

Школьный алгоритмический язык

Школьный алгоритмический язык (АЯ)— формальный язык)— формальный язык, используемый для записи, реализации и изучения алгоритмов.

Алгоритмический язык с русским синтаксисом был введён в употребление академиком А. П. Ершовым в середине 1980-х годов, в качестве основы для «безмашинного» курса информатики.

Для единообразной записи алгоритмов введена система обозначений и правил – псевдокод.

- ❑ В псевдокоде не приняты строгие синтаксические правила для записи команд, присущие формальным языкам, что облегчает запись алгоритма на стадии его проектирования.
- ❑ В псевдокоде, так же, как и в формальных языках, есть служебные слова, смысл которых определен раз и навсегда. Они выделяются в печатном тексте жирным шрифтом, а в рукописном тексте подчеркиваются. Единого или формального определения псевдокода не существует, поэтому возможны различные псевдокоды, отличающиеся набором служебных слов и основных (базовых) конструкций.
- ❑ Примером псевдокода является школьный алгоритмический язык в русской нотации (школьный АЯ),

Основные служебные слова

алг (алгоритм)	дано	или
арг (аргумент)	надо	не
рез (результат)	если	да
нач (начало)	то	нет
кон (конец)	иначе	при
цел (целый)	все	выбор
вещ (вещественный)	пока	ввод
сим (символьный)	для	вывод
лог (логический)	от	утв
таб (таблица)	до	
нц (начало цикла)	знач	
кц (конец цикла)	и	

Общий вид алгоритма:

алг название алгоритма (аргументы и результаты)

дано условия применимости алгоритма

надо цель выполнения алгоритма

нач описание промежуточных величин
последовательность команд (тело алгоритма)

кон

- Заголовок - часть алгоритма от слова **алг** до слова **нач**.
- Тело алгоритма - часть, заключенная между словами **нач** и **кон**.
- В заголовке после названия алгоритма в круглых скобках указываются характеристики (арг, рез) и тип значения (цел, вещ, сим, лит или лог) всех входных (аргументы) и выходных (результаты) переменных.
- При описании массивов (таблиц) используется служебное слово **таб**, дополненное граничными парами по каждому индексу элементов массива.

Команды школьного АЯ

- **Оператор присваивания:** $A := B$
знак " $:=$ " означает команду заменить прежнее значение переменной, стоящей в левой части, на вычисленное значение выражения, стоящего в правой части.
Например, $a := (b+c)*4$
 $i := i+1$
- **Для ввода и вывода данных** используют команды **ВВОД** имена переменных
ВЫВОД имена переменных, выражения, тексты.
- **Для ветвления** применяют команды **если** и **выбор**,
для **организации циклов** — команды **для** и **пока**.

Пример вычисления суммы квадратов

алг Сумма квадратов (арг цел n , рез цел S)

дано $n > 0$

Надо $S = 1*1 + 2*2 + 3*3 + \dots + n*n$

нач цел i

ВВОД n ; $S:=0$

НЦ для i от 1 до n $S:=S+i*i$

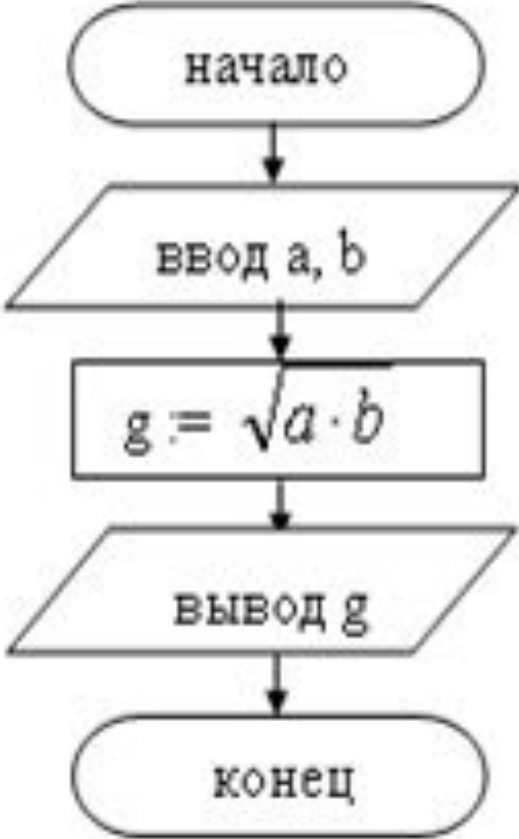
КЦ

ВЫВОД "S = ", S

КОН

Соответствие блок-схемы и АЯ

Линейный алгоритм:

Блок-схема	Псевдокоды
 <pre>graph TD; Start([начало]) --> Input[/ввод a, b/]; Input --> Process[g := √(a · b)]; Process --> Output[/вывод g/]; Output --> End([конец]);</pre>	<p><u>алг</u> среднее геометрическое <u>вещ</u> a, b, g</p> <p><u>нач</u></p> <p><u>ввод</u> a, b</p> <p>$g := (a * b)^{(1/2)}$</p> <p><u>вывод</u> g</p> <p><u>кон</u></p>

Алгоритм-ветвление:

Блок – схема



Псевдокоды

алг большее из чисел

вещ a, b, маx

нач

ввод a, b

если a > b

то маx := a

иначе маx := b

все

вывод маx

кон

Алгоритм – цикл:

Блок – схема	Псевдокоды
<pre>graph TD; Start([Начало]) --> Input[/Ввод n/]; Input --> S0[S:=0]; S0 --> i1[i:=1]; i1 --> Cond{i <= n}; Cond -- да --> Splus[S:=S+i]; Splus --> iplus[i:=i+2]; iplus --> Cond; Cond -- нет --> Output[/Вывод S/]; Output --> End([Конец]);</pre>	<p><u>алг</u> сумма нечетных чисел</p> <p><u>нач</u></p> <p><u>ввод</u> n</p> <p>S:=0</p> <p>i:=1</p> <p><u>пока</u> i<=n</p> <p>S:=S+i</p> <p>i:=i+2</p> <p><u>кц</u></p> <p><u>вывод</u> S</p> <p><u>кон</u></p>

Основные понятия в алгоритмическом языке

Имена (идентификаторы) — употребляются для обозначения объектов программы (переменных, массивов, функций и др.).

МОЖНО использовать

- латинские буквы (A-Z), русские буквы (А-Я)

заглавные и строчные буквы различаются

- цифры

имя не может начинаться с цифры

- знак подчеркивания _

НЕЛЬЗЯ использовать

- ~~скобки~~
- ~~знаки +, =, !, ? и др.~~

Операции : арифметические операции

+ сложение - вычитание

* умножение / деление

`div` деление нацело (остаток отбрасывается)

`mod` остаток от деления

```
цел a, b
```

```
a := 7*3 - 4
```

```
a := a * 5
```

```
b := div(a, 10)
```

```
a := mod(a, 10)
```

логические операции : И, ИЛИ, НЕ;

отношения <, >, <=, >=, =, <> .

Данные - величины, обрабатываемые программой:

Константы — это данные, которые зафиксированы в тексте программы и не изменяются в процессе ее выполнения:

- **числовые** 7,5 ; 12
- **логические** да (истина), нет (ложь);
- **символьные** "А", "+".

Переменные обозначаются именами и могут изменять свои значения в ходе выполнения программы:

- **целые,**
- **вещественные,**
- **логические,**
- **символьные.**

Массив (таблица) — последовательность однотипных элементов, число которых фиксировано и которым присвоено одно имя. Положение элемента в массиве однозначно определяется его индексами.

Какие операторы неправильные?

алг Ошибки

нач

цел a, b

вещ x, y

a := 5

10 := x

y := 7,8

b := 2.5

x := 2 * (a + y)

a := b + x

имя переменной должно
быть слева от знака :=

целая и дробная часть
отделяются точкой

нельзя записывать
вещественное значение в
целую переменную

кОН

Вывод данных

ВЫВОД a

| вывод значения
| переменной a

ВЫВОД a, \n

| вывод значения
| переменной a и переход
| на новую строку

ВЫВОД "Привет!"

| вывод текста

ВЫВОД "Ответ: ", c

| вывод текста и значения переменной c

ВЫВОД a, "+", b, "=", c

Команда «вывод»

```
цел a = 1, b = 3
```

```
вывод a, "+", b, "=", a+b
```

СПИСОК ВЫВОДА

- элементы разделяются запятыми
- элементы в кавычках – выводятся без изменений
- выражения (элементы без кавычек) вычисляются и выводится их результат



Что будет выведено?

1+3=4

Что будет выведено?

цел $a = 1, b = 3$

ВЫВОД "a+", b, "=a+b"

$a+3=a+b$

цел $a = 1, b = 3$

ВЫВОД a, "=F(", b, ")"

$1=F(3)$

цел $a = 1, b = 3$

ВЫВОД "a=F(", b, ");"

$a=F(3);$

цел $a = 1, b = 3$

ВЫВОД $a+b$, ">", b, "!"

$4>3!$

цел $a = 1, b = 3$

ВЫВОД "F(", b, ")=X(", a, ")"

$F(3)=X(1)$

Как записать оператор «вывод»?

цел $a = 1, b = 3$

Вывод `"X(", b, "=", a`

$X(3) = 1$

цел $a = 1, b = 3$

Вывод `a+b, "=", a, "+", b`

$4 = 1 + 3$

цел $a = 1, b = 3$

Вывод `"f(", a, ") > f(", b, ")"`

$f(1) > f(3)$

цел $a = 1, b = 3$

Вывод `"<", a, "<>", b, ">"`

$\langle 1 \langle \rangle 3 \rangle$

цел $a = 1, b = 3$

Вывод `a, "+", b, "=?"`

$1 + 3 = ?$

Выражения — предназначены для выполнения необходимых вычислений, состоят из констант, переменных, указателей функций, объединенных знаками операций.

Таблица стандартных функций

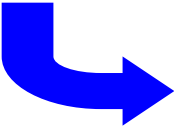
Название и математическое обозначение функции		Указатель функции
Абсолютная величина (модуль)	$ x $	abs(x)
Корень квадратный		sqrt(x)
Натуральный логарифм	$\ln x$	ln(x)
Десятичный логарифм	$\lg x$	lg(x)
Экспонента (степень числа $e \approx 2.72$)	e^x	exp(x)
Знак числа x (-1, если $x < 0$; 0, если $x = 0$; 1, если $x > 0$)	$\text{sign } x$	sign(x)
Целая часть x (т.е. максимальное целое число, не превосходящее x)		int(x)
Минимум из чисел x и y		min(x,y)
Максимум из чисел x и y		max(x,y)
Случайное число в диапазоне от 0 до $x-1$		rnd(x)
Синус (угол в радианах)	$\sin x$	sin(x)
Косинус (угол в радианах)	$\cos x$	cos(x)
Тангенс (угол в радианах)	$\text{tg } x$	tg(x)

Порядок выполнения операций

- 1) вычисление выражений в скобках
- 2) умножение, деление, `div`, `mod` слева направо
- 3) сложение и вычитание слева направо

1 2 4 5 3 6
z := (5*a+c) / a* (b-c) / b

$$x = \frac{5c^2 - d(a+b)}{(c+d)(d-2a)}$$


$$z = \frac{5a+c}{ab} (b-c)$$

2 3 5 4 1 10 6 9 8 7
x := (5*c*c-d*(a+b)) / ((c+d) * (d-2*a))

Примеры записи арифметических выражений

Математическая запись	Запись на школьном алгоритмическом языке
$\frac{xy}{z}$	<code>x*y/z</code>
$\frac{x}{yz}$	<code>x/(y*z)</code> или <code>x/y/z</code>
$\frac{a^3 + b^3}{bc}$	<code>(a**3+b**3)/(b*c)</code>
$\frac{a_{i+1} + b_{i-1}}{2xy}$	<code>(a[i+1]+b[i-1])/(2*x*y)</code>
$\frac{-b + \sqrt{b^2 - 4ac}}{2a}$	<code>(-b+sqrt(b*b-4*a*c))/(2*a)</code>
$0,49e^{a^3-b^3} + \ln^3 \cos a^2$	<code>0.49*exp(a*a-b*b)+ln(cos(a*a))**3</code>
$1 + \frac{x}{3 + (2x)^3}$	<code>x/(1+x*x/(3+(2*x)**3))</code>

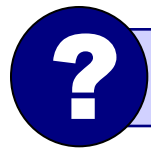
Примеры записи логических выражений, истинных при выполнении указанных условий

Условие	Запись на школьном АЯ
Дробная часть вещественного числа a равна нулю	$\text{int}(a) = 0$
Целое число a — четное	$\text{mod}(a,2) = 0$
Целое число a — нечетное	$\text{mod}(a,2) = 1$
Целое число k кратно семи	$\text{mod}(a,7) = 0$
Каждое из чисел a, b положительно	$(a > 0)$ и $(b > 0)$
Только одно из чисел a, b положительно	$((a > 0) \text{ и } (b \leq 0))$ или $((a \leq 0) \text{ и } (b > 0))$
Хотя бы одно из чисел a, b, c является отрицательным	$(a < 0)$ или $(b < 0)$ или $(c < 0)$
Число x удовлетворяет условию $a < x < b$	$(x > a)$ и $(x < b)$
Число x имеет значение в промежутке [1, 3] Число x имеет значение в промежутке [1, 3]	$(x \geq 1)$ и $(x \leq 3)$
Точка с координатами (x, y) лежит в круге радиуса r с центром в точке (a, b)	$(x-a)**2+(y-b)**2 < r*r$

Сложные условия

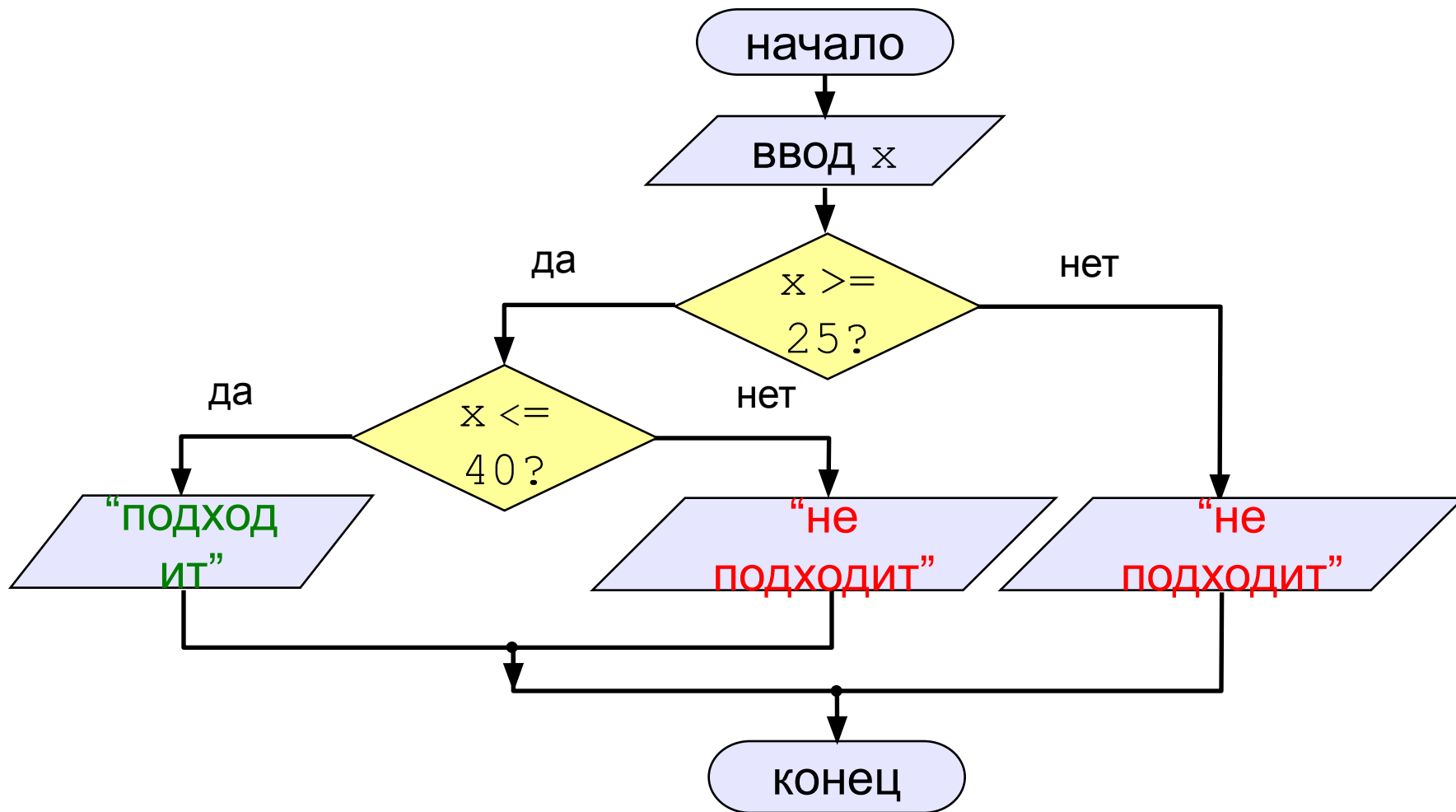
Задача. Фирма набирает сотрудников от 25 до 40 лет включительно. Ввести возраст человека и определить, подходит ли он фирме (вывести ответ «подходит» или «не подходит»).

Особенность: надо проверить, выполняются ли два условия одновременно.



Можно ли решить известными методами?

Вариант 1. Алгоритм



Вариант 1. Программа

```
алг Сотрудник
```

```
нач
```

```
    цел x
```

```
    вывод "Введите ваш возраст", нс
```

```
    ввод x
```

```
    если x >= 25 то
```

```
        если x <= 40 то
```

```
            вывод "Подходит!"
```

```
        иначе
```

```
            вывод "Не подходит."
```

```
        все
```

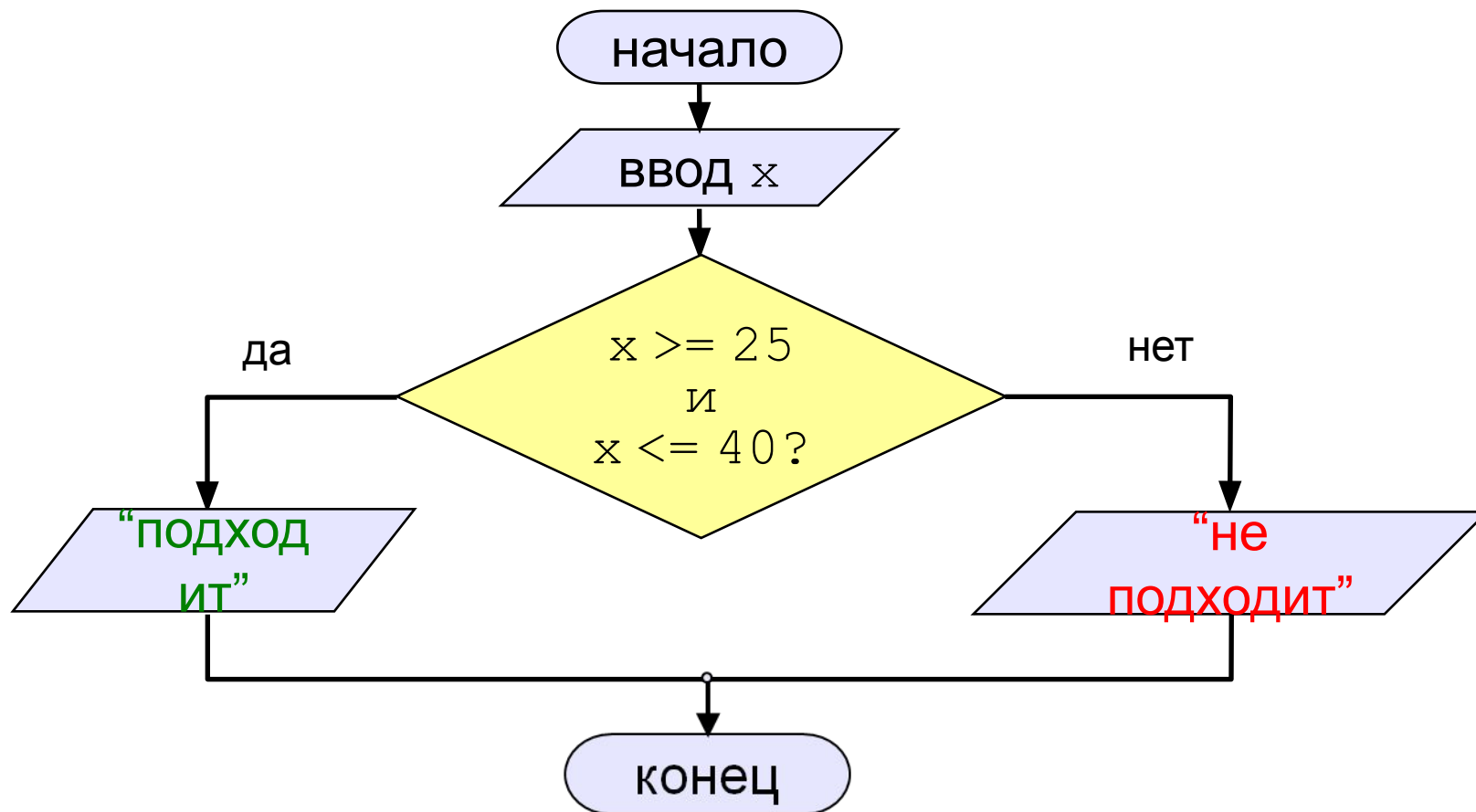
```
    иначе
```

```
        вывод "Не подходит."
```

```
    все
```

```
кон
```

Вариант 2. Алгоритм



Вариант 2. Программа

```
алг Сотрудник
```

```
нач
```

```
  цел x
```

```
  вывод "Введите ваш возраст", нс
```

```
  ввод x
```

```
  если  $x \geq 25$  и  $x \leq 40$  то
```

```
    вывод "Подходит!"
```

```
  иначе
```

```
    вывод "Не подходит."
```

```
  все
```

```
конец
```

сложное
условие

Массивы

Массив – это группа однотипных элементов, имеющих общее имя и расположенных в памяти рядом.

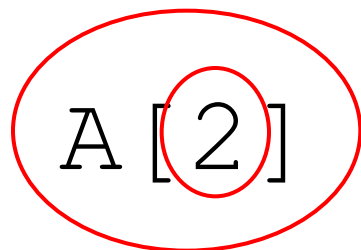
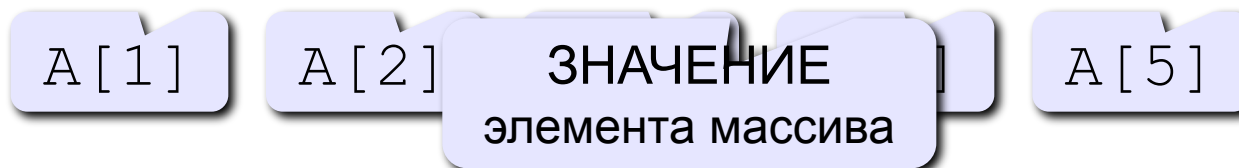
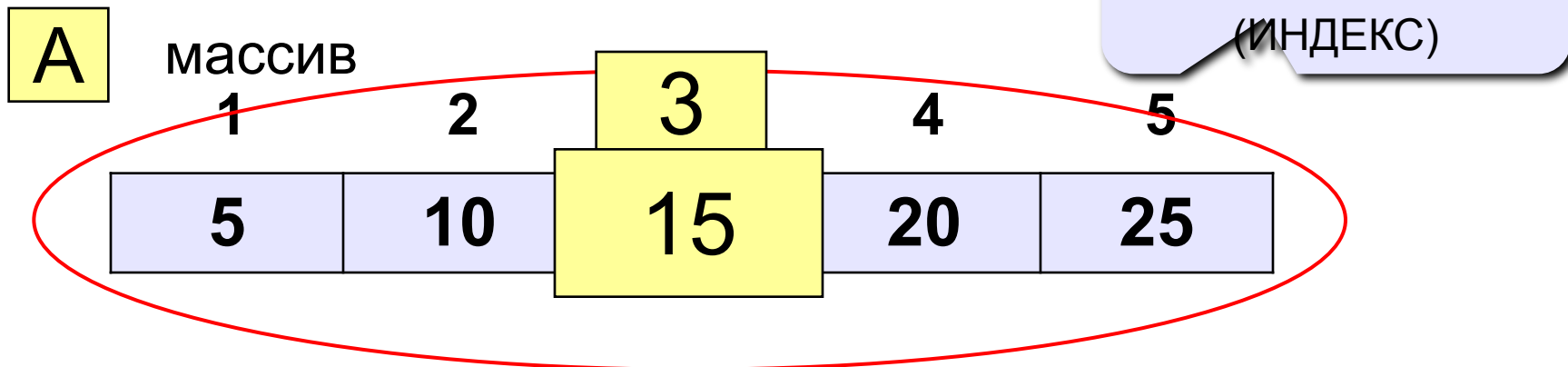
Особенности:

- все элементы имеют один тип
- весь массив имеет одно имя
- все элементы расположены в памяти рядом

Примеры:

- список учеников в классе
- квартиры в доме
- школы в городе
- данные о температуре воздуха за год

Массивы



НОМЕР (ИНДЕКС)
элемента массива: 2

ЗНАЧЕНИЕ
элемента массива: 10

Объявление массивов

Зачем объявлять?

- определить имя массива
- определить тип массива
- определить число элементов
- выделить место в памяти

Массив целых чисел:



Размер через переменную:

```
цел N = 5  
целтаб А [ 1 : N ]
```

Объявление массивов

Массивы других типов:

вещтаб X[1:10], Y[1:20] | вещественные

X[1] := 4.25

симтаб S[1:20] | СИМВОЛЬНЫЙ

S[1] := "z"

логтаб L[1:5] | логический

L[1] := да | да или нет

Другой диапазон индексов:

целтаб Q[0:9]

вещтаб C[-5:13]

Что неправильно?

```
целтаб А [1:1  
          0]
```

...

```
А[5] := 4.5;
```

```
целтаб А[1:10]
```

...

```
А[15] := 'а'
```


Как обработать все элементы массива?

Объявление:

```
цел N = 5  
целтаб A[1:N]
```

Обработка:

```
| обработать A[1]  
| обработать A[2]  
| обработать A[3]  
| обработать A[4]  
| обработать A[5]
```

?

1) если N велико (1000, 1000000)?

2) при изменении N программа не должна меняться!

Как обработать все элементы массива?

Обработка с переменной:

```
i := 1  
| обработать A[i]  
i := i + 1  
| обработать A[i]  
i := i + 1  
| обработать A[i]  
i := i + 1  
| обработать A[i]  
i := i + 1  
| обработать A[i]
```

```
i := i + 1
```



Обработка в цикле:

```
i := 1  
нц пока i <= N  
    | обработать A[i]  
    i := i + 1  
кц
```

Цикл с переменной:

```
нц для i от 1 до N  
    | обработать A[i]  
кц
```



Простейшая программа

Объявление:

```
цел N = 5, i  
целтаб A[1:N]
```

Ввод с клавиатуры:

```
нц для i от 1 до N  
    вывод "A[" , i, "]" = "  
    ввод A[i]  
кц
```

```
A[1] = 5  
A[2] = 12  
A[3] = 34  
A[4] = 56  
A[5] = 13
```

Обработка каждого элемента:

```
нц для i от 1 до N  
    A[i] := A[i] + 1  
кц
```

Вывод на экран:

```
вывод "Массив A", нс  
нц для i от 1 до N  
    вывод A[i], " "  
кц
```

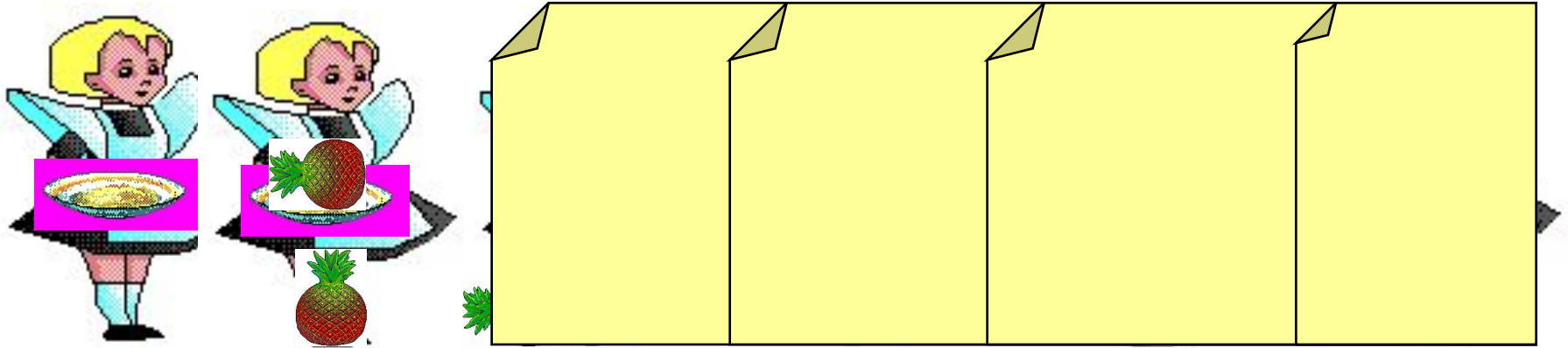
Массив A:

6 13 35 57 14

Максимальный элемент

Задача: найти в массиве максимальный элемент.

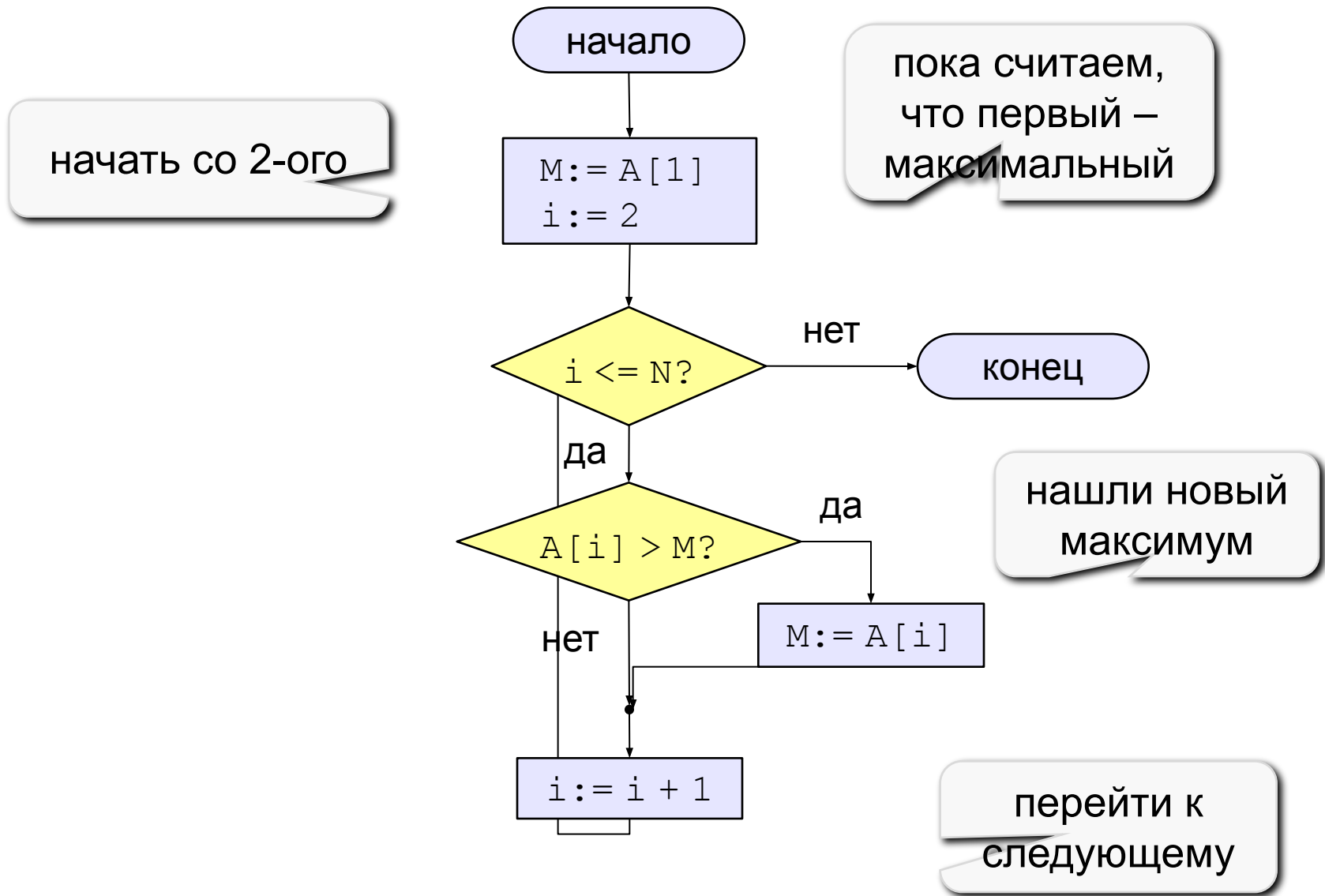
Алгоритм:



Решение:

- 1) считаем, что первый элемент – максимальный
- 2) просмотреть остальные элементы массива:
если очередной элемент $> M$,
то записать $A[i]$ в M
- 3) вывести значение M

Максимальный элемент



Максимальный элемент

```
алг Максимум
```

```
нач
```

```
  цел  $i$ ,  $N = 5$ ,  $M$ 
```

```
  целтаб  $A[1:N]$ 
```

```
  | здесь заполнить массив
```

```
   $M := A[1]$  | пока максимальный -  $A[1]$ 
```

```
  нц для  $i$  от 2 до  $N$ 
```

```
    если  $A[i] > M$  то  $M := A[i]$  все
```

```
кц
```

```
вывод "Максимальный элемент ",  $M$ 
```

```
кон
```

Максимальный элемент

Дополнение: как найти номер максимального элемента?

пока первый – максимальный

```
iMax := 1
```

```
нц для i от 2 до N | проверяем остальные
```

```
если  $A[i] > A[iMax]$  то | нашли еще больше
```

```
    iMax := i | запомнить i
```

```
все
```

```
кц
```

```
вывод "Максимальный элемент A[" , iMax, "] = " ,  $A[iMax]$ 
```



Как упростить?

По номеру элемента $iMax$ всегда можно найти его значение $A[iMax]$. Поэтому везде меняем M на $A[iMax]$ и убираем переменную M .