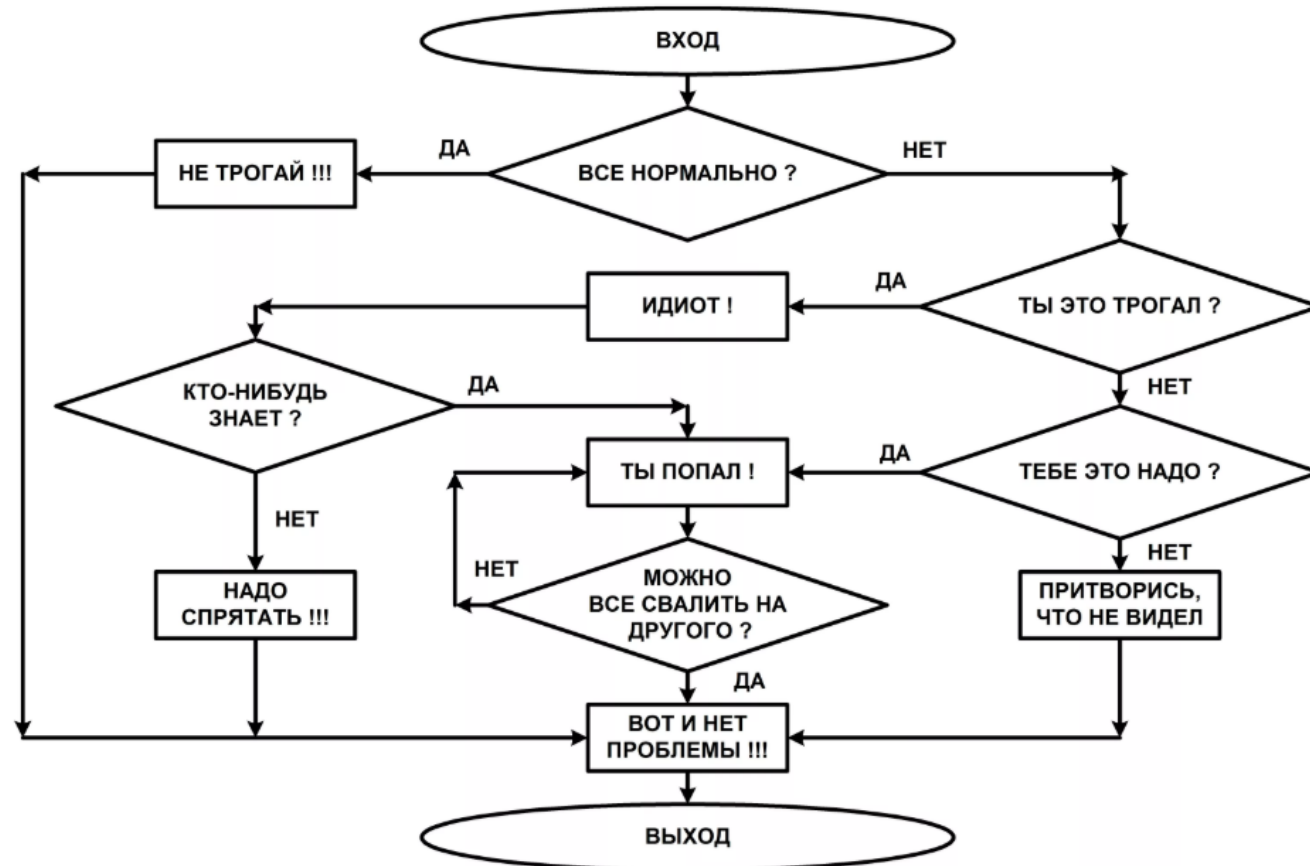


# ЧАСТЬ 7. Алгоритмы



# Что такое алгоритм?

**Алгоритм** – набор понятных и точных инструкций, описывающих порядок действий исполнителя для достижения результата за конечное число действий



# Виды алгоритмов

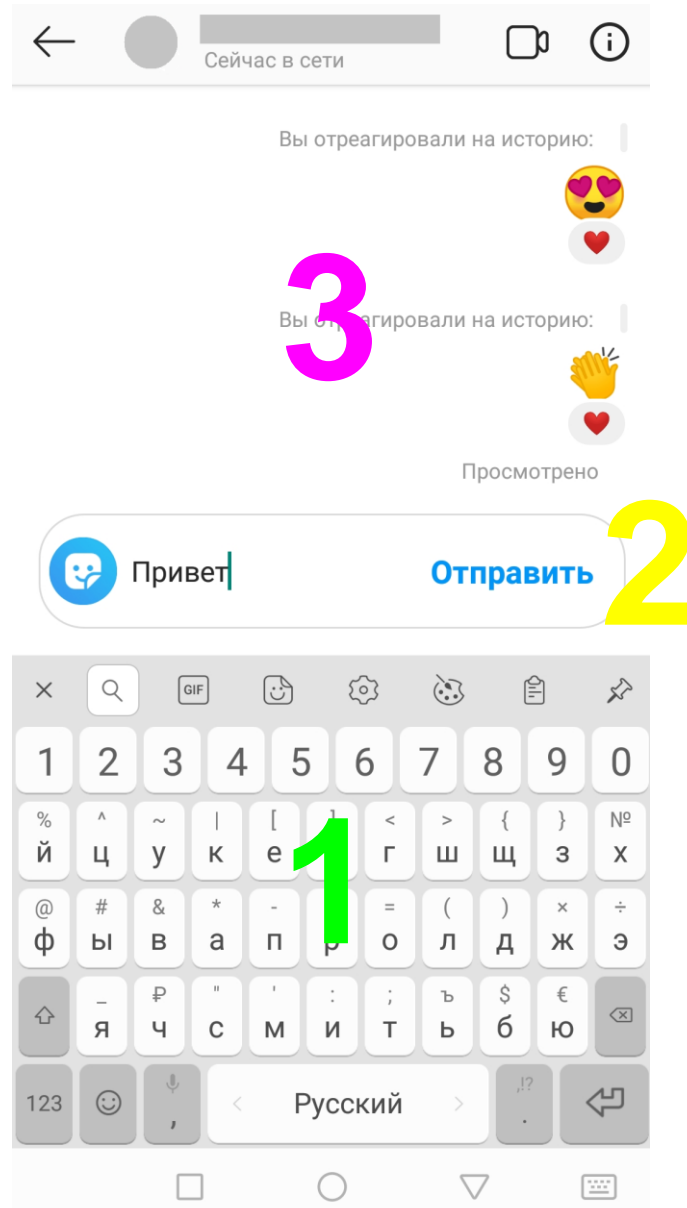
- Линейные
- Разветвляющиеся (алгоритмы с ветвлением)
- Циклические (циклы)

# Линейный алгоритм

**Линейный алгоритм** – алгоритм, все шаги которого выполняются строго последовательно, без пропусков и повторений.



# Линейный алгоритм



# Линейный алгоритм

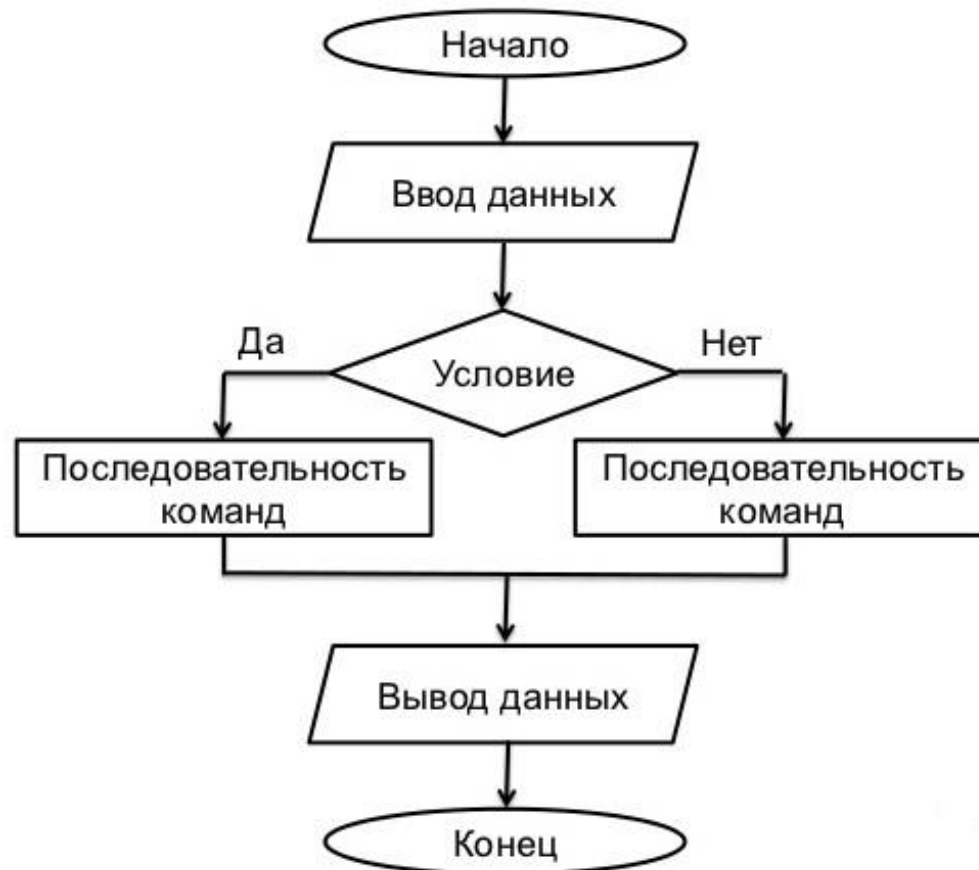


```
1 ▾ #include <iostream>
2   using namespace std;
3
4 ▾ int main() { // Тело главной функции
5     // Объявляем переменные
6     int sumOfNumbers = 0, firstNumber, secondNumber;
7     // Выводим текст в консоль
8     cout << "Введите первое число: ";
9     // Вводим число с клавиатуры
10    cin >> firstNumber;
11    // Повторяем для второго числа
12    cout << "Введите второе число: ";
13    cin >> secondNumber;
14    // Вычисляем сумму
15    sumOfNumbers = firstNumber + secondNumber;
16    // Выводим результат в консоль
17    cout << "Сумма чисел равна: " << sumOfNumbers;
18 }
```



# Разветвляющийся алгоритм

**Разветвляющийся алгоритм** – алгоритм, в котором в зависимости от некоторого условия выполняется либо одна, либо другая последовательность действий



# Разветвляющийся алгоритм

Для нарушения линейного порядка выполнения программы используются **операторы передачи управления**:

- Условный оператор
- Оператор множественного выбора

# Разветвляющийся алгоритм (1)

## УСЛОВНЫЙ ОПЕРАТОР

### Общий вид:

**ЕСЛИ** <условие> **ТО** <оператор 1> **ИНАЧЕ** <оператор 2>

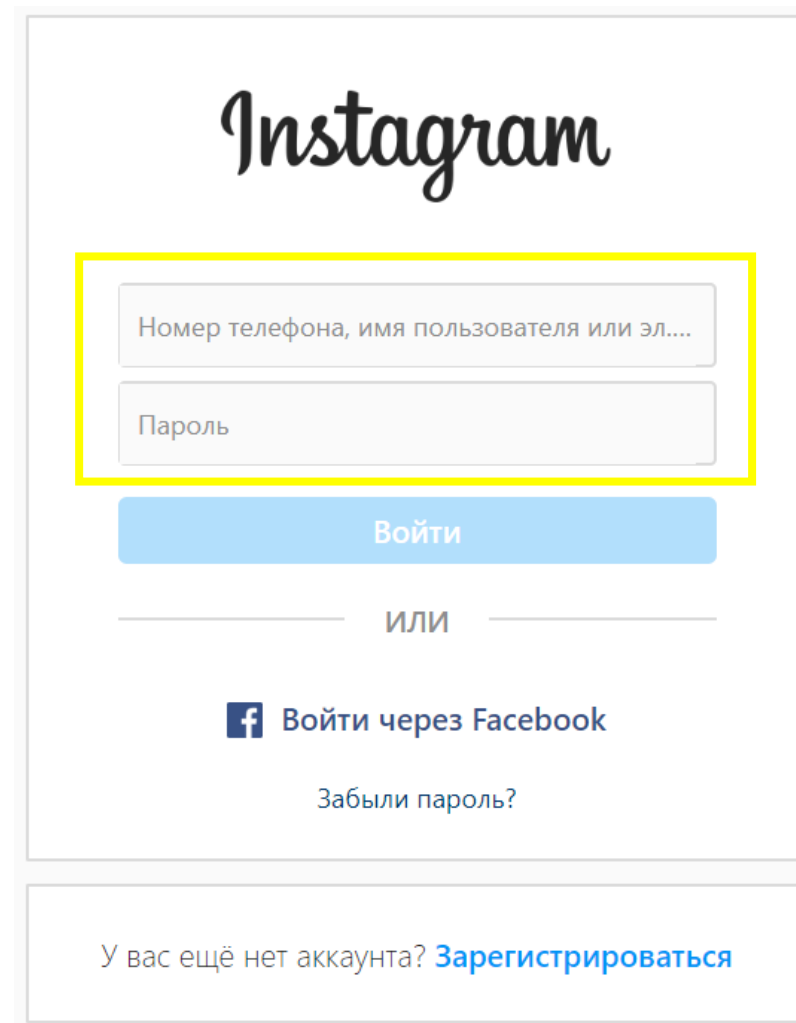
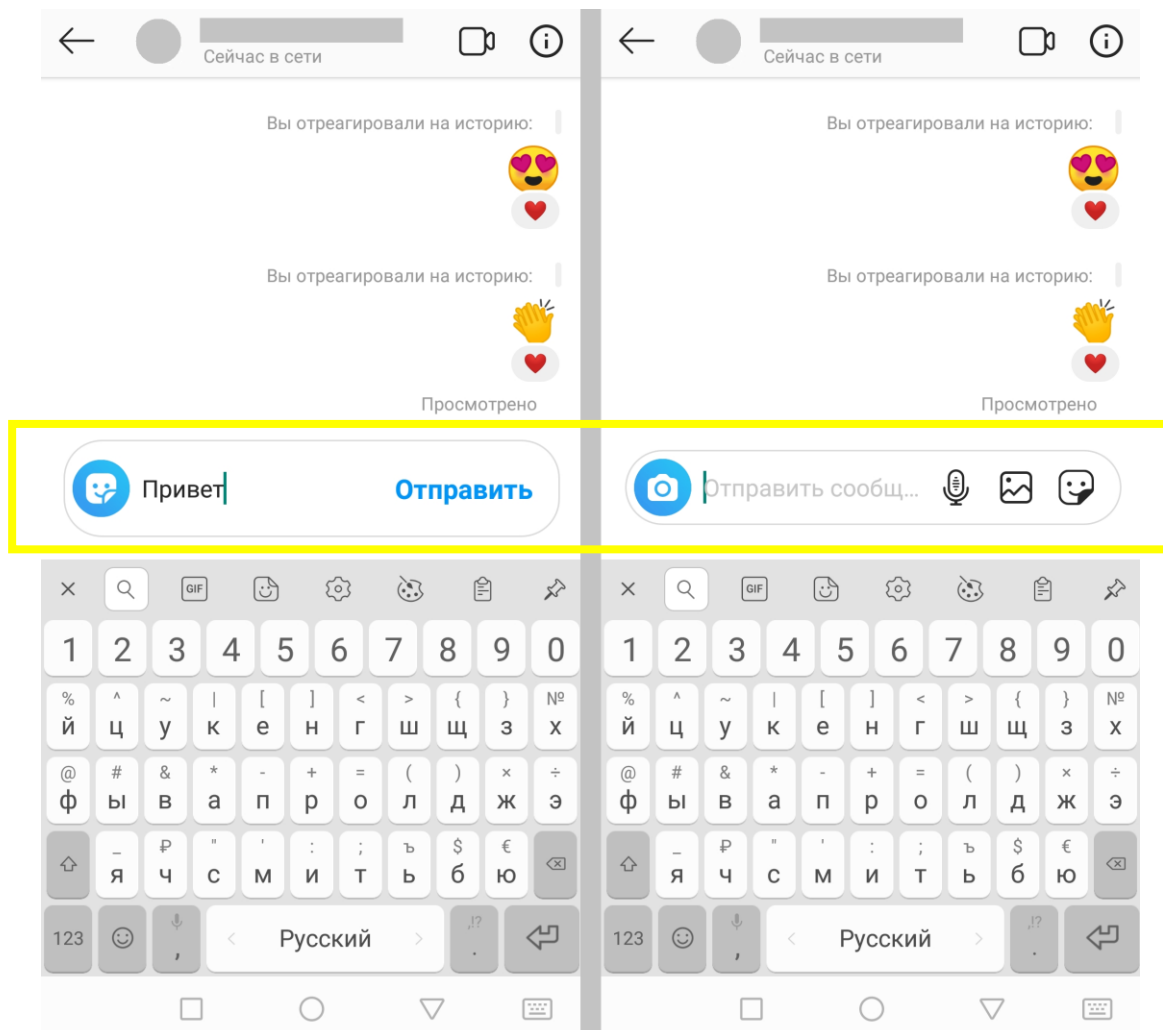
Неполное ветвление: **ЕСЛИ** <условие> **ТО** <оператор>

Условие обычно строится из операций сравнения

### В C++ имеет вид:

```
if (выполняется условие) {  
    оператор1;  
}  
else {  
    оператор2;  
}
```

# Разветвляющийся алгоритм (1)



# Разветвляющийся алгоритм (1)



```
1 ▾ #include <iostream>
2   using namespace std;
3
4 ▾ int main() {
5     int result = 0, firstNumber, secondNumber;
6     cout << "Введите первое число: ";
7     cin >> firstNumber;
8     cout << "Введите второе число: ";
9     cin >> secondNumber;
10    // Пошло ветвление
11 ▾    if (firstNumber == secondNumber) {
12        result = firstNumber * 2;
13        cout << "Числа равны" << endl;
14    }
15 ▾    else {
16        result = firstNumber + secondNumber;
17    }
18    // Выводим результат
19    cout << "Результат: " << result;
20 }
```

# Разветвляющийся алгоритм (2)

## ОПЕРАТОР МНОЖЕСТВЕННОГО ВЫБОРА

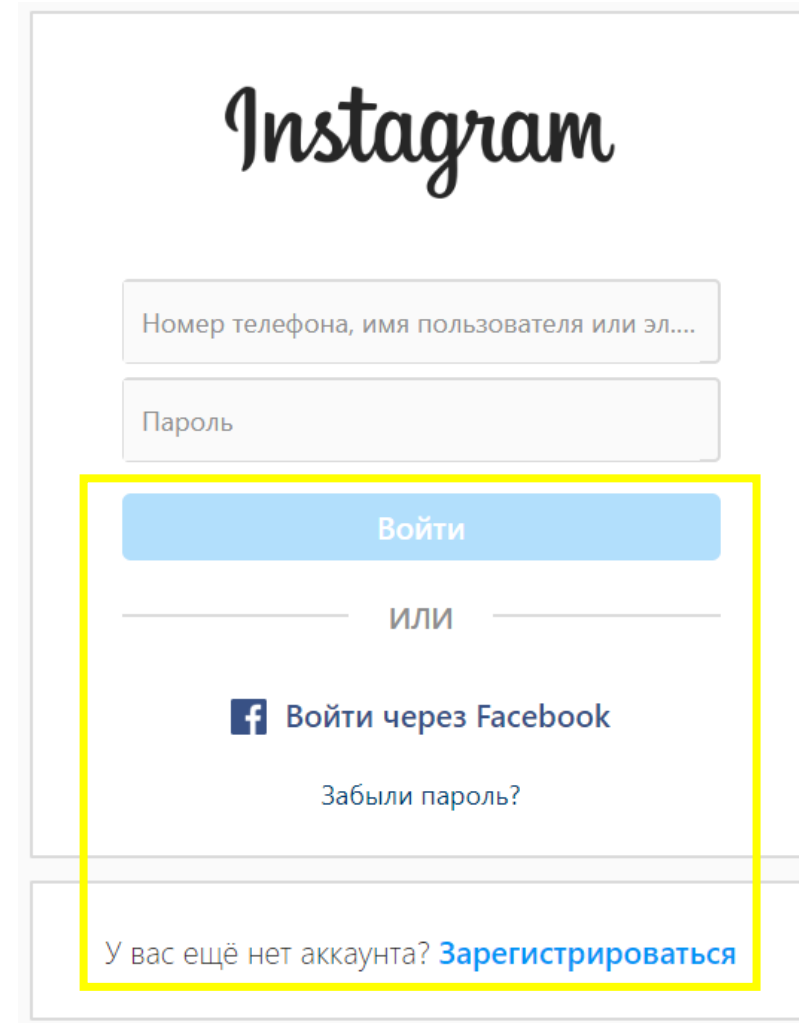
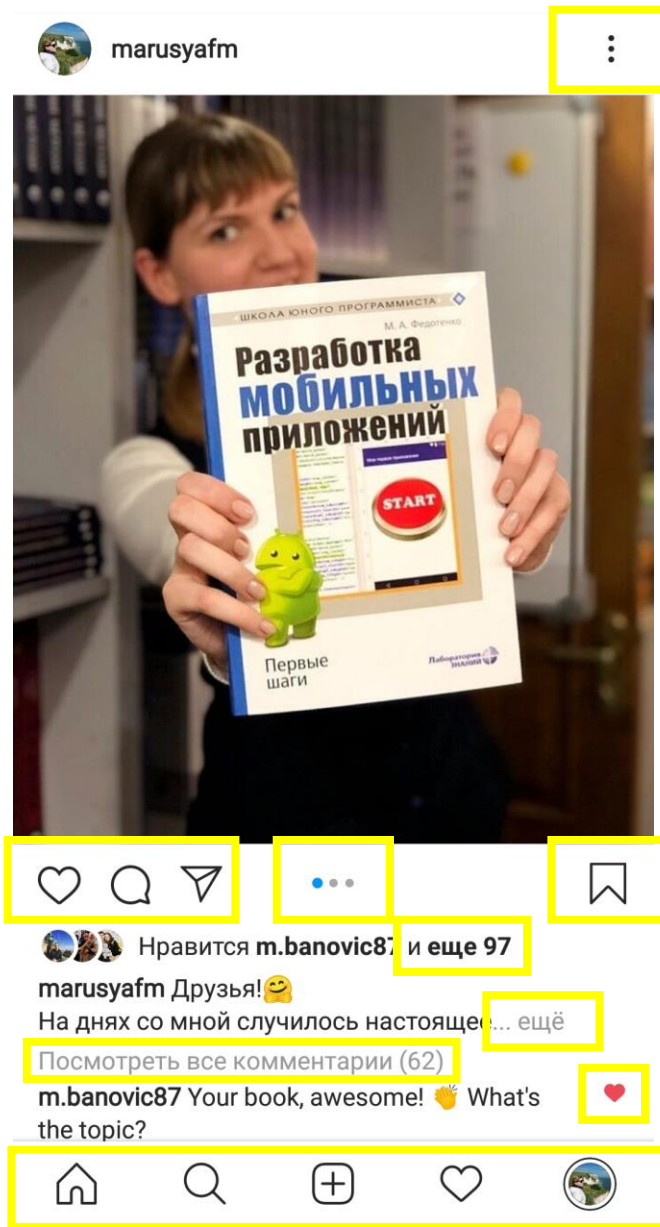
Используется как альтернатива `if` при достаточно большом (известном!) количестве вариантов

### В C++ имеет вид:

```
switch(ВыражениеДляСравнения) {  
    case <значение1>:  
        <действия1>;  
        break;  
    case <значение2>:  
        <действия2>;  
        break;  
    .....;  
    default: <действия4>;  
        break;  
}
```



# Разветвляющийся алгоритм (2)



# Разветвляющийся алгоритм (2)

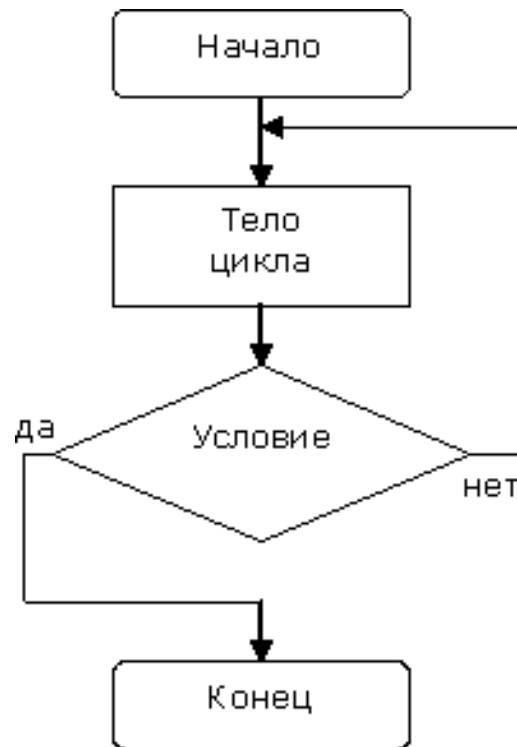


```
1 ▾ #include <iostream>
2   using namespace std;
3
4 ▾ int main() {
5     int result = 0, someNumber, operation;
6     cout << "Введите число: ";
7     cin >> someNumber;
8     cout << "Выберите операцию (1: *2, 2: -2, 3: +2): ";
9     cin >> operation;
10    // Пошло ветвление
11 ▾    switch (operation) {
12        case 1: result = someNumber * 2;
13            break;
14        case 2: result = someNumber - 2;
15            break;
16        case 3: result = someNumber + 2;
17            break;
18        default: cout << "Попробуйте снова";
19            break;
20    }
21    // Выводим результат
22    cout << "Результат: " << result;
23 }
```

# Цикл

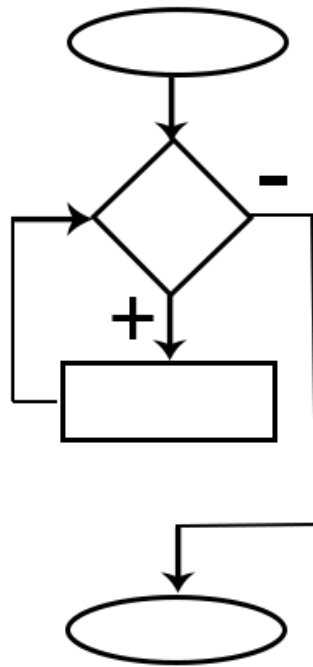
**Цикл** – алгоритм, блоки которого выполняются до тех пор, пока не будет выполнено определенное условие.

**Тело цикла** – последовательность действий, многократно выполняемых в процессе выполнения цикла.

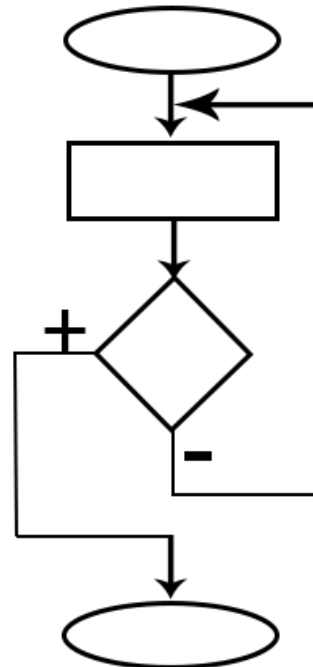


# Виды циклов

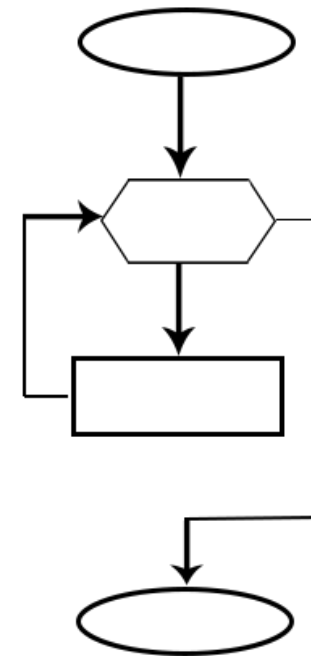
Цикл с  
предусловием



Цикл с  
постусловием



Цикл с  
параметром



# Цикл с предусловием

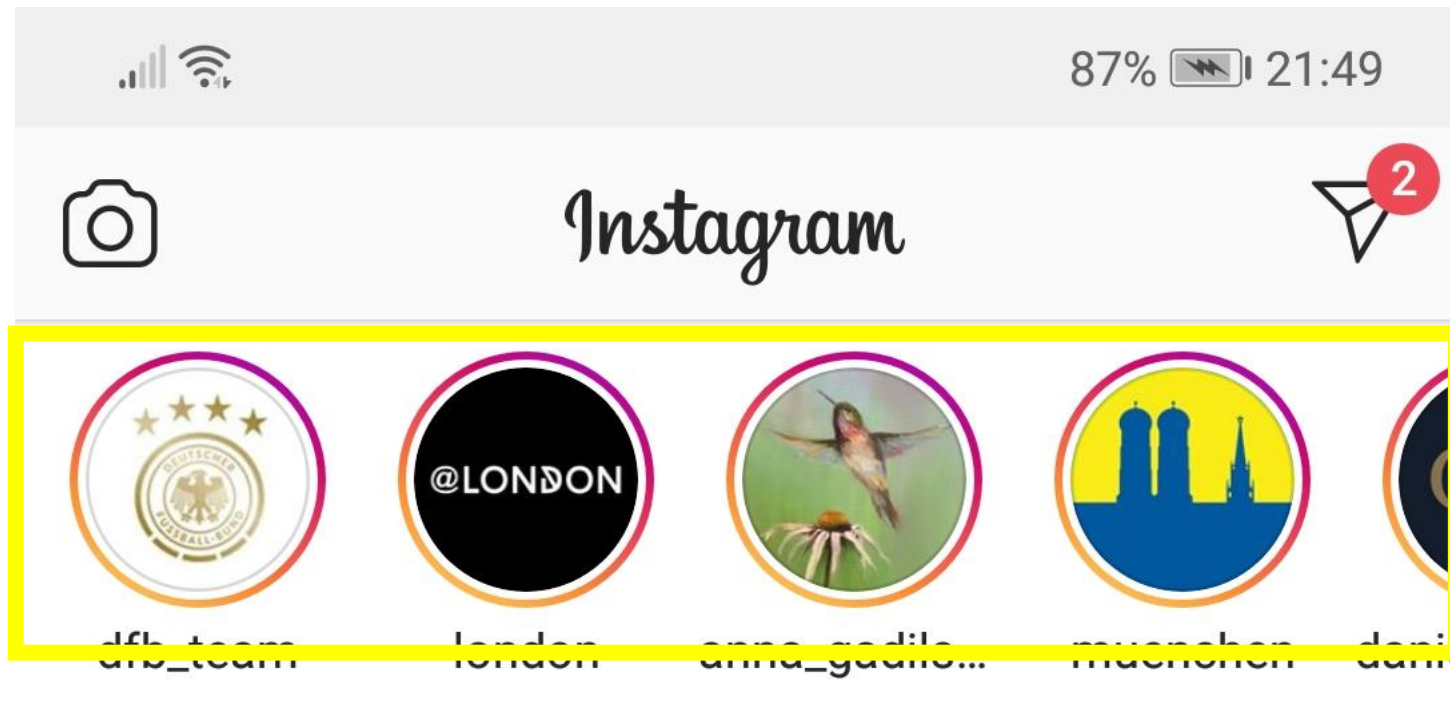
**While** — это цикл, в котором условие стоит перед телом цикла

Тело цикла выполняется только **пока** условие истинно (**true**); как только условие становится ложно (**false**) - выполнение цикла прекращается

**В C++ имеет вид:**

```
while(условие) {  
    тело цикла;  
}
```

# Цикл с предусловием





# Цикл с постусловием

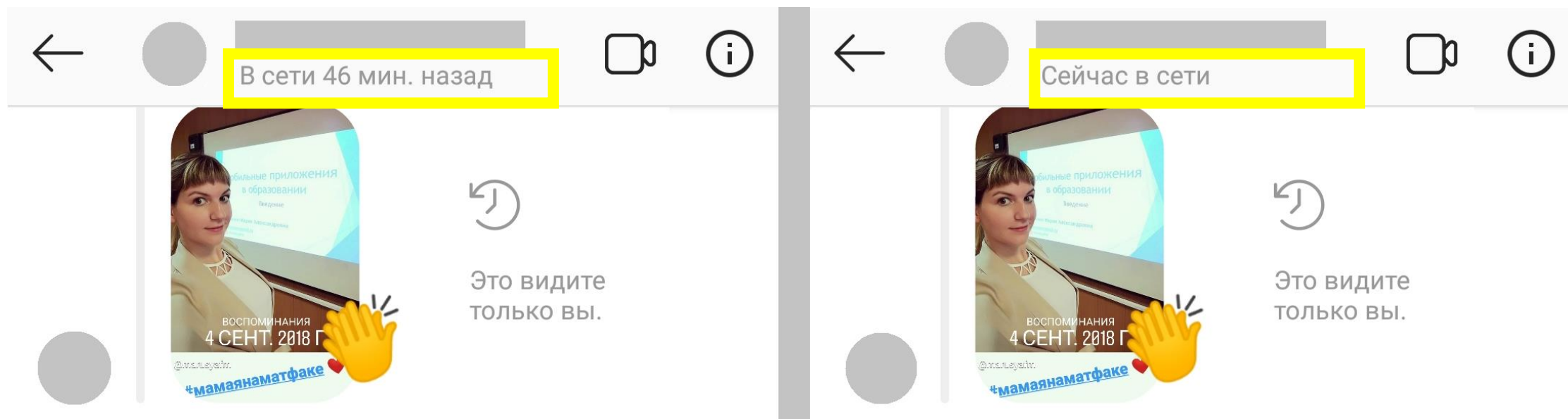
**Repeat (do...while)** — это цикл, в котором условие стоит после тела цикла

Тело цикла выполняется только до тех пор, пока условие ложно (**false**); как только условие становится истинно (**true**) - выполнение цикла прекращается

**В C++ имеет вид:**

```
do {  
    тело цикла;  
} while(условие-логическое выражение);
```

# Цикл с постусловием



# Цикл с параметром

**For** — это цикл, в котором тело цикла выполняется заданное количество раз (итераций)

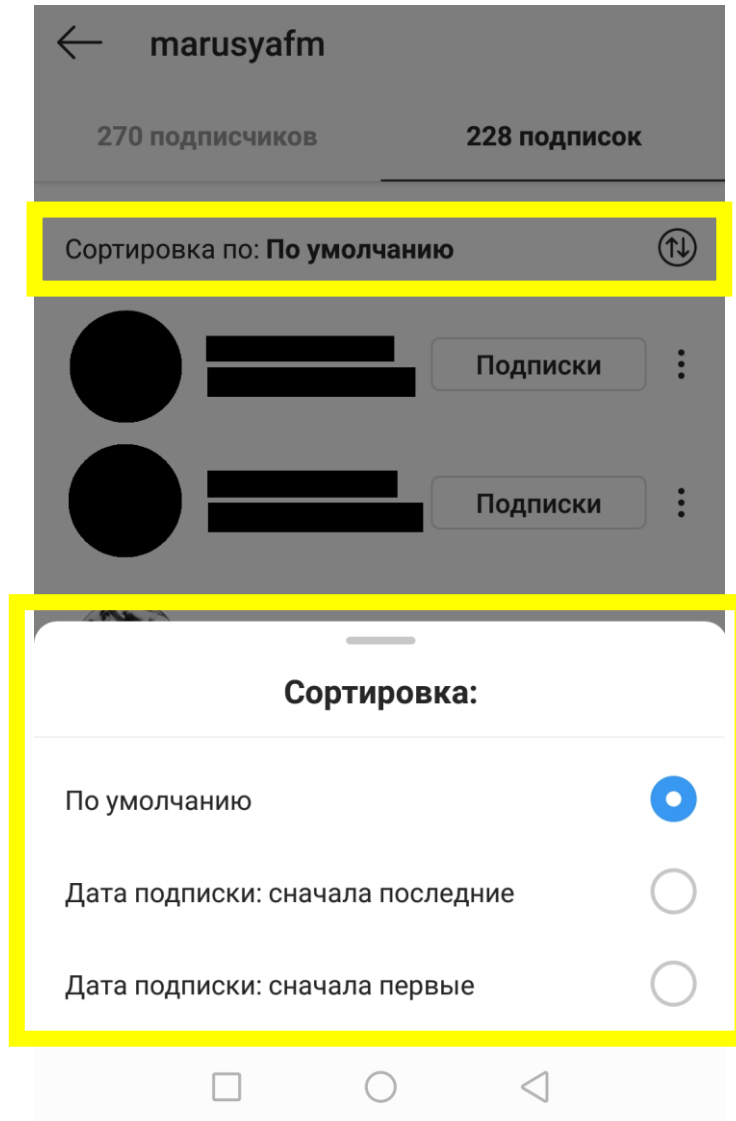
**Итерация** – один проход цикла

**Счетчик цикла** — это переменная, в которой хранится количество проходов данного цикла

**В C++ имеет вид:**

```
for (действие до начала; условие продолжения; шаг) {  
    тело цикла;  
}
```

# Цикл с параметром



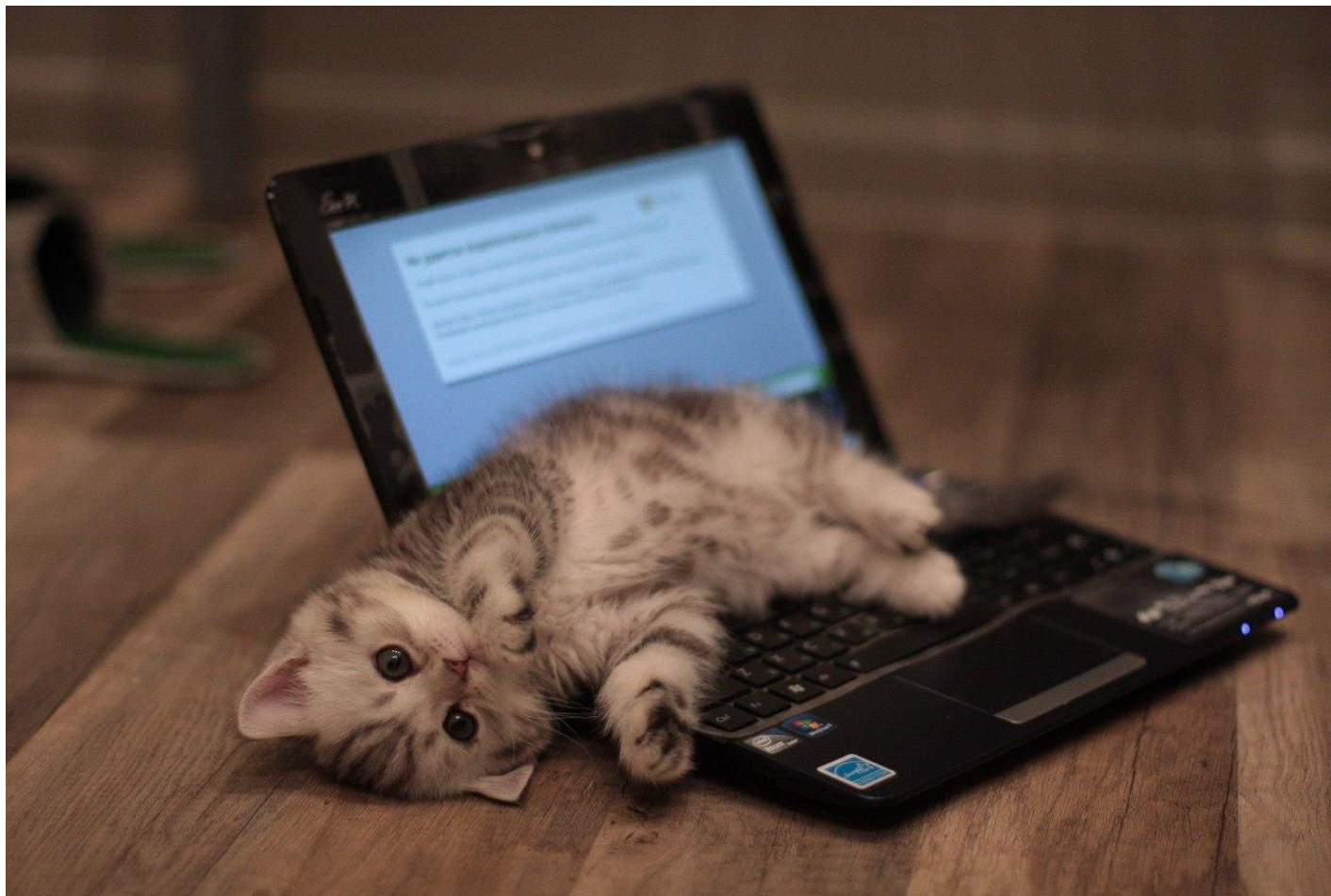
# Циклы



**Задача:** посчитать сумму чисел от 0 до 10


С предусловием	С постусловием	С параметром
<pre>#include &lt;iostream&gt; using namespace std;  int main() {      int i = 0;     int summa = 0;      while (i &lt; 10) {         summa += i;         i++;     }      cout &lt;&lt; "Сумма чисел от 0 до 10 = " &lt;&lt;     summa;  }</pre>	<pre>#include &lt;iostream&gt; using namespace std;  int main() {      int i = 0;     int summa = 0;      do {         summa += i;         i++;     } while (i &lt; 10);      cout &lt;&lt; "Сумма чисел от 0 до 10 = " &lt;&lt;     summa;  }</pre>	<pre>#include &lt;iostream&gt; using namespace std;  int main() {      int summa = 0;      for (int i = 0; i &lt;= 10; i++) {         summa += i;     }      cout &lt;&lt; "Сумма чисел от 0 до 10 = " &lt;&lt;     summa;  }</pre>

# Благодарю за внимание!





# Задания

1. Найти в презентации все слайды, отмеченные значком  (их 9).
2. Переписать, скомпилировать и запустить программы, код которых представлен на этих слайдах.
3. Видоизменить код (имена переменных + типы данных + операторы + добавить текст комментариев) для получения собственных программ.

# Онлайн компиляторы

Задания можно выполнять в онлайн компиляторах.

Предлагаемые онлайн компиляторы:

1. Repl IT – <https://repl.it/languages/cpp>
2. Dcoder – <https://code.dcoder.tech/>
3. OnlineGDB – [https://www.onlinegdb.com/online\\_c++\\_compiler](https://www.onlinegdb.com/online_c++_compiler)
4. Jdoodle – <https://www.jdoodle.com/online-compiler-c++/>
5. IDEone – <https://ideone.com/l/cpp>
6. Programiz – <https://www.programiz.com/cpp-programming/online-compiler/>