

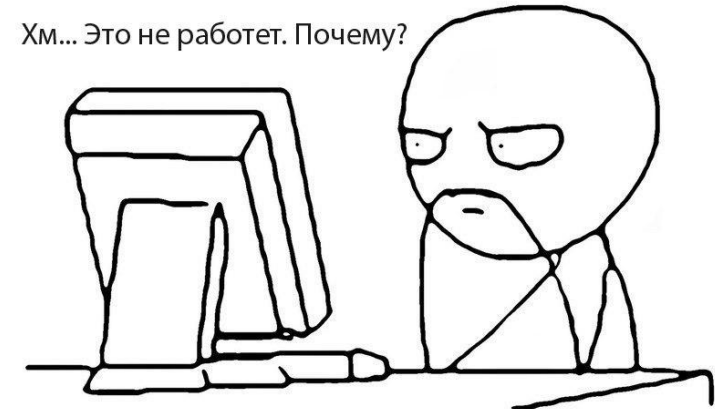
ОСНОВЫ ПРОГРАММИРОВАНИЯ

Занятие первое

Программирование это

ожидание:

ь:



Программирование это

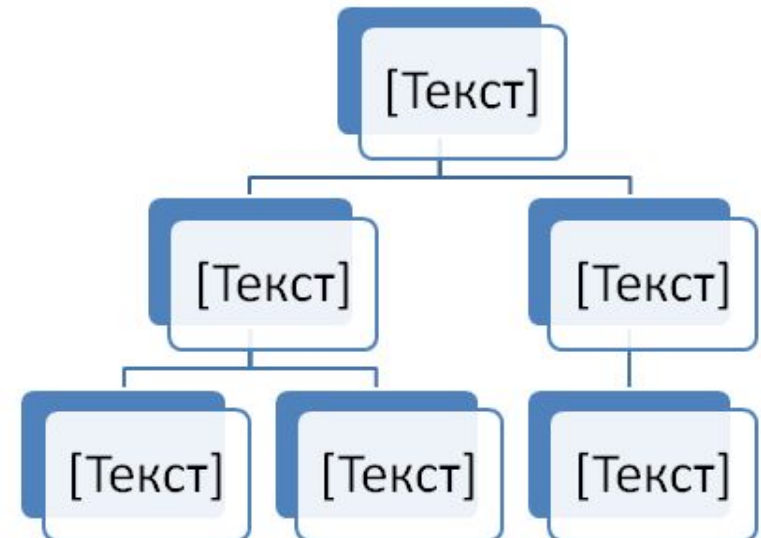
- Процесс преобразования содержимого команд в дальнейшем преобразующихся в машинный код.
- По выражению одного из основателей языков программирования Никлауса Вирта «Программы = алгоритмы + структуры данных».



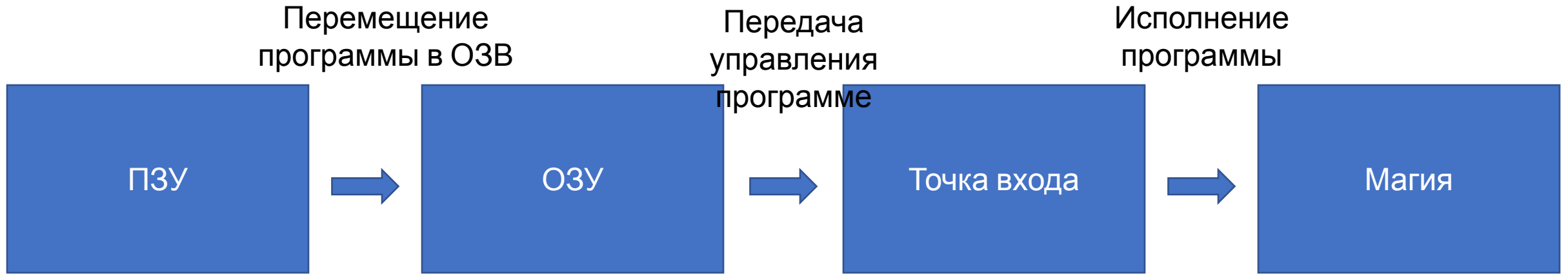
Виды языков программирования

- Языки низкого и высокого уровня
- Компилируемые и интерпретируемые
- Строго и слабо типизированные
- ...

Типы языков программирования



Запуск программы



Точка входа

- Адрес в оперативной памяти, с которого начинается выполнение программы. Другими словами — адрес, по которому хранится первая команда программы.

Язык Си

- компилируемый статически типизированный язык программирования общего назначения.



Точка входа в Си

- Точкой входа в языке C/C++ является функция main

```
int main() {  
    return 0;  
}
```

- Функция - фрагмент программного кода (подпрограмма), к которому можно обратиться из другого места программы.

Вывод сообщения на экран

- `printf(format, arg1, arg2, arg3, ..., argN);`

- Примеры:

- `printf("Hello world");` // Hello world
- `printf("2 + 2 = %d", 4);` // 2 + 2 = 4
- `printf("Hi, %s", "Ivan");` // Hi, Ivan

%c	символ
%d	целое десятичное число
%i	целое десятичное число
%e	десятичное число в виде x.xx e+xx
%E	десятичное число в виде x.xx E+xx
%f	десятичное число с плавающей запятой xx.xxxx
%F	десятичное число с плавающей запятой xx.xxxx
%g	%f или %e, что короче
%G	%F или %E, что короче
%o	восьмеричное число
%s	строка символов
%u	беззнаковое десятичное число
%x	шестнадцатеричное число
%X	шестнадцатеричное число
%%	символ %
%p	указатель

Спецификаторы формата

A man with long hair and glasses is sitting in a computer room, holding a laptop. He is wearing a patterned t-shirt and a jacket. The room is dimly lit with blue and green light. There are several computer monitors and keyboards on desks in the background. The text "I am a programmer" is overlaid on the image in white.

I am a programmer

Типы данных C++

Тип	байт	Диапазон принимаемых значений
целочисленный (логический) тип данных		
bool	1	0 / 255 (0 – false, 1-255 – true)
целочисленный (символьный) тип данных		
char	1	0 / 255
целочисленные типы данных		
short int	2	-32 768 / 32 767
unsigned short int	2	0 / 65 535
int	4	-2 147 483 648 / 2 147 483 647
unsigned int	4	0 / 4 294 967 295
long int	4	-2 147 483 648 / 2 147 483 647
unsigned long int	4	0 / 4 294 967 295
типы данных с плавающей точкой		
float	4	-2 147 483 648.0 / 2 147 483 647.0
long float	8	-9 223 372 036 854 775 808 .0 / 9 223 372 036 854 775 807.0
double	8	-9 223 372 036 854 775 808 .0 / 9 223 372 036 854 775 807.0

Операции с типами данных.

- Бинарные
 - Сложение +
 - Вычитание –
 - Умножение *
 - Деление /
 - Присваивание =
- Унарные
 - Инкрементирование ++
 - Декрементирование --
 - Изменение знака –

P.S. Остальное тут
<https://prog-cpp.ru/c-operation/>

Объявление переменной.

- Переменная – именованная область памяти используемая для хранения данных.
- Шаблон объявления: “<Тип> <Имя>;”
- Пример:
 - int a;
 - char b;
 - double d;

Пример использования операций

```
int a;
```

```
int a = 2;
```

```
int a = 2;
```

```
int a = 2;
```

```
int b;
```

```
int b = 3;
```

```
a = a + 1;
```

```
a++;
```

```
int c;
```

```
int c = a + b;
```

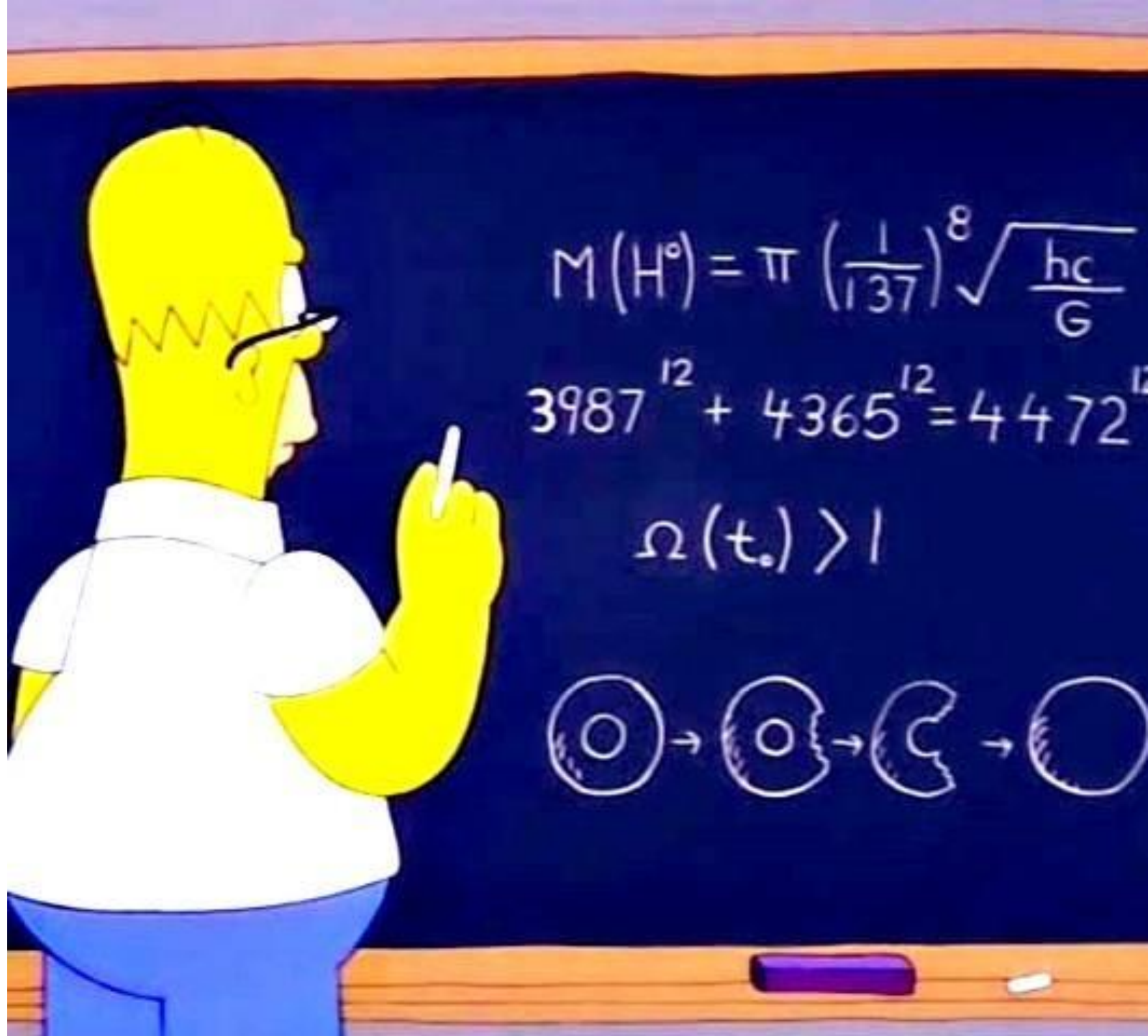
```
a = 2;
```

```
b = 3;
```

```
c = a + b;
```

Немного
посчитаем.

- $x + y = ?$
- $x^2 + y^2 = ?$
- $x / y = ?$





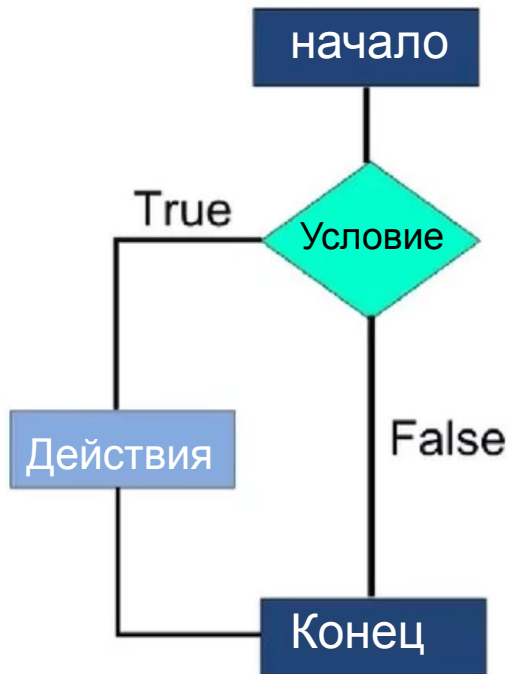
I can program

Давайте посчитаем

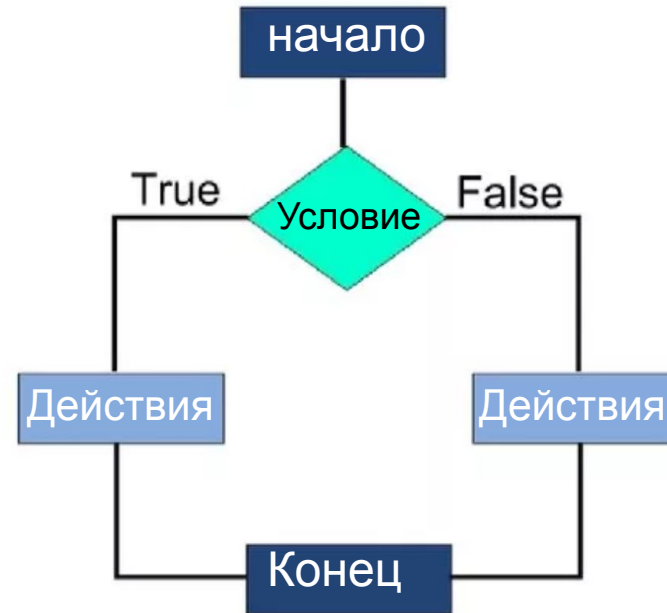
- Корень квадратного уравнения



Конструкция – **if**



Конструкция **if - else**



Условные конструкции (if-else)

if (выражение_условия)

 Действие;

else

 Действие;

- Выполняет одну инструкцию/блок кода после условия.

- P.S. Другие условные операторы
<https://metanit.com/cpp/c/2.11.php>

Булева алгебра

- Непустое множество A с двумя бинарными операциями \wedge (конъюнкции), \vee (дизъюнкции), одной унарной операцией \neg (отрицания) и двумя элементами: 0 (Ложь) и 1 (Истина).

A	B	Отрицание Инверсия (НЕ) $\neg A$	Конъюнкция Логическое умножение (И) $A \wedge B$	Дизъюнкция Логическое сложение (ИЛИ) $A \vee B$
0	0	1	0	0
0	1	1	0	1
1	0	0	0	1
1	1	0	1	1

Логика в программировании

- && - Конъюнкция (логическое И)
- || - Дизъюнкция (логическое ИЛИ)
- ! – Отрицание (логическое НЕ)

Потренируемся.

A	B	C	$A \ \&\& \ B \ \ C$	$!A \ \&\& \ !(B \ \ !C)$	$A \ \ (!B \ \&\& \ !A) \ \ (C \ \&\& \ A)$	$(A \ \ B) \ \&\& \ (!A \ \&\& \ !B) \ \ C$
0	0	0				
0	0	1				
0	1	0				
0	1	1				
1	0	0				
1	0	1				
1	1	0				
1	1	1				

Где хранить?

- `bool` – переменная размером 1 байт. Хранит в себе значения:
 - 0 – `false`
 - 1 – `true`
- **`bool`** `a = true; !a = false;`
- **`bool`** `b = false; !b = true;`
- `a && b – false; a || b – true;`



Нежданчик

- В языке Си нет bool
- Есть unsigned char

```
int main() {  
    bool a;  
    return 0;  
}
```


Операторы сравнения

Равенство	==	a == b
Неравенство	!=	a != b
Больше	>	a > b
Меньше	<	a < b
Больше или равно	>=	a >= b
Меньше или равно	<=	a <= b

```
int a = 2; int b = 3;
```

a == b	false
a != b	true
a < b	true
a <= b	true
a > b	false
a >= b	false

Потренируемся

```
int a = 7; int b = 5; int c = 8; int d = 7;
```

<code>a == b</code>	false
<code>a <= d</code>	true
<code>b > c</code>	false
<code>a != d</code>	false
<code>b != c</code>	true
<code>b >= c</code>	false
<code>c + d < a + b</code>	false
<code>a - b == d - b</code>	true

Немного
практики.

Учим
чётные числа
2, 4, 6, 8, 10.





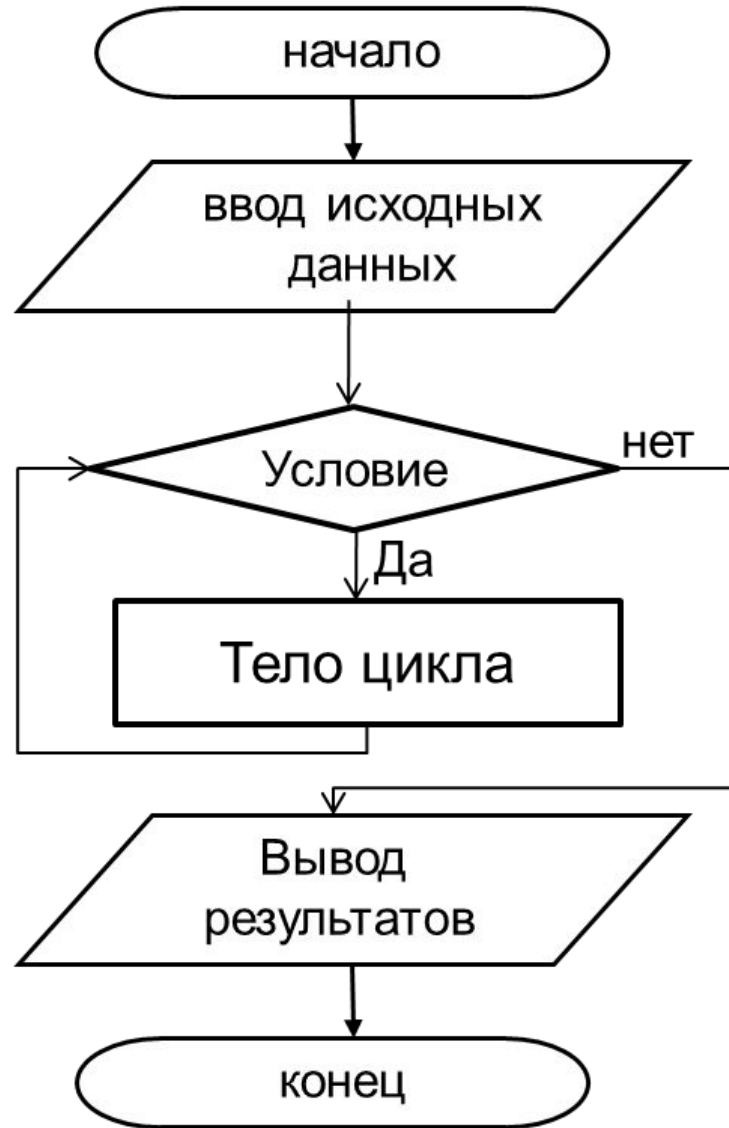
Вопросы?



**ТЕБЕ НЕ НУЖНО УЧИТЬ ТАБЛИЦУ
УМНОЖЕНИЯ**

ЕСЛИ ТЫ ЗАПРОГРАММИРУЕШЬ ЕЕ

risovach.ru



Циклы

- Разновидность управляющей конструкции в высокоуровневых языках программирования, предназначенная для организации многократного исполнения набора инструкций.

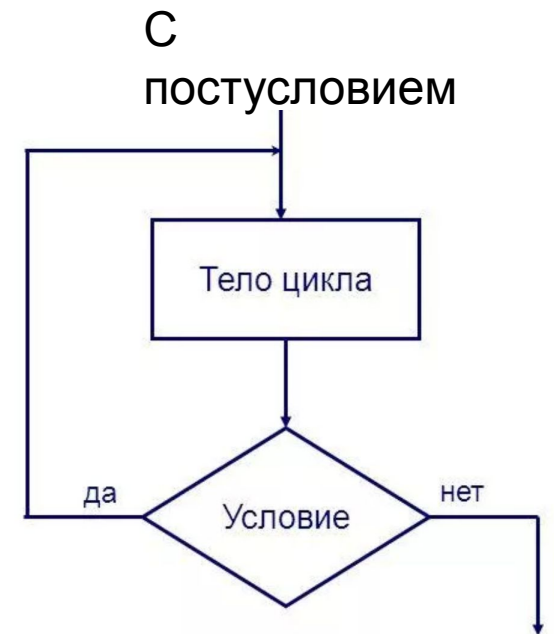
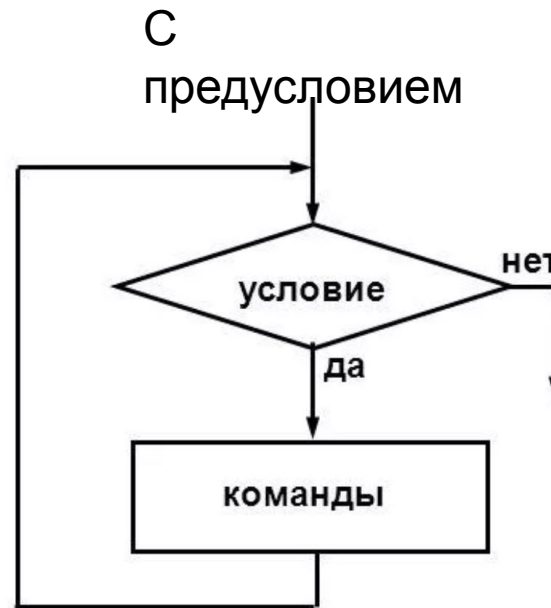
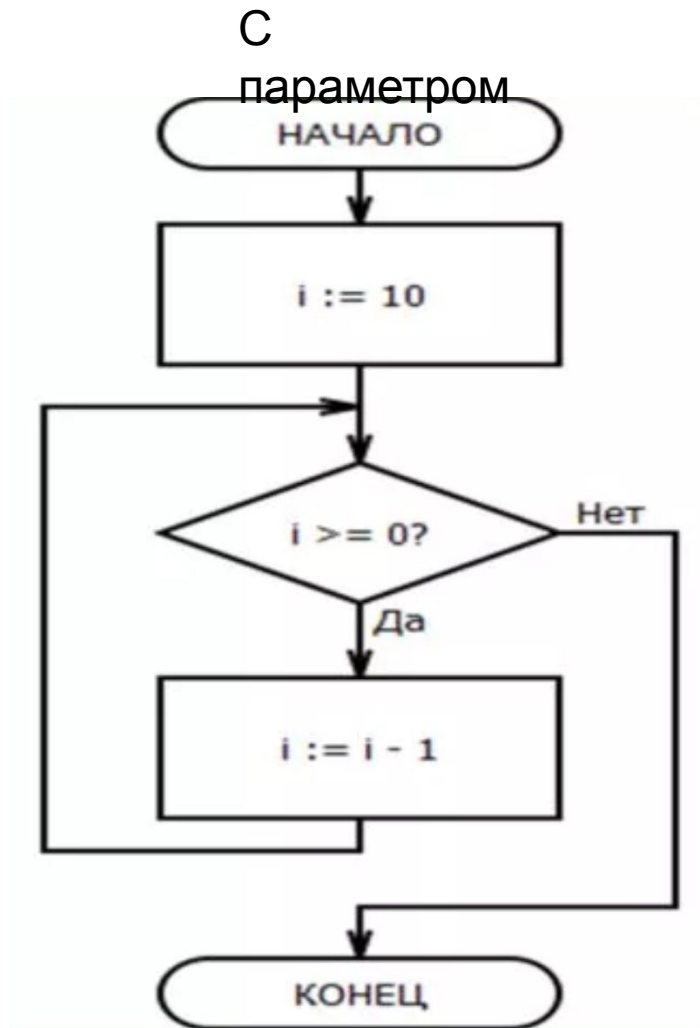
Виды ЦИКЛОВ

С предусловием — цикл, который выполняется, пока истинно некоторое условие, указанное перед его началом. Это условие проверяется до выполнения тела цикла, поэтому тело может быть не выполнено ни разу

С постусловием — цикл, в котором условие проверяется после выполнения тела цикла. Отсюда следует, что тело всегда выполняется хотя бы один раз.

С параметром – отличается от цикла с предусловием тем что имеет параметр изменяющийся во время выполнения.

Где кто?



Как выглядит в Си

```
do {  
    // Действия  
} while(Условие);
```

```
while(Условие) {  
    // Действия  
};
```

```
for (Объявление; Условие;  
Изменение;) {  
    // Действия  
};
```

```
int a = 0;
while(a < 10) {
    printf("%d", a);
    a++;
}
```

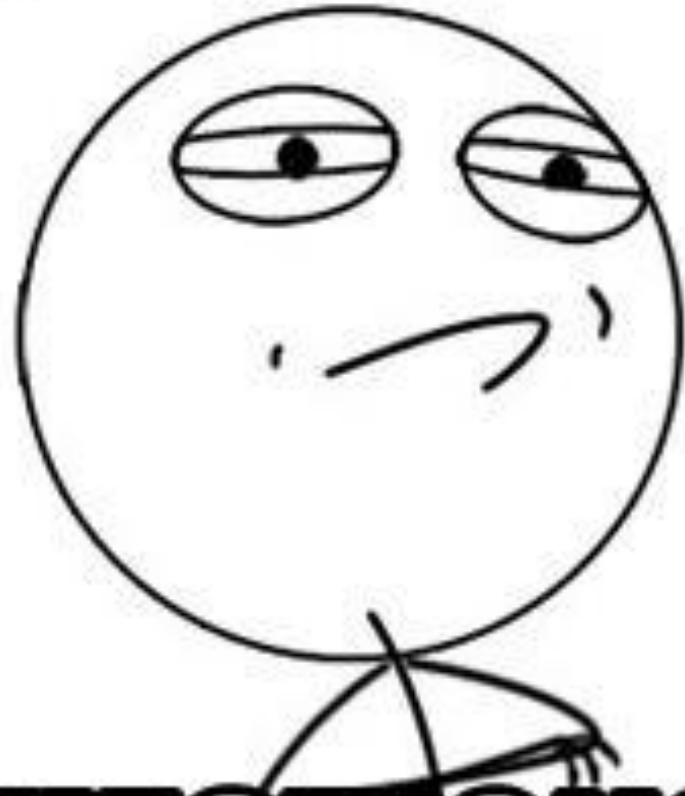
```
for (int i = 0; i < 10; i++) {
    printf("%d", i);
}
```

```
int a = 0;
do {
    printf("%d", a);
    a++;
} while (a < 10);
```

Пример

Напечатать все цифры.

END OF PRESENTATION



QUESTIONS?