

Программирование на алгоритмическом языке. Часть II

1. [Массивы](#)
2. [Максимальный элемент массива](#)
3. [Обработка массивов](#)
4. [Символьные строки](#)
5. [Матрицы](#)
6. [Алгоритмы-функции](#)
7. [Построение графиков функций](#)

Программирование на алгоритмическом языке. Часть II

Тема 1. Массивы

Массивы

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

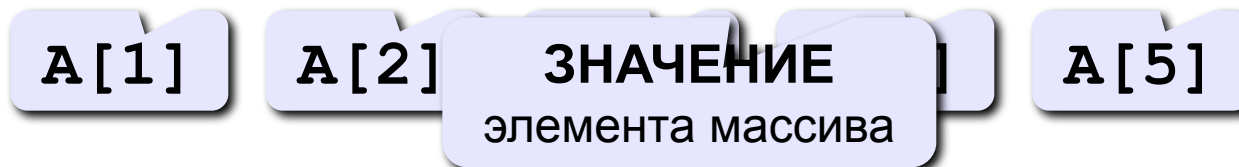
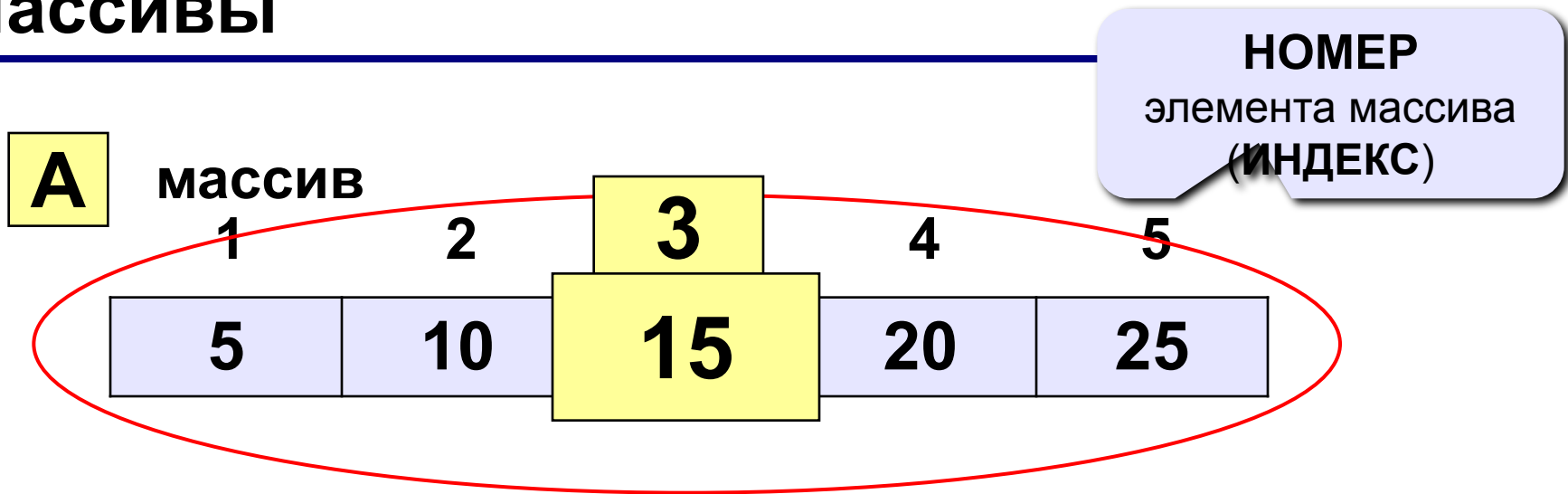
Особенности:

- 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

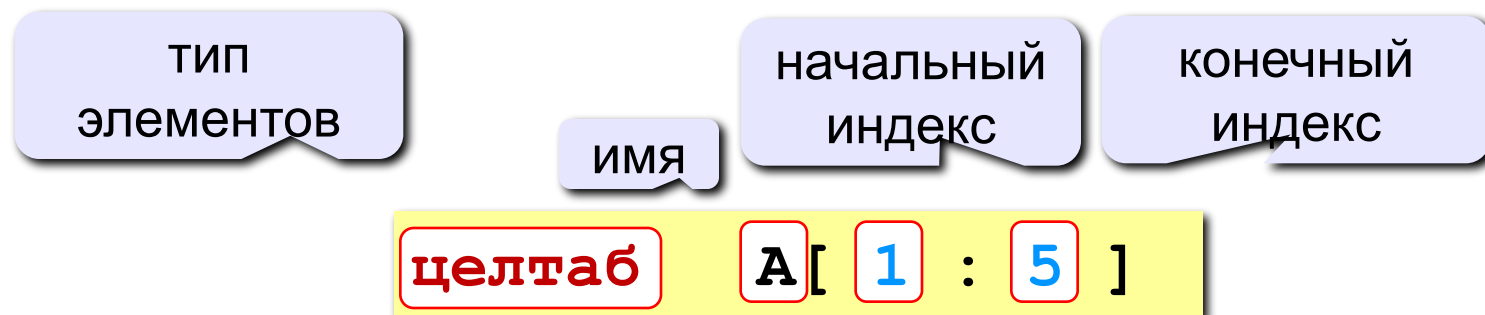
ЗНАЧЕНИЕ
элемента массива: 10

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: Вывод на экран:

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

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

Массив A:

6 13 35 57 14

Задания

«3»: Ввести с клавиатуры массив из 5 элементов, умножить их на 2 и вывести на экран.

Пример:

Введите пять чисел:

4 15 3 10 14

Результат: 8 30 6 20 28

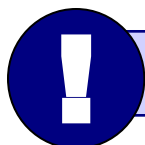
«4»: Ввести с клавиатуры массив из 5 элементов, найти среднее арифметическое всех элементов массива.

Пример:

Введите пять чисел:

4 15 3 10 14

среднее арифметическое 9.200



При изменении N остальная программа не должна изменяться!

Задания

«5»: Ввести с клавиатуры массив из 5 элементов, найти минимальный из них.

Пример:

Введите пять чисел:

4 15 3 10 14

минимальный элемент 3

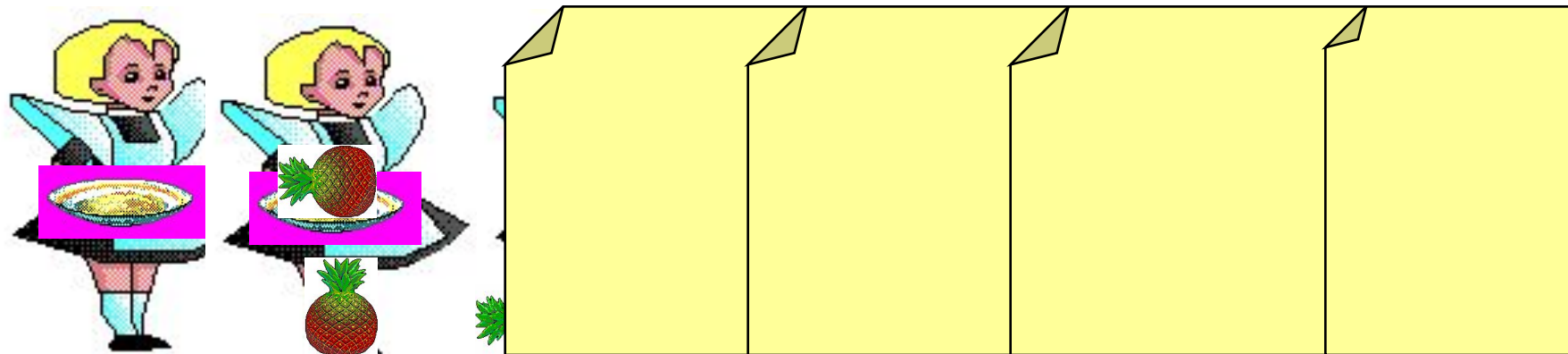
Программирование на алгоритмическом языке. Часть II

Тема 2. Максимальный элемент массива

Максимальный элемент

Задача: найти в массиве максимальный элемент.

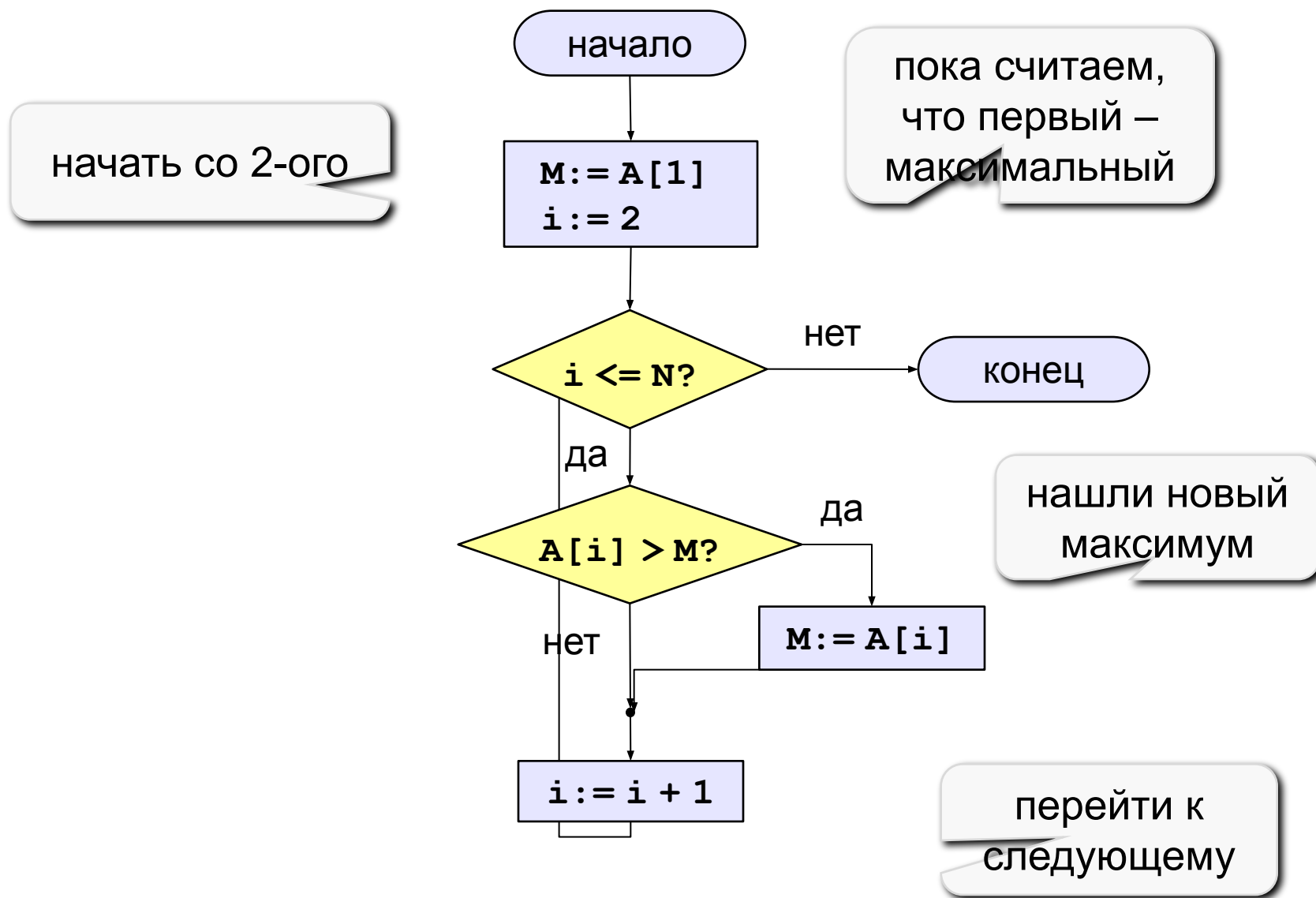
Алгоритм:



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**.

Задания

«3»: Ввести с клавиатуры массив из 5 элементов, найти в нем минимальный элемент и его номер.

Пример:

Исходный массив:

4 -5 10 -10 5

минимальный $A[4] = -10$

«4»: Ввести с клавиатуры массив из 5 элементов, найти в нем максимальный и минимальный элементы и их номера.

Пример:

Исходный массив:

4 -5 10 -10 5

максимальный $A[3] = 10$

минимальный $A[4] = -10$

Задания

«5»: Ввести с клавиатуры массив из 5 элементов, найти в нем два максимальных элемента и их номера.

Пример:

Исходный массив:

4 -5 10 -10 5

максимальные $A[3]=10$, $A[5]=5$

Программирование на алгоритмическом языке. Часть II

Тема 3. Обработка массивов

Случайность и ее моделирование

Случайно...

- 1) встретить друга на улице
- 2) разбить тарелку
- 3) найти 10 рублей
- 4) выиграть в лотерею

Случайный выбор:

- 1) жеребьевка на соревнованиях
- 2) выигравшие номера в лотерее

Как получить случайность?



Случайные числа на компьютере

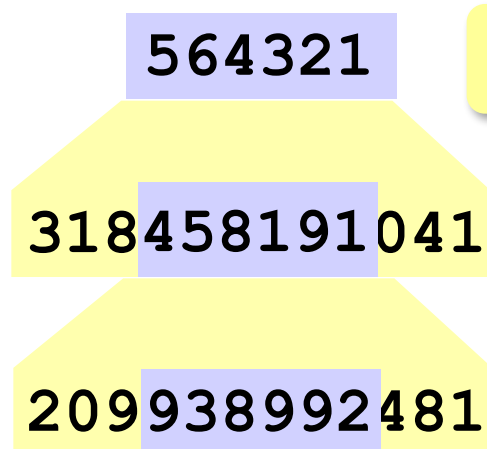
Электронный генератор



- нужно специальное устройство
- нельзя воспроизвести результаты

Псевдослучайные числа – обладают свойствами случайных чисел, но каждое следующее число вычисляется по заданной формуле.

Метод середины квадрата (Дж. фон Нейман)

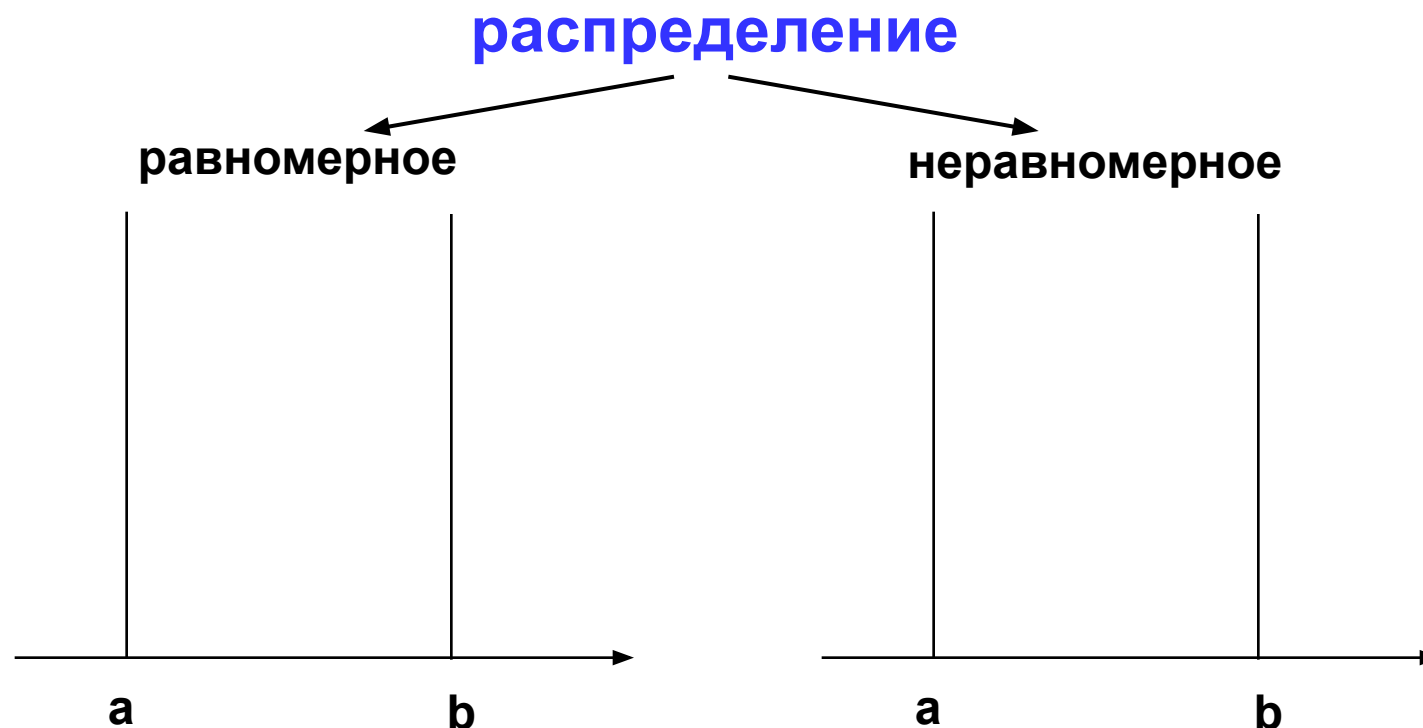


в квадрате

алый период
(последовательность
повторяется через 10^6 чисел)

Распределение случайных чисел

Модель: снежинки падают на отрезок $[a, b]$

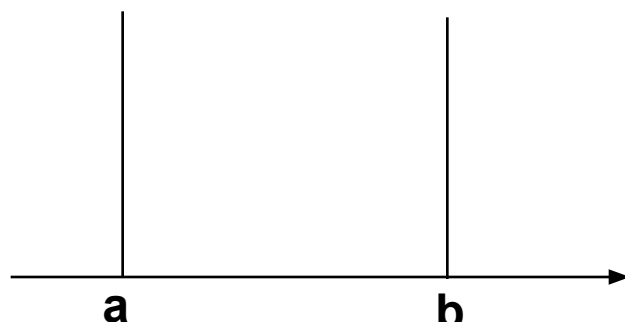



Сколько может быть разных распределений?

Распределение случайных чисел

Особенности:

- распределение – это характеристика **всей последовательности**, а не одного числа
- **равномерное** распределение одно, компьютерные датчики случайных чисел дают равномерное распределение
- неравномерных – много
- любое неравномерное можно получить с помощью равномерного


$$x = \frac{x_1 + x_2}{2}$$


$$x = \frac{x_1 + x_2 + \dots + x_{12}}{12}$$

Генератор случайных чисел

Вещественные числа в интервале $[0,10)$:

вещ X, Y

X := rand(0, 10) | интервал от 0 до 10 (<10)

Y := rand(0, 10) | это уже другое число!

англ. *random* – случайный

Целые числа в интервале $[0,10]$:

цел K, L

K := irand(0, 10) | интервал от 0 до 10 (<10)

L := irand(0, 10) | это уже другое число!

англ. *integer* – целый

Заполнение массива случайными числами

```
цел i, N = 5  
целтаб A[1:N]  
вывод "Массив ", нс  
нц для i от 1 до N  
    A[i] := irand(10, 30);  
    вывод A[i], " "  
кц
```



Зачем сразу выводить?

Подсчет элементов

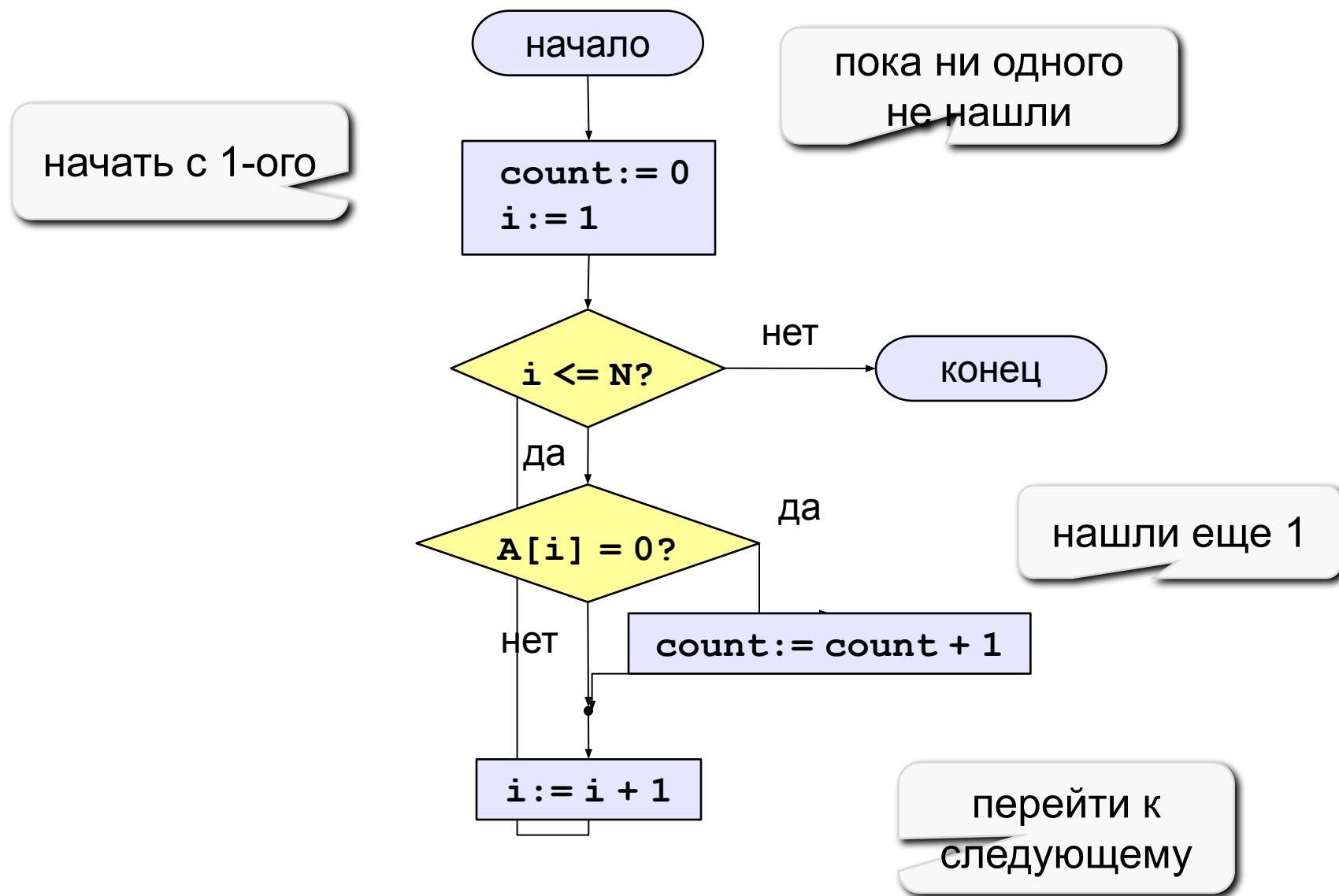
Задача: заполнить массив случайными числами в интервале $[-1, 1]$ и подсчитать количество нулевых элементов.

Идея: используем переменную-счётчик.

Решение:

- 1) записать в счётчик ноль
- 2) просмотреть все элементы массива:
 если очередной элемент = 0,
 то увеличить счётчик на 1
- 3) вывести значение счётчика

Подсчет элементов



Подсчет элементов

```
цел count, i, N = 10
```

```
целтаб A[1:N]
```

```
| здесь надо заполнить массив
```

```
count := 0
```

```
нц для i от 1 до N
```

```
    если A[i] = 0
```

```
        то count := count + 1
```

```
    все
```

```
кц
```

перебираем все
элементы массива

Задания

- «3»:** Заполнить массив случайными числами в интервале $[-2, 2]$ и подсчитать количество положительных элементов.
- «4»:** Заполнить массив случайными числами в интервале $[20, 100]$ и подсчитать отдельно число чётных и нечётных элементов.
- «5»:** Заполнить массив случайными числами в интервале $[1000, 2000]$ и подсчитать число элементов, у которых вторая с конца цифра – четная.

Сумма выбранных элементов

Задача: заполнить массив случайными числами в интервале $[-10, 10]$ и подсчитать сумму положительных элементов.

Идея: используем переменную S для накопления суммы.

$S := 0$ $S := A[1]$ $S := A[1] + A[2]$

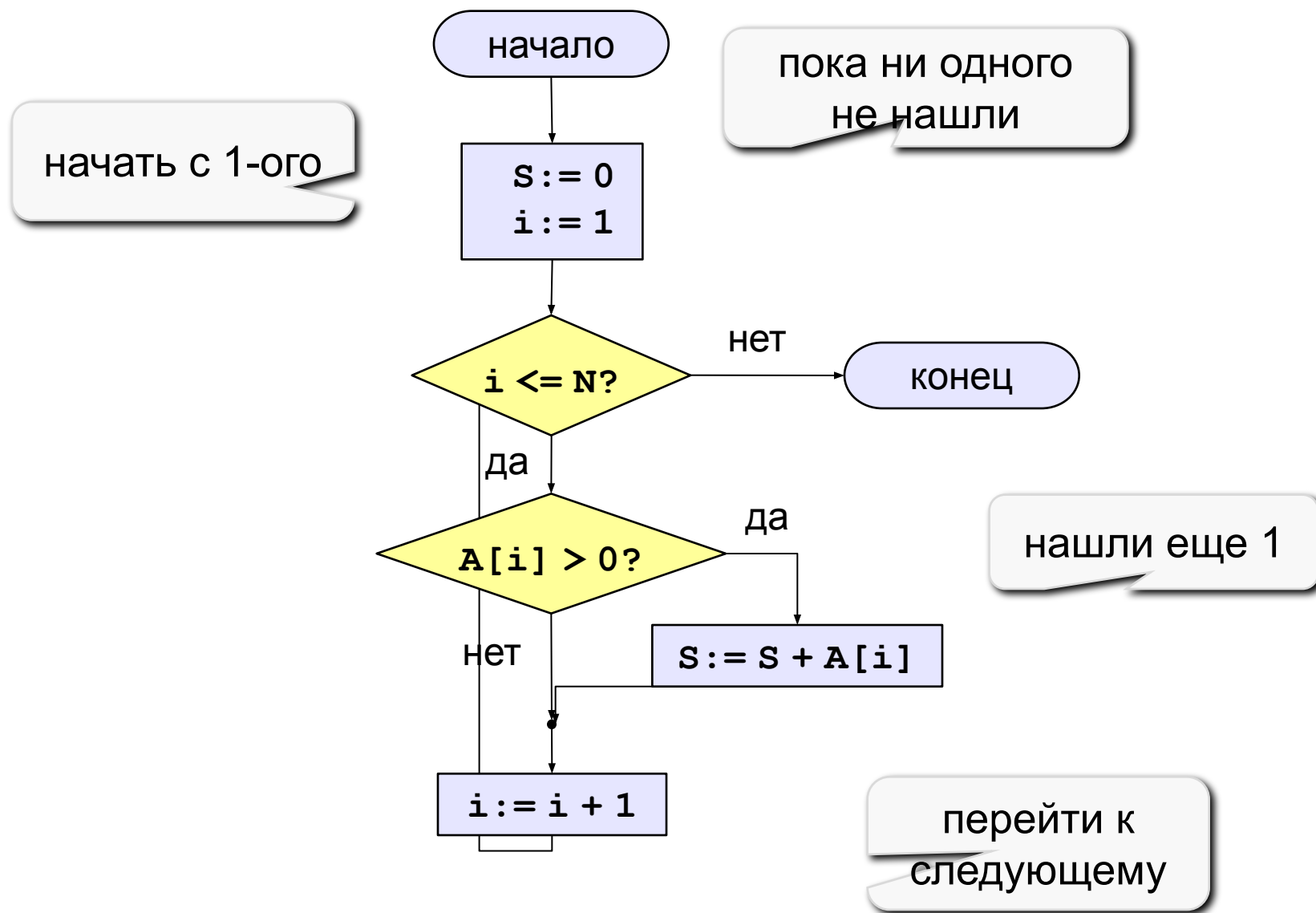
$S := A[1] + A[2] + A[3] \rightarrow S := A[1] + A[2] + \dots + A[N]$

Решение:

- 1) записать в переменную S ноль
- 2) просмотреть все элементы массива:
 если очередной элемент > 0 ,
 то добавить к сумме этот элемент
- 3) вывести значение суммы

$S := S + A[i]$

Сумма выбранных элементов



Сумма выбранных элементов

```
цел S, i, N = 10
```

```
целтаб A[1:N]
```

| здесь надо заполнить массив

```
S := 0
```

```
нц для i от 1 до N
```

```
    если A[i] > 0
```

```
        то S := S + A[i]
```

```
    все
```

```
кц
```

перебираем все
элементы массива

Задания

- «3»:** Заполнить массив из 10 элементов случайными числами в интервале $[-10, 10]$ и подсчитать сумму всех отрицательных элементов.
- «4»:** Заполнить массив из 10 элементов случайными числами в интервале $[0, 100]$ и подсчитать среднее значение всех элементов, которые < 50 .
- «5»:** Заполнить массив из 10 элементов случайными числами в интервале $[10, 12]$ и найти длину самой длинной последовательности стоящих рядом одинаковых элементов.

Пример:

Исходный массив:

10 10 11 12 12 12 10 11 11 12

Длина последовательности: 3

Поиск в массиве

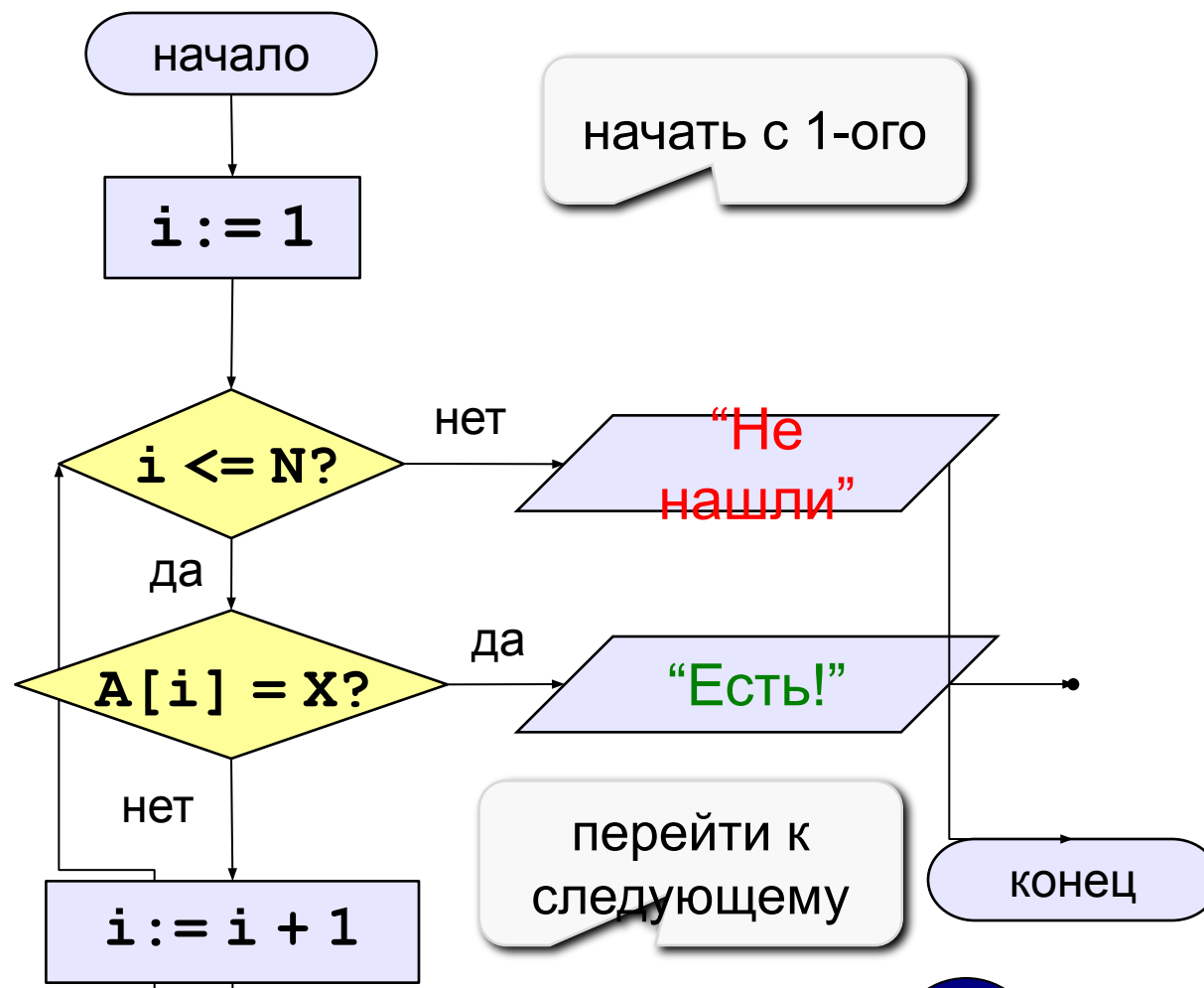
Задача – найти в массиве элемент, равный **X**, или установить, что его нет.

Пример: если в классе ученик с фамилией Пупкин?

Алгоритм:

- 1) начать с 1-ого элемента ($i := 1$)
- 2) если очередной элемент ($A[i]$) равен X , то закончить поиск
иначе перейти к следующему элементу:

Поиск элемента, равного X



Как найти номер?

Поиск в массиве

алг Поиск

нач

цел i , $N = 5$, X

целтаб $A[1:N]$

| здесь нужно заполнить массив

вывод нс, "Что ищем? "

ввод X

$i := 1$

нц пока $i \leq N$

если $A[i] = X$ то выход все

$i := i + 1$

кц

если $i \leq N$ то

вывод " $A["$, i , $"]="$, X

иначе

вывод "Не нашли!"

все

кон

выход из цикла

i – номер нужного
элемента в массиве



Как определить,
нашли или нет?

Задания

«3»: Заполнить массив из 10 элементов случайными числами в интервале $[10..20]$ и найти элемент, равный X.

Пример:

Исходный массив:

13 10 18 12 20 11 13 14 15 20

Что ищем? 20

A[5] = 20

«4»: Заполнить массив из 10 элементов случайными числами в интервале $[0..4]$ и вывести номера всех элементов, равных X.

Пример:

Исходный массив:

4 0 1 2 0 1 3 4 1 0

Что ищем? 0

A[2], A[5], A[10]

Задания

«5»: Заполнить массив из 10 элементов случайными числами в интервале $[0..4]$ и определить, есть ли в нем одинаковые соседние элементы.

Пример:

Исходный массив:

4 0 1 2 0 1 3 1 1 0

Ответ: **есть**

Программирование на алгоритмическом языке. Часть II

Тема 4. Символьные строки

Чем плох массив символов?

Это массив символов:

симтаб `s[1:10]`

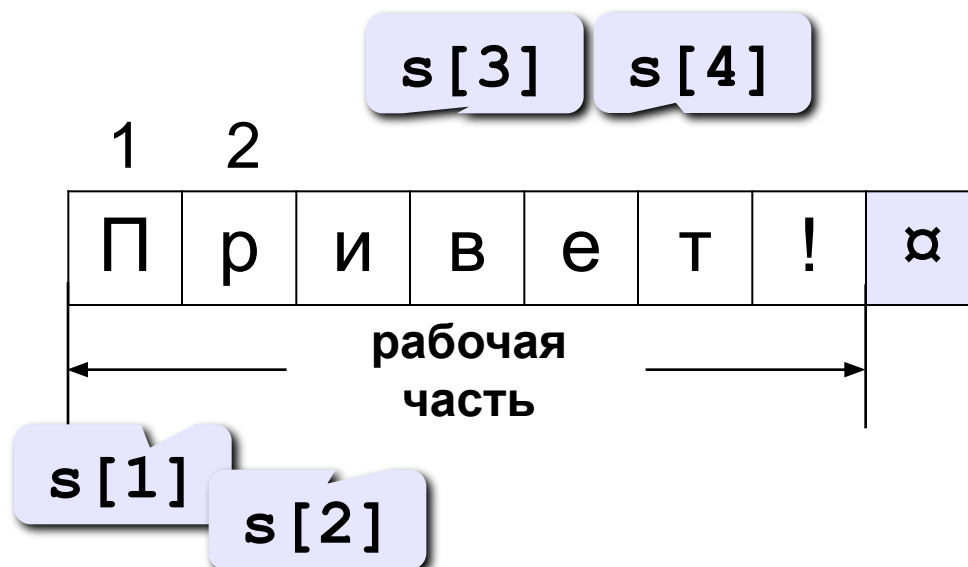
- каждый символ – отдельный объект;
- массив имеет длину, которая задана при объявлении

Что нужно:

- обрабатывать последовательность символов как единое целое
- строка должна иметь переменную длину

Символьные строки

Объявление строки: **лит s**



Длина строки:

цел n

n := длин (s)

Символьные строки

Задача: ввести строку с клавиатуры и заменить все буквы «а» на буквы «б».

алг **Замена а на б**

нач

лит s

вывод "Введите строку: "

ввод s

цел i

нц для i от 1 до **длин**(s)

если s[i] = "а"

то s[i] := "б"

длина строки

все

кц

вывод s

кон

Задания

«3»: Ввести символьную строку и заменить все буквы «а» на буквы «б», как заглавные, так и строчные.

Пример:

Введите строку:

ааббссААББСС

Результат:

ббббссББББСС

«4»: Ввести символьную строку и заменить все буквы «а» на буквы «б» и наоборот, как заглавные, так и строчные.

Пример:

Введите строку:

ааббссААББСС

Результат:

ббаассББААСС

Задания

«5»: Ввести символьную строку и проверить, является ли она **палиндромом** (палиндром читается одинаково в обоих направлениях).

Пример:

Введите строку:

АБВГДЕ

Результат:

Не палиндром.

Пример:

Введите строку:

КАЗАК

Результат:

Палиндром.

Операции со строками

```
лит s, s1, s2
```

Запись нового значения:

```
s := "Вася"
```

Объединение: добавить одну строку в конец другой.

```
s1 := "Привет"
```

```
s2 := "Вася"
```

```
s := s1 + ", " + s2 + "!" | "Привет, Вася!"
```

Подстрока: выделить часть строки в другую строку.

```
s := "123456789"
```

```
s1 := s[3:8] | "345678"
```

```
s2 := s1[2:4] | "456"
```

Удаление и вставка

Удаление части строки:

```
s := "123456789"
```

6 штук

```
s := удалить (s, 3, 6) ;
```

'12~~345678~~9'

'129'

строка
меняется!

с 3-его символа

Вставка в строку:

```
s := "123456789"
```

начиная с 3-его символа

```
s := вставить ("ABC", s, 3)
```

'12ABC3456789'

что
вставляем

куда
вставляем

```
s := вставить ("Q", s, 5)
```

'12ABQC3456789'

Поиск в строке (**найти**)

Поиск в строке:

```
s := "Здесь был Вася."  
n := найти("е", s)  
если n > 0 то  
    вывод "Буква е – это s[", n, "]"  
иначе вывод "Не нашли"  
все  
n := найти("Вася", s)  
s1 := s[n:n+3]
```

s[3]

цел n

3

n = 11

Вася

Особенности:

- функция **найти** возвращает номер символа, с которого начинается образец в строке
- если образец не найден, возвращается -1
- поиск с начала (находится **первое** слово)

Примеры

```
s := "Вася Петя Митя"  
n := найти ( "Петя", s )  
s := удалить (s, n, 4)  
s := вставить ("Куку", s, n)
```

6

'Вася Митя'

'Вася Куку Митя'

```
s := "Вася Петя Митя"  
n := длин ( s )  
s1 := s[1:4]  
s2 := s[11:14]  
s3 := s[6:9]  
s := s3 + s1 + s2  
n := длин ( s )
```

14

'Вася'

'Митя'

'Петя'

'ПетяВасяМитя'

12

Пример решения задачи

Задача: Ввести имя, отчество и фамилию. Преобразовать их к формату «фамилия-инициалы».

Пример:

Введите имя, отчество и фамилию:

Василий Алибабаевич Хрюндиков

Результат:

Хрюндиков В.А.

Алгоритм:

- найти первый пробел и выделить имя
- удалить имя с пробелом из основной строки
- найти первый пробел и выделить отчество
- удалить отчество с пробелом из основной строки
- «сцепить» фамилию, первые буквы имени и фамилии, точки, пробелы...

Программа

использовать Строки

алг ФИО

нач

лит s, имя, отчество

цел n

вывод "Введите имя, отчество и фамилию"

ввод s

n := найти (" ", s);

имя := s[1:n-1] | вырезать имя

s := удалить (s, 1, n)

n := найти (" ", s)

отчество := s[1:n-1] | вырезать отчество

s := удалить (s, 1, n) | осталась фамилия

s := s + " " + имя[1] + "." + отчество[1] + "."

вывод s

кон

Задания

«3»: Ввести в одну строку фамилию, имя и отчество, разделив их пробелом. Вывести инициалы и фамилию.

Пример:

Введите фамилию, имя и отчество:

Иванов Петр Семёнович

Результат:

П.С. Иванов

«4»: Ввести имя файла (возможно, без расширения) и изменить его расширение на «.exe».

Пример:

Введите имя файла:

qqq

Результат:

qqq.exe

Введите имя файла:

qqq.com

Результат:

qqq.exe

Задания

«5»: Ввести путь к файлу и «разобрать» его, выводя каждую вложенную папку с новой строки

Пример:

Введите путь к файлу:

C:\Мои документы\10-Б\Вася\qq.exe

Результат:

C:

Мои документы

10-Б

Вася

qq.exe

Задачи на обработку строк

Задача: с клавиатуры вводится символьная строка, представляющая собой сумму двух целых чисел, например:

12+35

Вычислить эту сумму:

12+35=47

Алгоритм:

- 1) найти знак «+»
- 2) выделить числа слева и справа в отдельные строки
- 3) перевести строки в числа
- 4) сложить
- 5) вывести результат

Преобразования «строка»-«число»

Из строки в число:

s := "123"

да или нет

N := лит_в_цел(s, ОК) | N = 123

если не ОК то вывод "Ошибка!" все

s := "123.456";

X := лит_в_вещ(s, ОК) | X = 123.456

если не ОК то вывод "Ошибка!" все

цел N, вещ X,
лит s, лог ОК

Из числа в строку:

N := 123

s := цел_в_лит(N) | "123"

X := 123.456

s := вещ_в_лит(X) | "123.456"

Программа

использовать **Строки**

алг **Калькулятор** **слагаемые-строки**

нач

лит s, s1, s2

**слагаемые-
числа**

цел n, n1, n2, sum

лог ОК

сумма

вывод "Введите выражение (сумму двух чисел): "

ввод s

n := **найти** ("+", s)

s1 := s[1:n-1]

s2 := s[n+1:**длин**(s)]

слагаемые-строки

n1 := **лит_в_цел**(s1, ОК)

**слагаемые-
числа**

n2 := **лит_в_цел**(s2, ОК)

sum := n1 + n2

вывод n1, "+", n2, "=", sum

кон

Задания

«3»: Ввести арифметическое выражение: разность двух чисел. Вычислить эту разность.

Пример:

25-12

Ответ: 13

«4»: Ввести арифметическое выражение: сумму трёх чисел. Вычислить эту сумму.

Пример:

25+12+34

Ответ: 71

Задания

«5»: Ввести арифметическое выражение с тремя числами, в котором можно использовать сложение и вычитание. Вычислить это выражение.

Пример:

25+12+34

Ответ: 71

Пример:

25+12-34

Ответ: 3

Пример:

25-12+34

Ответ: 47

Пример:

25-12-34

Ответ: -21

Задания

«6»: Ввести арифметическое выражение с тремя числами, в котором можно использовать сложение, вычитание и умножение. Вычислить это выражение.

Пример:

$25+12*3$

Ответ: 61

Пример:

$25*2-34$

Ответ: 16

Пример:

$25-12+34$

Ответ: 47

Пример:

$25*2*3$

Ответ: 150

Программирование на алгоритмическом языке. Часть II

Тема 5. Матрицы

Матрицы

Задача: запомнить положение фигур на шахматной доске.



1



2



3



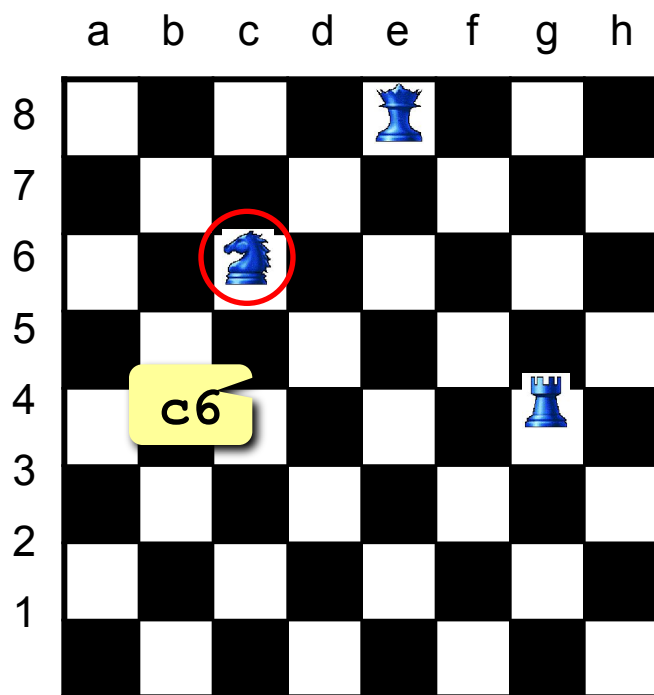
4



5



6



	1	2	3	4	5	6	7	8
8	0	0	0	0	2	0	0	0
7	0	0	0	0	0	0	0	0
6	0	0	3	0	0	0	0	0
5	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	4	0
3	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0

A[6,3]

Матрицы

Матрица – это прямоугольная таблица чисел (или других элементов одного типа).

Матрица – это массив, в котором каждый элемент имеет два индекса (номер строки и номер столбца).

А

	1	2	3	4	5
1	1	4	7	3	6
2	2	-5	0	15	10
3	8	9	11	12	20

столбец 3

строка 2

ячейка **А**[3, 4]

Матрицы

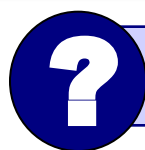
Объявление:

```
цел N = 3, M = 4
целтаб A[1:N, 1:M]
целтаб B[-3:0, -8:M]
```

Ввод с клавиатуры:

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

i	j	
1	1	2
1	2	5
1	3	4
	=	4
3	4	5
	=	4



Если переставить циклы?

Матрицы

Заполнение случайными числами

```
нц для i от 1 до N
  нц для j от 1 до M
    A[i,j] := irand(20,80)
  кц
кц
```

цикл по строкам

цикл по столбцам

Вывод на экран

вывод строки

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

21	65	41	43
56	71	32	47
31	45	22	23



Что плохо?



Если переставить циклы?

Обработка всех элементов матрицы

Задача: заполнить матрицу из 3 строк и 4 столбцов случайными числами и вывести ее на экран. Найти сумму элементов матрицы.

алг Сумма

нач

цел $N = 3$, $M = 4$, i , j

целтаб $A[1:N, 1:M]$

| заполнение матрицы и вывод на экран

цел $S = 0$

нц для i от 1 до N

нц для j от 1 до M

$S := S + A[i, j]$

кц

кц

вывод "Сумма элементов матрицы: ", S

кон

Задания

Заполнить матрицу из 8 строк и 5 столбцов случайными числами в интервале $[10,90]$ и вывести ее на экран. Затем ...

«3»: Удвоить все элементы матрицы и вывести её на экран.

«4»: Найти минимальный и максимальный элементы в матрице их номера. Формат вывода:

Минимальный элемент $A[3,4]=11$

Максимальный элемент $A[2,2]=89$

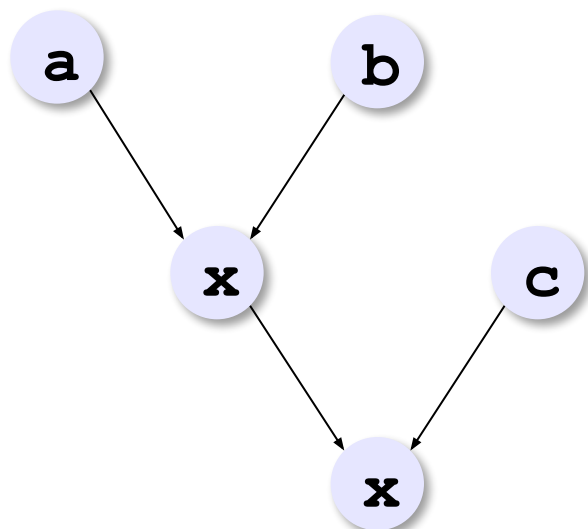
«5»: Вывести на экран строку, сумма элементов которой максимальна. Формат вывода:

Строка 2: 13 25 18 79 38

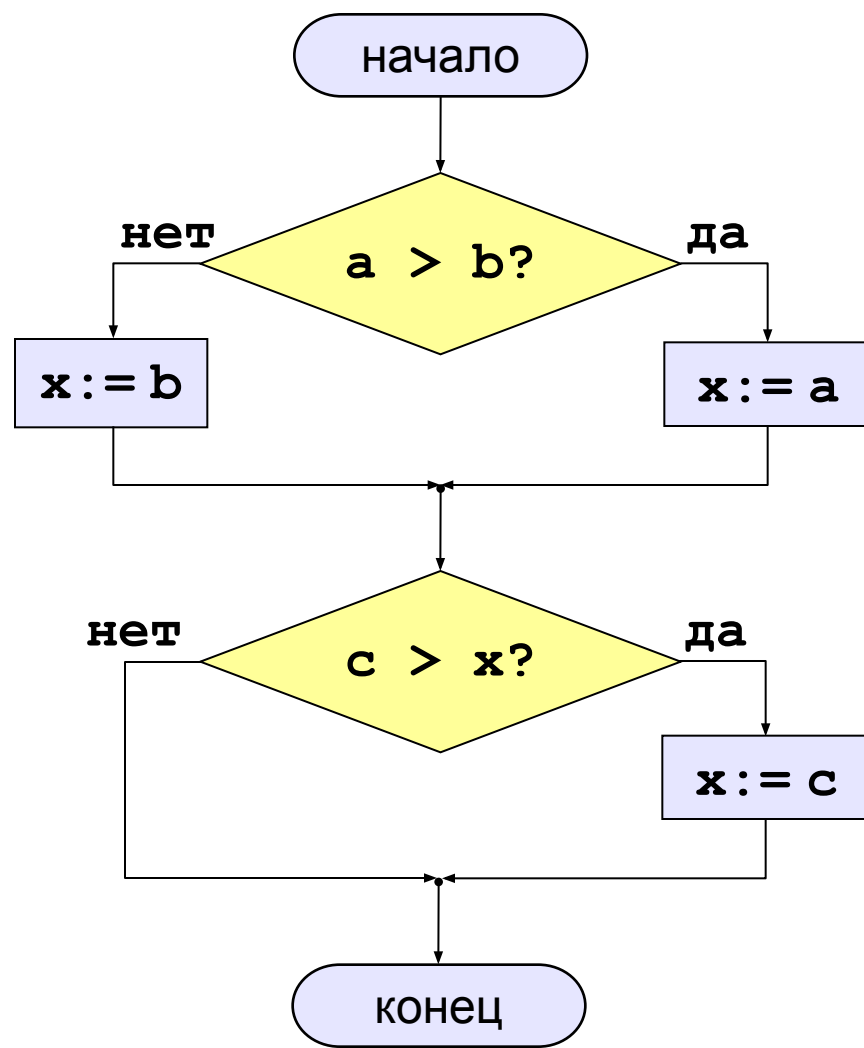
Программирование на алгоритмическом языке

Тема 7. Алгоритмы-функции

Максимум из трех



Где ответ?



Максимум из трех

результат – целое число

алг **цел** Макс3 (**цел** a, b, c)

нач

цел x

если a > b

то x := a

иначе x := b

все

если c > x то x := c все

знач := x

кон

внутренняя (локальная)
переменная

результат (значение)
функции

Максимум из трех

алг **Максимум**

нач

цел Z, X, C, M

вывод "Введите три числа "

ввод Z, X, C

M := **Макс3** (Z, X, C)

вывод "Максимум: ", M

кон

вызов функции

алг **цел** **Макс3** (**цел** a, b, c)

нач

...

кон

имена параметров
могут быть любые!

Максимум из пяти



Как с помощью этой функции найти максимум из пяти чисел?

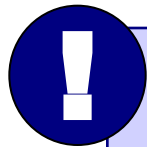
```
алг Максимум  
нач  
    цел Z, X, C, V, B, M  
    ввод Z, X, C, V, B  
    ???  
    вывод "Максимум: ", M  
кон
```

```
алг цел Макс3 (цел a, b, c)  
нач  
    . . .  
кон
```

НОД трёх чисел

Задача: ввести три числа и найти наибольший общий делитель (НОД). Используйте формулу

$$\text{НОД}(a, b, c) = \text{НОД}(\text{НОД}(a, b), c)$$



Нужно два раза применить один и тот же алгоритм вычисления НОД двух чисел.

Решение: построить вспомогательный алгоритм для вычисления НОД двух чисел и применить (вызвать) его два раза.

Общий вид программы

алг **НОД** трёх чисел

нач

цел a, b, c, N

вывод "Введите три числа", $нс$

ввод a, b, c

???

вывод N

кон

основной
алгоритм

алг **цел** **НОД**(**цел** x, y)

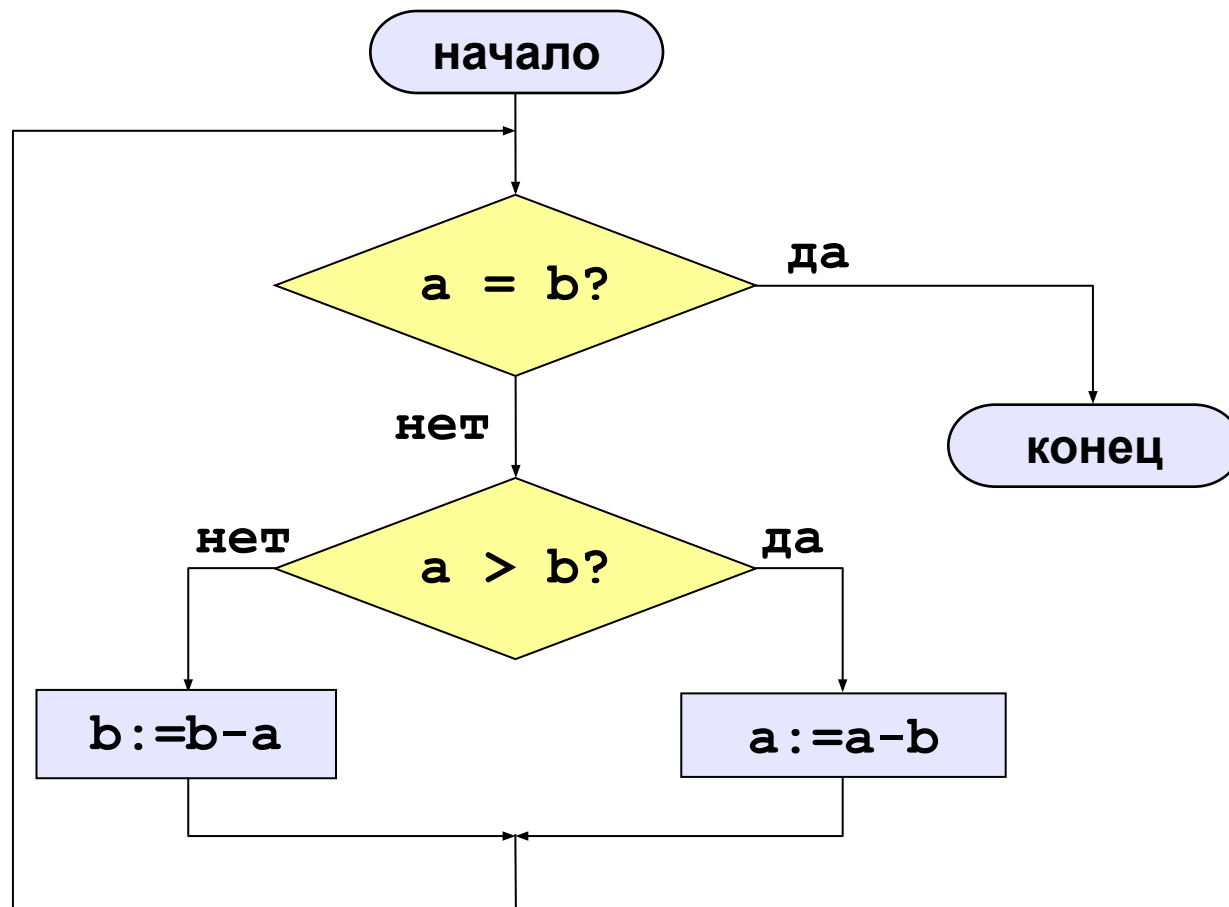
нач

| здесь нужно вычислить $НОД(x, y)$

кон

вспомогательный
алгоритм

Блок-схема алгоритма Евклида



Алгоритм Евклида

```
алг цел НОД(цел x, y)
нач
    цел a, b
    a := x; b := y    | копии параметров
    нц пока a <> b
        если a > b
            то a := a - b
            иначе b := b - a
        все
    кц
    знач := a    | результат функции
кон
```

Рекурсивные функции

Факториал:

$$N! = \begin{cases} 1, & \text{если } N = 1, \\ N \cdot (N-1)!, & \text{если } N > 1. \end{cases}$$

$$1! = 1, \quad 2! = 2 \cdot 1! = 2 \cdot 1, \quad 3! = 3 \cdot 2! = 3 \cdot 2 \cdot 1, \quad 4! = 4 \cdot 3! = 4 \cdot 3 \cdot 2 \cdot 1$$

```
алг цел факт (цел N)
нач
  если N < 2
    то знач := 1
    иначе знач := N * факт (N - 1)
  все
кон
```

```
алг цел факт (цел N)
нач
  цел i
  знач := 1
  нц для i от 2 до N
    знач := знач * i
  кц
кон
```



Алгоритм без рекурсии, как правило, работает быстрее и требует меньше памяти.

Задания

«3»: Составить функцию, которая определяет **НАИМЕНЬШЕЕ** из трёх чисел и привести пример ее использования.

Пример:

Введите три числа:

28 15 10

Наименьшее число 10.

«4»: Составить функцию, которая вычисляет **НАИМЕНЬШЕЕ** из четырёх чисел и привести пример ее использования.

Пример:

Введите четыре числа:

10 20 5 25

Наименьшее число 5.

Задания

«5»: Составить функцию, которая определяет сумму всех чисел от 1 до N и привести пример ее использования.

Пример:

Введите число:

100

Сумма чисел от 1 до 100 = 5050

Логические функции

Задача: составить функцию, которая определяет, верно ли, что заданное число – четное.

Особенности:

- ответ – логическое значение (**да** или **нет**)
- результат функции можно использовать как логическую величину в условиях (**если**, **пока**)

Алгоритм: если число делится на 2, оно четное.

```
если mod(N, 2) = 0
    то      | число N четное
иначе     | число N нечетное
все
```

Логические функции

логическое значение (**да** или **нет**)

алг **лог** Четное

нач

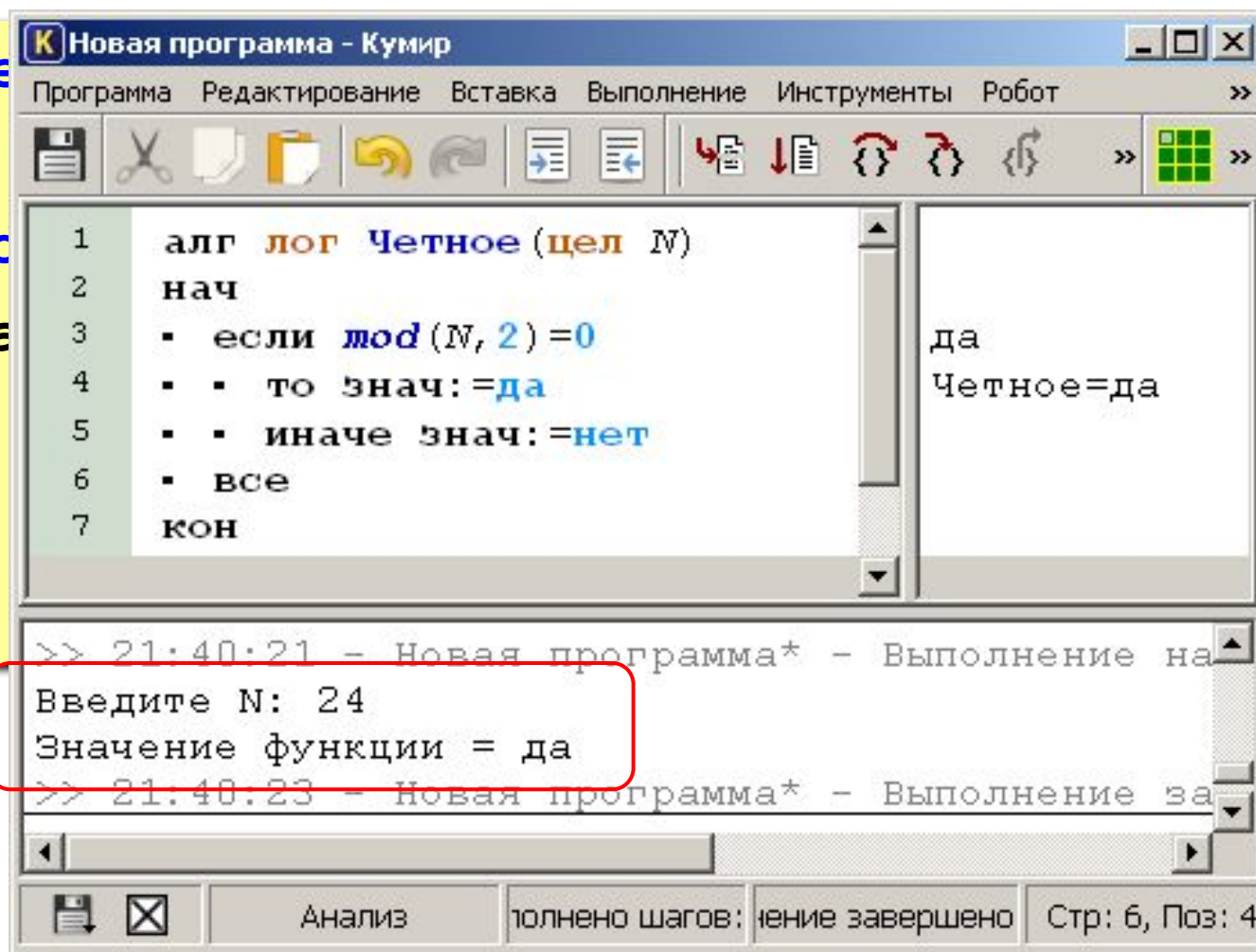
если **то**

то знач

иначе

все

кон



Логические функции

алг Проверка четности

нач

цел x

вывод "Введите целое число: "

ввод x

если Четное (x)

то вывод "Число четное."

иначе вывод "Число нечетное."

все

кон

алг лог Четное (цел N)

нач

...

кон

Задания

«3»: Составить функцию, которая определяет, верно ли, что число оканчивается на 0.

Пример:

Введите число: Введите число:

170

237

Верно .

Неверно .

«4»: Составить функцию, которая определяет, верно ли, что в числе вторая цифра с конца больше 6.

Пример:

Введите число: Введите число:

178

237

Верно .

Неверно .

Задания

«5»: Составить функцию, которая определяет, верно ли, что переданное ей число – простое (делится только на само себя и на единицу).

Пример:

Введите число: Введите число:

29

28

Простое число. Составное число.

Программирование на алгоритмическом языке. Часть II

Тема 8. Построение графиков функций

Построение графиков функций

Задача: построить график функции $y = x^2$ на интервале от -2 до 2.

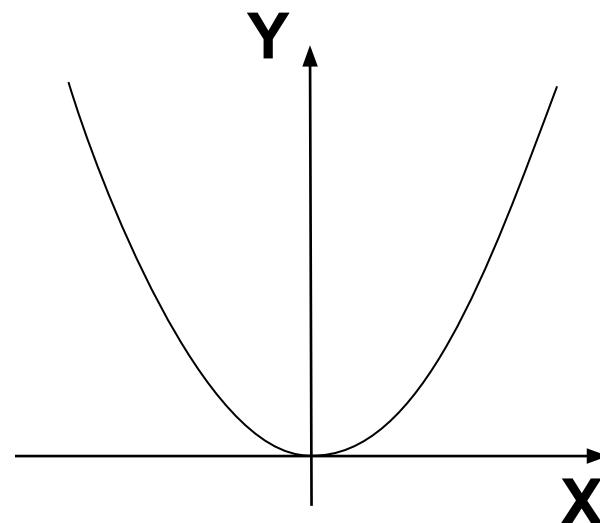
Анализ:

максимальное значение

$$y_{\max} = 4 \quad \text{при} \quad x = \pm 2$$

минимальное значение

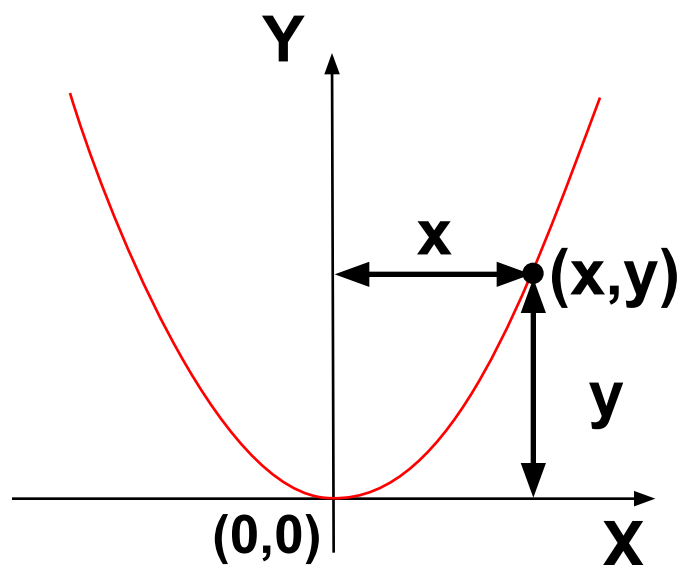
$$y_{\min} = 0 \quad \text{при} \quad x = 0$$



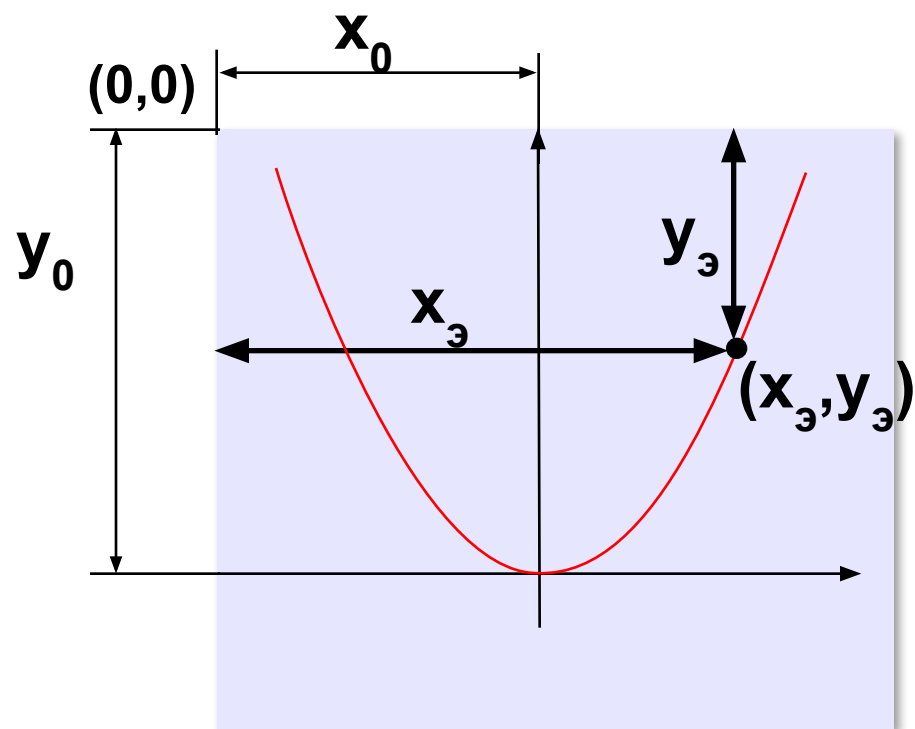
Проблема: функция задана в математической системе координат, строить надо на экране, указывая координаты в пикселях.

Преобразование координат

Математическая
система координат



Экранная система
координат (пиксели)



k – масштаб (длина
изображения единичного
отрезка на экране)

$$\begin{aligned}x_э &= \\y_э &= \end{aligned}$$

Программа

использовать **Рисователь**

алг **График**

нач

цел x0=150, y0=250, k=50, xe, ye

вещ xmin=-2, xmax=2, x, y, h

в точку (0, y0)

линия в точку (x0+150, y0)

в точку (x0, 0)

линия в точку (x0, y0+20)

x:=xmin; h:=0.02

нц пока x <= xmax

y:= x*x

xe:= x0 + int(k*x)

ye:= y0 - int(k*y)

пиксель (xe, ye, "красный")

x:=x+h

кц

кон

на экране

h – шаг изменения x

оси координат

цикл
построения
графика



Что плохо?

Как соединить точки?

Алгоритм:

Если первая точка
перейти в точку $(x_э, y_э)$
иначе
линия в точку $(x_э, y_э)$

выбор
варианта
действий

Программа:

лог переменная
лог первая

...

первая := да

нц пока $x \leq x_{\max}$

...

если первая то

в точку $(x_э, y_э)$

первая := нет

иначе

линия в точку $(x_э, y_э)$

все

кц

логическая
переменная

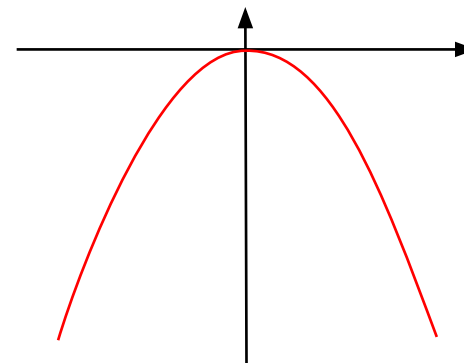
начальное значение

Задания

«3»: Построить график функции

$$y = -x^2$$

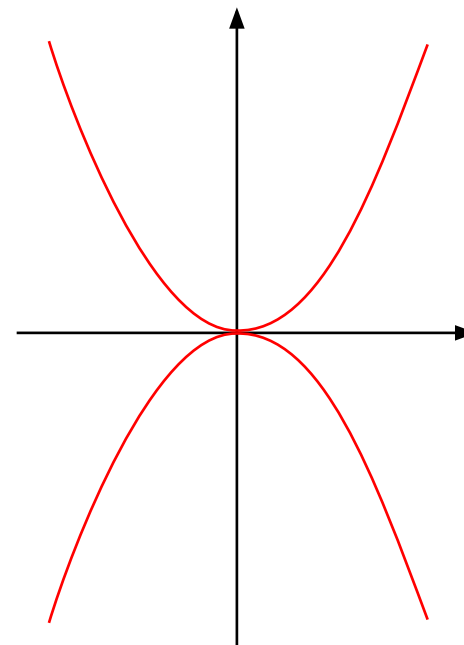
на интервале $[-2, 2]$.



«4»: Построить графики функций

$$y = x^2 \quad y = -x^2$$

на интервале $[-2, 2]$.

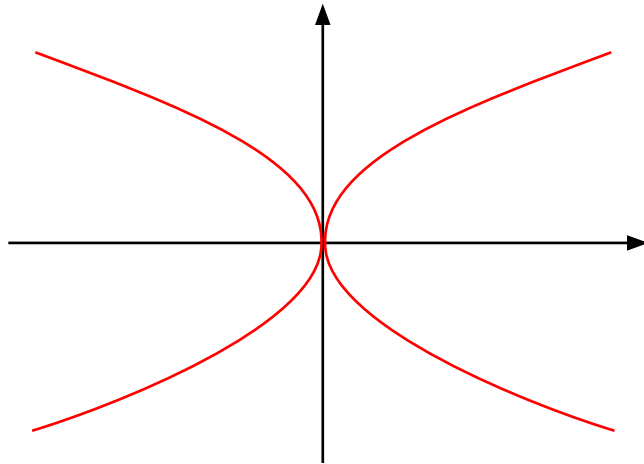


Задания

«5»: Построить графики функций

$$x = y^2 \quad x = -y^2$$

на интервале $[-2, 2]$. Соединить точки линиями.



Конец фильма

ПОЛЯКОВ Константин Юрьевич
д.т.н., учитель информатики высшей
категории,
ГОУ СОШ № 163, г. Санкт-Петербург
kpolyakov@mail.ru