

# Программирование (АлгЯзык)

## **§ 20. Программирование циклических алгоритмов**

# Зачем нужен цикл?

Задача. Вывести 5 раз «Привет!».

```
вывод 'Привет' , нс  
вывод 'Привет' , нс  
вывод 'Привет' , нс  
вывод 'Привет' , нс  
вывод 'Привет' , нс
```



А если 5000?

Цикл «N раз»:

```
нц 5 раз  
    вывод 'Привет' , нс  
кц
```

# Как работает цикл?

**!** Нужно запоминать, сколько раз цикл уже выполнен!

переменная-счётчик

```
счётчик := 0
нц пока счётчик < 5
  вывод 'Привет', нс
  счётчик := счётчик + 1
кц
```

ещё не делали

сделали ещё раз

# Как работает цикл?

---

Идея: запоминать, сколько шагов осталось.

счётчик := 5

нц пока счётчик > 0

    вывод 'Привет', нс

    счётчик := счётчик - 1

кц

# Цикл с предусловием

- условие проверяется при входе в цикл
- как только условие становится ложным, работа цикла заканчивается
- если условие ложно в самом начале, цикл не выполняется **ни разу**

**нц пока** *условие*

...

**кц**

тело цикла



Если условие никогда не станет ложно?

**нц пока** *да*

...

**кц**

бесконечный цикл  
(зацикливание)

# Сумма цифр числа

---

*Задача.* Вычислить сумму цифр введённого числа.

$$123 \rightarrow 1 + 2 + 3 = 6$$

Выделить последнюю цифру числа в переменной  $N$ :

$$d := \text{mod}(N, 10) \quad 123 \rightarrow 3$$

Отбросить последнюю цифру числа в переменной  $N$ :

$$N := \text{div}(N, 10) \quad 123 \rightarrow 12$$

Добавить к переменной  $sum$  значение переменной  $d$ :

$$sum := sum + d \quad \begin{aligned} sum = 6 &\rightarrow 6 + 4 = 10 \\ d &= 4 \end{aligned}$$

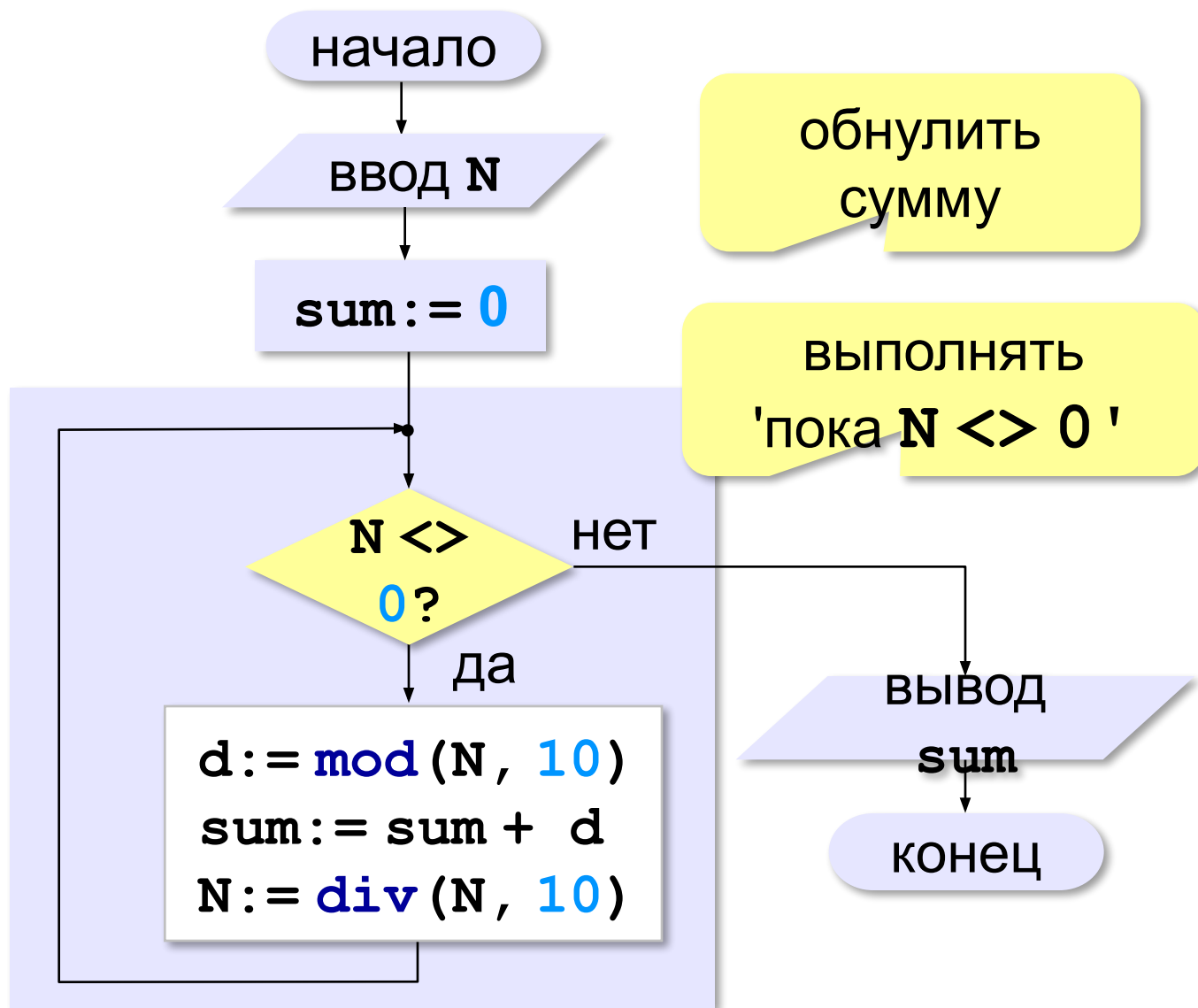
# Сумма цифр числа

- выделяем последнюю цифру числа (**mod**)
- увеличиваем сумму на значение цифры (**sum:=sum+d**)
- отсекаем последнюю цифру числа (**div**)

N	d	sum
123		0

начальные значения

# Сумма цифр числа





# Сумма цифр числа

алг Сумма цифр

нач

цел N, d, sum, N1

Вывод 'Введите целое число', НС

Ввод N ; N1 := N

sum := 0

нц пока N <> 0

d := mod(N, 10)

sum := sum + d

N := div(N, 10)

кц

Вывод 'Сумма цифр числа', N1, 'равна', sum

кон



Что плохо?

# Задачи

---

**«А»:** Напишите программу, которая получает с клавиатуры количество повторений и выводит столько же раз какое-нибудь сообщение.

**Пример:**

Сколько раз повторить? **3**

Привет!

Привет!

Привет!

**«В»:** Напишите программу, которая получает с клавиатуры натуральное число и определяет количество цифр в этом числе.

**Пример:**

Введите число? **12345**

Цифр в числе: 5

# Задачи

---

**«С»:** Напишите программу, которая получает с клавиатуры натуральное число и находит наибольшую цифру в его десятичной записи.

**Пример:**

Введите число: **311**

Наибольшая цифра: 3

**«D»:** Напишите программу, которая получает с клавиатуры натуральное число и определяет, есть ли в его десятичной записи одинаковые цифры, стоящие рядом.

**Пример:**

Введите число: **553**

Ответ: да.

Введите число: **535**

Ответ: нет.

# Алгоритм Евклида

**Задача.** Найти наибольший общий делитель (НОД) двух натуральных чисел.

Заменяем большее из двух чисел **разностью** большего и меньшего до тех пор, пока они не станут равны. Это и есть НОД.

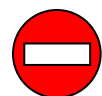
$$\begin{aligned}\text{НОД}(a, b) &= \text{НОД}(a - b, b) \\ &= \text{НОД}(a, b - a)\end{aligned}$$



Евклид  
(365-300 до. н. э.)

## Пример:

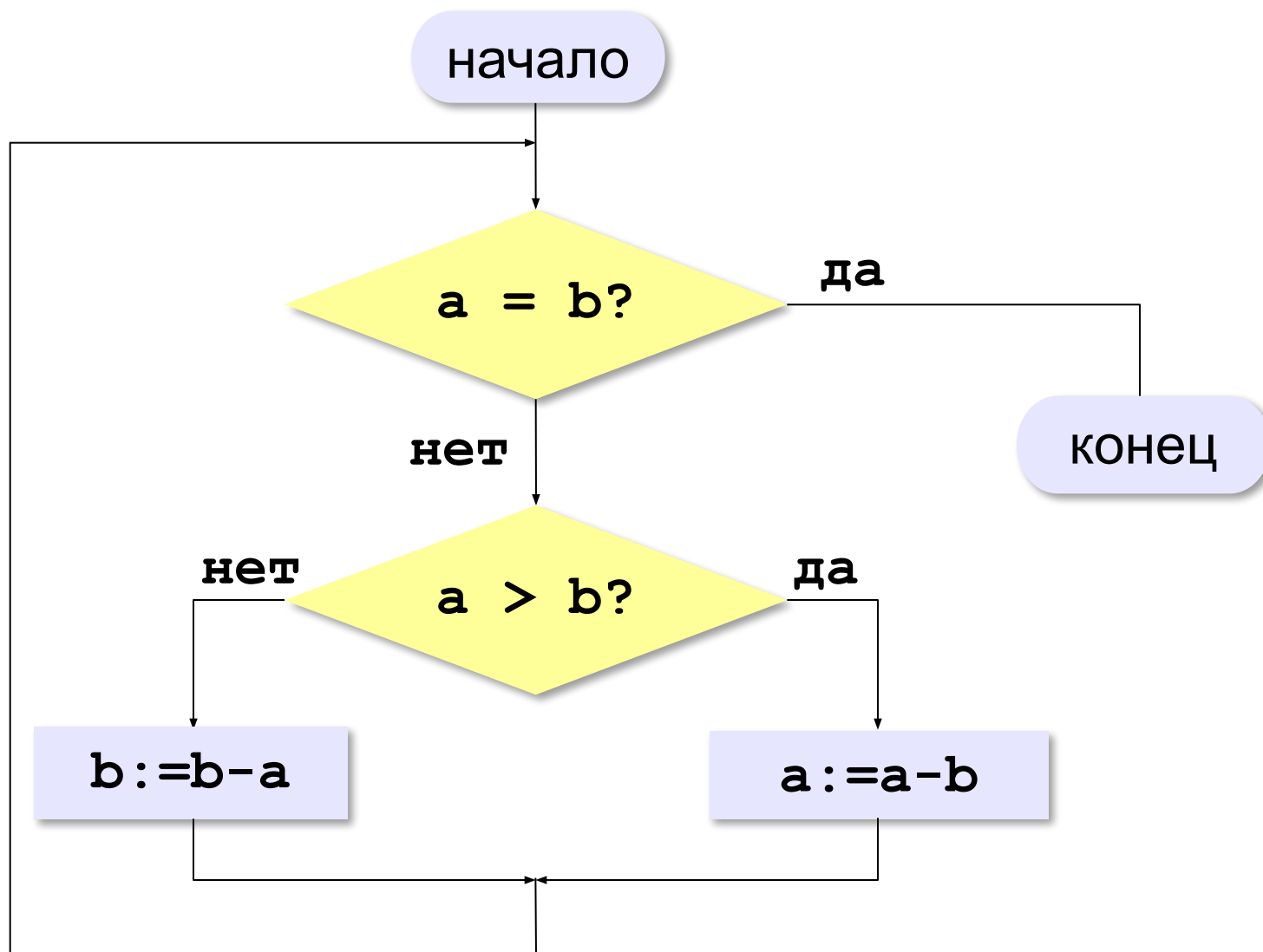
$$\begin{aligned}\text{НОД}(14, 21) &= \text{НОД}(14, 21 - 14) = \text{НОД}(14, 7) \\ &= \text{НОД}(7, 7) = 7\end{aligned}$$



много шагов при большой разнице чисел:

$$\text{НОД}(1998, 2) = \text{НОД}(1996, 2) = \dots = 2$$

# Алгоритм Евклида



# Алгоритм Евклида

```
нц пока  $a \neq b$   
  если  $a > b$   
    то  $a := a - b$   
  иначе  $b := b - a$   
все  
кц
```

- ? Где будет НОД? Как его вывести?
- ? Как вывести НОД в формате  $\text{НОД}(14, 21) = 7$ ?
- ? А без дополнительных переменных?

# Модифицированный алгоритм Евклида

Заменяем большее из двух чисел **остатком от деления** большего на меньшее до тех пор, пока меньшее не станет **равно нулю**. Тогда большее — это НОД.

$$\begin{aligned}\text{НОД}(a, b) &= \text{НОД}(\text{mod}(a, b), b) \\ &= \text{НОД}(a, \text{mod}(b, a))\end{aligned}$$

**Пример:**

$$\text{НОД}(14, 21) = \text{НОД}(14, 7) = \text{НОД}(0, 7) = 7$$

# Модифицированный алгоритм

```
нц пока  $a \neq 0$  и  $b \neq 0$   
  если  $a > b$  то  
     $a := \text{mod}(a, b)$   
  иначе  
     $b := \text{mod}(b, a)$   
все  
кц
```

**?** Где будет НОД? Как его вывести?

```
если  $a \neq 0$   
  вывод  $a$   
иначе  
  вывод  $b$   
все
```



вывод  $a+b$



# В других языках программирования

## Паскаль:

```
while (a<>0) and  
      (b<>0) do  
  if a>b then  
    a:= a mod b  
  else  
    b:= b mod a;
```

## C:

```
while (a!=0 && b!=0)  
{  
  if (a > b)  
    a = a % b;  
  else  
    b = b % a;  
}
```

## Python:

```
while a!=0 and b!=0:  
  if a > b:  
    a = a % b  
  else:  
    b = b % a
```

# Задачи

«А»: Ввести с клавиатуры два натуральных числа и найти их НОД с помощью алгоритма Евклида.

**Пример:**

Введите два числа:

21 14

$\text{НОД}(21, 14) = 7$

«В»: Ввести с клавиатуры два натуральных числа и найти их НОД с помощью **модифицированного** алгоритма Евклида. Заполните таблицу:

a	64168	358853	6365133	17905514	549868978
b	82678	691042	11494962	23108855	298294835
НОД (a, b)					

# Задачи

---

**«С»:** Ввести с клавиатуры два натуральных числа и сравнить количество шагов цикла для вычисления их НОД с помощью обычного и модифицированного алгоритмов Евклида.

**Пример:**

Введите два числа:

**1998 2**

НОД(1998, 2) = 2

Обычный алгоритм: 998

Модифицированный: 1

# Обработка потока данных

**Задача.** На вход программы поступает поток данных — последовательность целых чисел, которая **заканчивается нулём**. Требуется найти сумму элементов этой последовательности.

```
нц пока  $x \neq 0$   
  |  добавить  $x$  к сумме  
  |   $x :=$  следующее число  
кц
```



Откуда возьмётся  $x$  в первый раз?

# Обработка потока данных

```
цел x, sum
sum := 0
ввод x | ввести первое число
нц пока x <> 0
    sum := sum + x
    ввод x | ввести следующее
кц
вывод 'Сумма ', sum
```



Как найти сумму положительных?

# Задачи

---

- «А»: На вход программы поступает неизвестное количество чисел целых, ввод заканчивается нулём. Определить, сколько получено чисел, которые делятся на 3.
- «В»: На вход программы поступает неизвестное количество чисел целых, ввод заканчивается нулём. Определить, сколько получено двузначных чисел, которые заканчиваются на 3.
- «С»: На вход программы поступает неизвестное количество чисел целых, ввод заканчивается нулём. Найти максимальное из введённых чётных чисел.

# Цикл с постусловием

- условие проверяется **после** завершения очередного шага цикла
- цикл всегда выполняется хотя бы один раз
- как только условие становится **ИСТИННЫМ**, работа цикла заканчивается

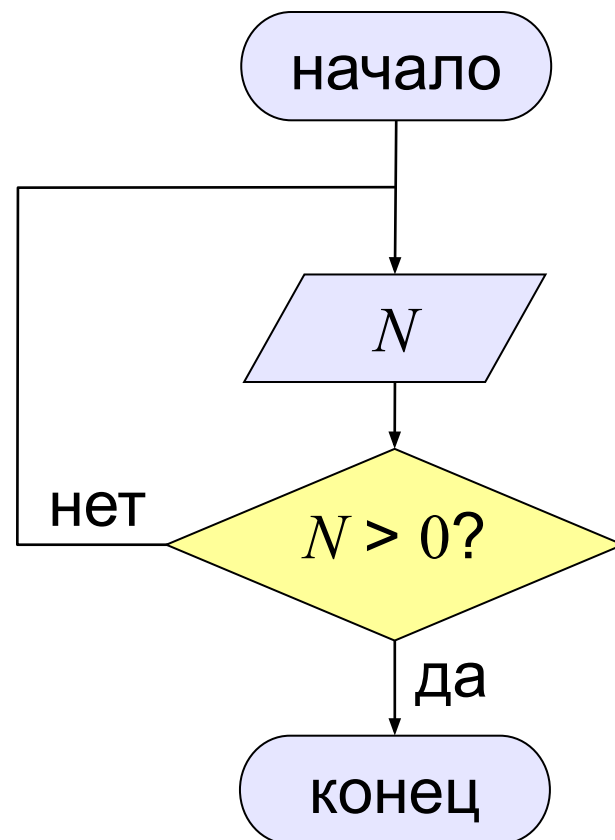
нц

**вывод** 'Введите N>0: '

**ввод** N

кц при N > 0

условие **окончания**  
работы цикла



# Задачи

---

- «А»: Напишите программу, которая предлагает ввести пароль и не переходит к выполнению основной части, пока не введён правильный пароль. Основная часть – вывод на экран «секретных сведений».
- «В»: Напишите программу, которая получает с клавиатуры натуральное число, которое больше 1, и определяет, простое оно или нет. Для этого нужно делить число на все натуральные числа, начиная с 2, пока не получится деление без остатка.
- «С»: Напишите программу, которая получает с клавиатуры два целых числа и вычисляет их произведение, используя только операции сложения.



# Задачи

---

«D»: Напишите программу, которая получает с клавиатуры натуральное число и вычисляет целый квадратный корень из него – наибольшее число, квадрат которого не больше данного числа.

# Цикл по переменной

*Задача.* Вывести на экран степени числа 2 от  $2^1$  до  $2^{10}$ .

**k := 1**

**N := 2**

**нц пока k <= 10**

**вывод N, нс**

**N := N \* 2**

**k := k + 1**

**кц**



Работа с k в трёх местах!

*Идея:* собрать всё вместе.

**N := 2**

**нц для k от 1 до 10**

**вывод N, нс**

**N := N \* 2**

**кц**

увеличение на 1  
по умолчанию

# Цикл по переменной

Задача. Найти сумму чисел от 1 до 1000.

```
цел sum, i
sum := 0
нц для i от 1 до 1000
    sum := sum + i
кц
```

Задача. Вывести квадраты чисел от 10 до 1 по убыванию.

```
нц для k от 10 до 1 шаг -1
    вывод k*k, нс
кц
```

любое целое

# Цикл по переменной

Задача. Найти сумму чётных чисел от 2 до 1000.

```
sum := 0
нц для i от 1 до 1000
  если mod(i, 2) = 0 то
    sum := sum + i
  все
кц
```



Что плохо?

↓

```
sum := 0
нц для i от 2 до 1000 шаг 2
  sum := sum + i
кц
```

# В других языках программирования

## Паскаль:

```
sum := 0;  
for i := 1 to 1000 do  
    sum := sum + i;
```

шаг только 1 или  
-1 (downto)

## C:

```
int sum, i;  
sum = 0;  
for (i = 1; i <= 1000; i++)  
    sum += i;
```

i = i + 1;

sum = sum + i;

## Python:

```
Sum = 0  
for i in range(1, 1001):  
    Sum += i
```

диапазон [1;1001)

# Задачи

---

«А»: Ипполит задумал трёхзначное число, которое при делении на 15 даёт в остатке 11, а при делении на 11 даёт в остатке 9. Напишите программу, которая находит все такие числа.

«В»: С клавиатуры вводится натуральное число  $N$ . Программа должна найти факториал этого числа (обозначается как  $N!$ ) – произведение всех натуральных чисел от 1 до  $N$ . Например,

$$5! = 1 \cdot 2 \cdot 3 \cdot 4 \cdot 5 = 120.$$

«С»: Натуральное число называется **числом Армстронга**, если сумма цифр числа, возведенных в  $N$ -ную степень (где  $N$  – количество цифр в числе) равна самому числу. Например,  $153 = 1^3 + 5^3 + 3^3$ . Найдите все трёхзначные Армстронга.