

Программирование на алгоритмическом языке (7 класс)

- | | |
|--|---|
| 1. <u>Введение</u> | 6. <u>Графика</u> |
| 2. <u>Ветвления</u> | 7. <u>Вспомогательные алгоритмы</u> |
| 3. <u>Сложные условия</u> | 8. <u>Анимация</u> |
| 4. <u>Циклы</u> | 9. <u>Случайные числа</u> |
| 5. <u>Циклы с переменной</u> | |

Программирование на алгоритмическом языке

Тема 1. Введение

Алгоритм

Алгоритм – это четко определенный план решения задачи для исполнителя.

Свойства алгоритма

- **дискретность**: состоит из отдельных шагов (команд)
- **понятность**: должен включать только команды, известные исполнителю
- **конечность**: позволяет получить решение задачи за конечное число шагов
- **определенность**: при одинаковых исходных данных всегда выдает один и тот же результат
- **массовость**: может применяться при различных исходных данных

Программа

Программа – это

- алгоритм, записанный на каком-либо языке программирования
- набор команд для исполнителя

Команда – это описание действий, которые должен выполнить исполнитель.

- откуда взять исходные данные?
- что нужно с ними сделать?

Простейшая программа

название алгоритма

алг **Первый**

нач | *начало алгоритма*

кон | *конец алгоритма*

комментарии после |
не обрабатываются



Что делает эта программа?

Вывод текста на экран

алг **Вывод на экран**

нач

▶ **ВЫВОД** "2+" 

▶ **ВЫВОД** "2=?" , **НС**

▶ **ВЫВОД** "Ответ: 4"

КОН

Протокол:

2+

Ответ: 4

Задания

«4»: Вывести на экран текст «лесенкой»

Вася

пошел

гулять

«5»: Вывести на экран рисунок из букв

```
  ж
 жжж
 жжжжж
 жжжжжжж
 жж жж
 жжжжж
```

Переменные

Задача. Ввести с клавиатуры два числа и найти их сумму.

Протокол:

Введите два целых числа

компьютер

25 30

пользователь

25+30=55

компьютер считает сам!



1. Как ввести числа в память?
2. Где хранить введенные числа?
3. Как вычислить?
4. Как вывести результат?

Программа

алг Сумма

нач

| ввести два числа

| вычислить их сумму

| вывести сумму на экран

кон

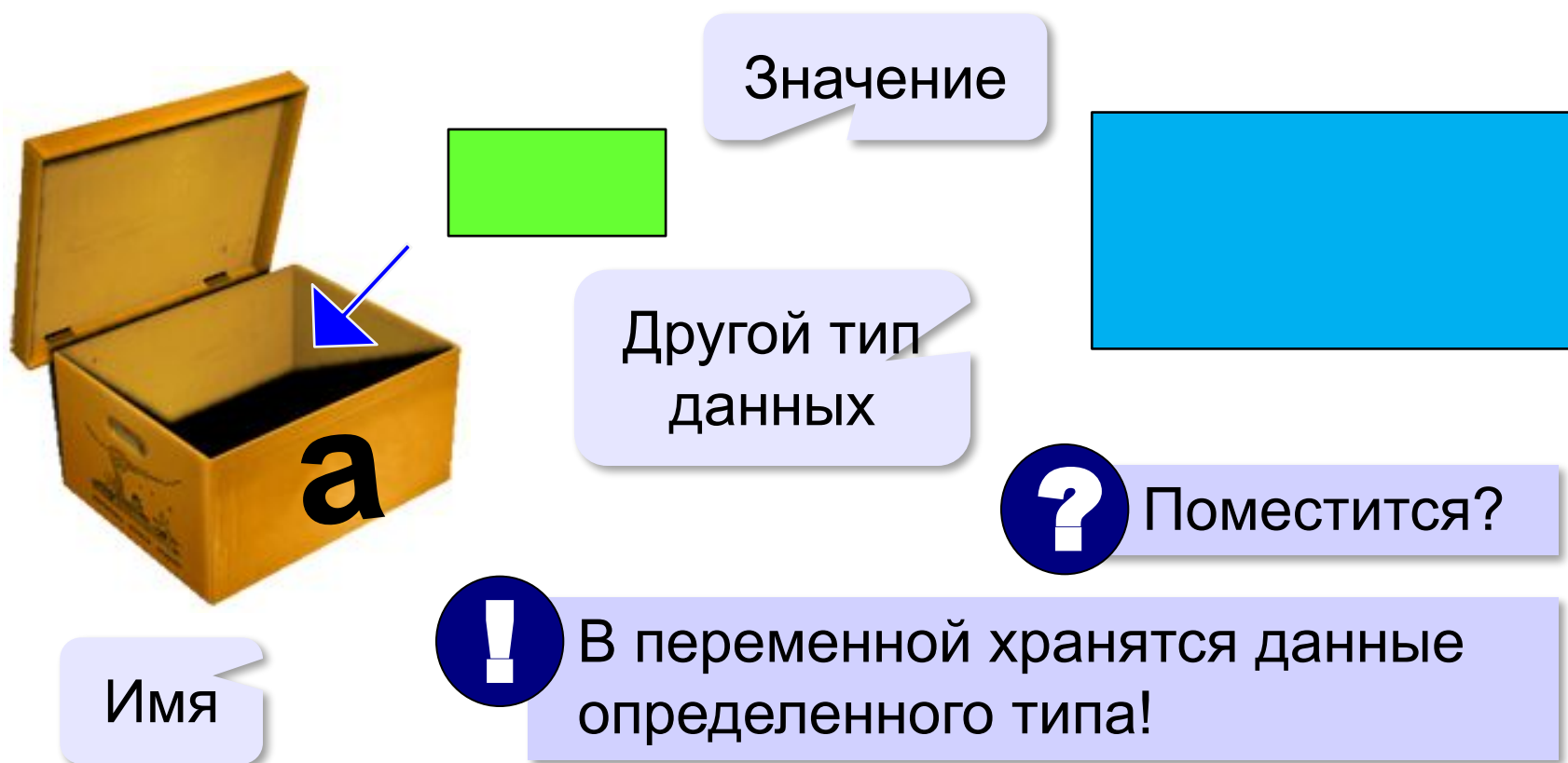
Псевдокод – алгоритм на русском языке с элементами языка программирования.



Компьютер не может исполнить псевдокод!

Переменные

Переменная — это величина, имеющая имя, тип и значение. Значение переменной можно изменять во время работы программы.



Имена переменных

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.~~

Какие имена правильные?

AXby R&B 4Wheel Вася “PesBarbos”

TU154 [QuQu] _ABBA A+B

Объявление переменных

Типы переменных:

- цел | целая
- вещ | вещественная
- и другие...

Объявление переменных:

выделение
места в памяти

тип — целые

список имен
переменных

цел **a, b, c**

Как записать значение в переменную?

Оператор
присваивания

a := 5

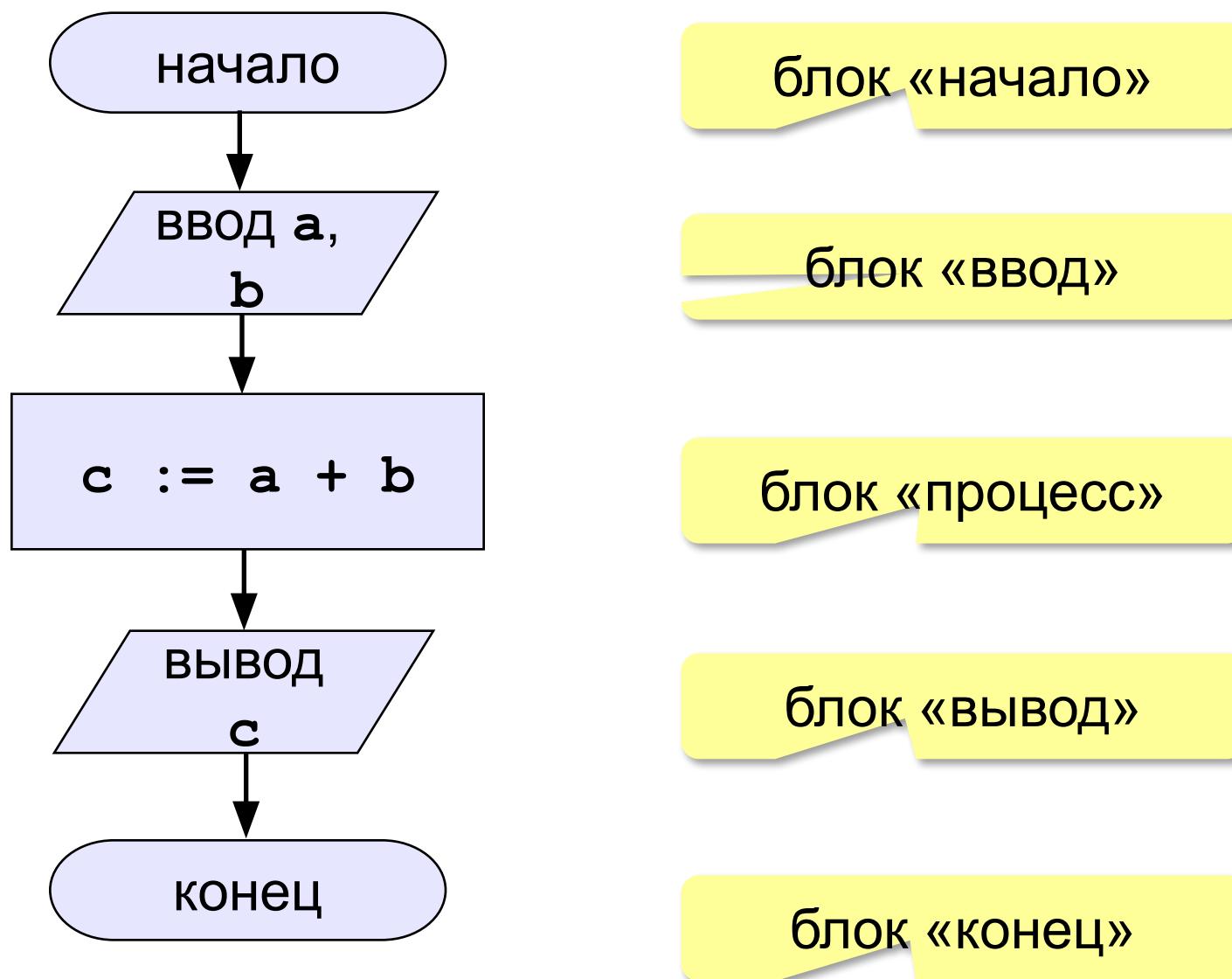


При записи нового
значения старое
стирается!

Оператор – это команда языка программирования (инструкция).

Оператор присваивания – это команда для записи нового значения в переменную.

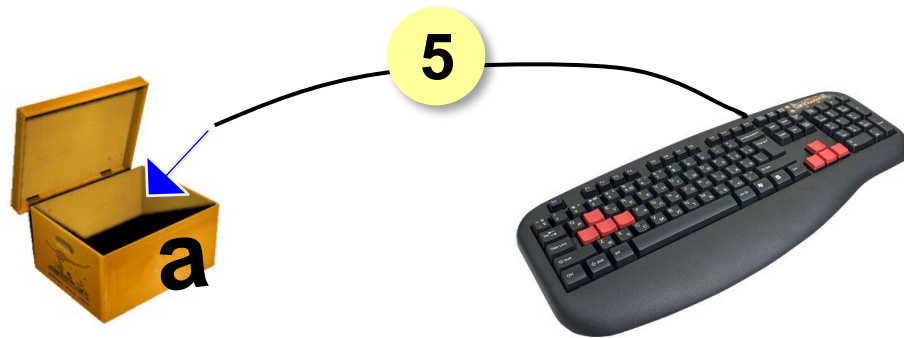
Блок-схема линейного алгоритма



Как ввести значение с клавиатуры?

Оператор
ввода

ВВОД а



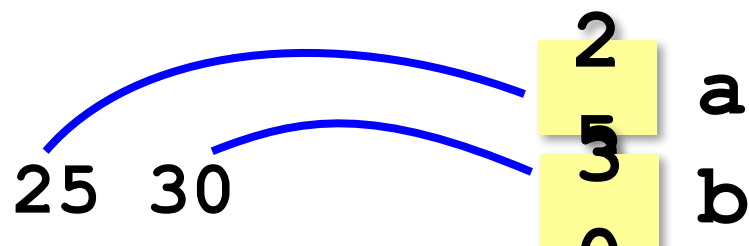
1. Программа ждет, пока пользователь введет значение и нажмет *Enter*.
2. Введенное значение записывается в переменную **a**.

Ввод значений двух переменных

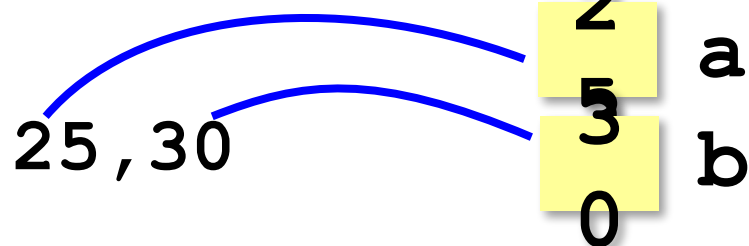
ввод a, b

Ввод значений двух переменных.

через пробел:



через запятую:



Изменение значения переменной

Пример:

алг **Тест**

нач

цел a, b

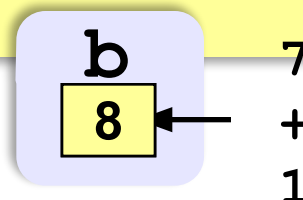
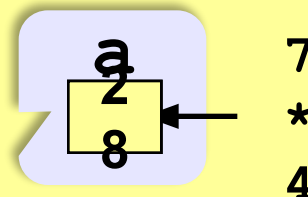
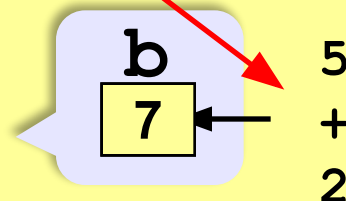
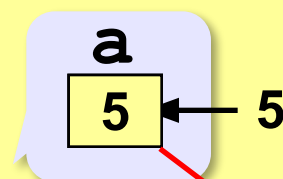
a := 5

b := a + 2

a := (a + 2) * (b - 3)

b := b + 1

кон



Арифметические операции

+ сложение

− вычитание

***** умножение

/ деление

цел a, b

a := 7 * 3 − 4

a := a * 5

Вывод данных

вывод **a**

| вывод значения
| переменной **a**

вывод **a, nc**

| 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**

Задача: сложение чисел

Задача. Ввести два целых числа и вывести на экран их сумму.

Простое решение:

```
алг Сумма
нач
    цел a, b, c
    ввод a, b
    c := a + b
    вывод c
кон
```



Что плохо?

Полное решение

алг Сумма

нач

цел a, b, c

вывод "Введите два целых числа"

ввод a, b

$c := a + b$

вывод $a, "+", b, "=", c$

кон

подсказка

Протокол:

Введите два целых числа

25 30

25+30=55

компьютер

пользователь

Задания

«3»: Ввести три числа, найти их сумму.

Пример:

Введите три числа:

4 5 7

$$4+5+7=16$$

«4»: Ввести три числа, найти их сумму и произведение.

Пример:

Введите три числа:

4 5 7

$$4+5+7=16$$

$$4*5*7=140$$

Задания

«5»: Ввести три числа, найти их сумму, произведение и среднее арифметическое.

Пример:

Введите три числа:

4 5 7

$$4+5+7=16$$

$$4*5*7=140$$

$$(4+5+7) / 3 = 5.333333$$

Какие операторы неправильные?

алг Ошибки

нач

цел а, b

вещ x, y

а := 5

10 := x

y := 7,8

b := 2.5

x := 2*(a + y)

а := b + x

кон

имя переменной должно
быть слева от знака :=

целая и дробная часть
отделяются **точкой**

нельзя записывать
вещественное значение в
целую переменную

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)
```

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))

Ручная прокрутка программы

алг **Тест**

нач

цел a, b

a := 5

b := a + 2

a := (a + 2) * (b - 3)

b := **div**(a, 5)

a := **mod**(a, b)

a := a + 1

b := **mod**(a+14, 7)

кон

a	b
?	?

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, ">"

<1<>3>

цел $a = 1, b = 3$

вывод a, "+", b, "=?"

1+3=?

Программирование на алгоритмическом языке

Тема 2. Ветвления

Разветвляющиеся алгоритмы

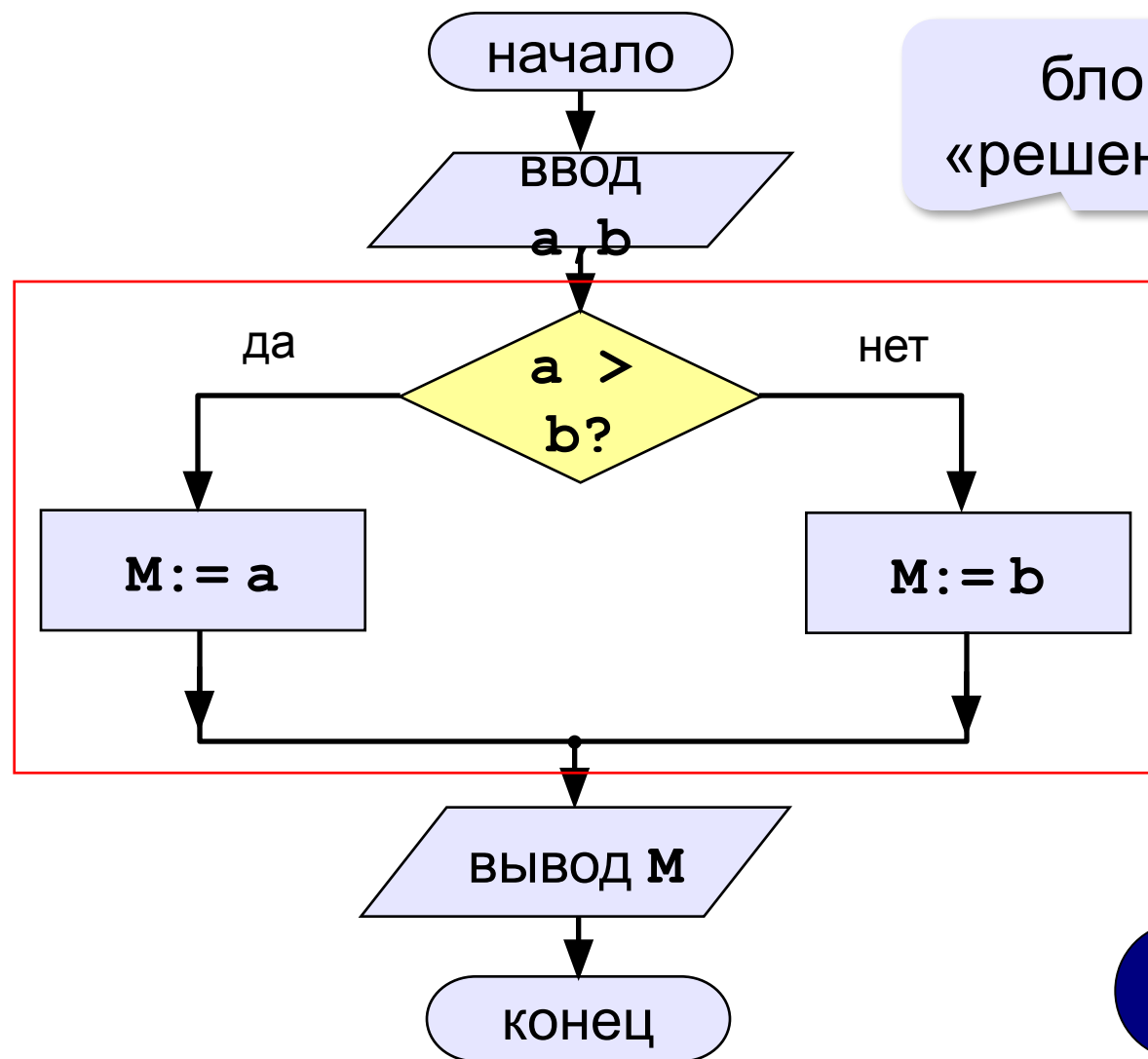
Задача. Ввести два целых числа и вывести на экран наибольшее из них.

Идея решения: надо вывести на экран первое число, если оно больше второго, или второе, если оно больше первого.

Особенность: действия исполнителя зависят от некоторых условий (***если ... иначе ...***).

Алгоритмы, в которых последовательность шагов зависит от выполнения некоторых условий, называются **разветвляющимися**.

Вариант 1. Блок-схема



блок
«решение»

полная
форма
ветвления



Если $a = b$?

Вариант 1. Программа

алг **Максимум**

нач

цел a, b, M

вывод **"Введите два целых числа"**, **нс**

ввод a, b

если $a > b$ то

$M := a$

иначе

$M := b$

все

вывод **"Наибольшее число "**, M

кон

полная форма
условного
оператора

Условный оператор

если **условие** то

| что делать, если условие верно

иначе

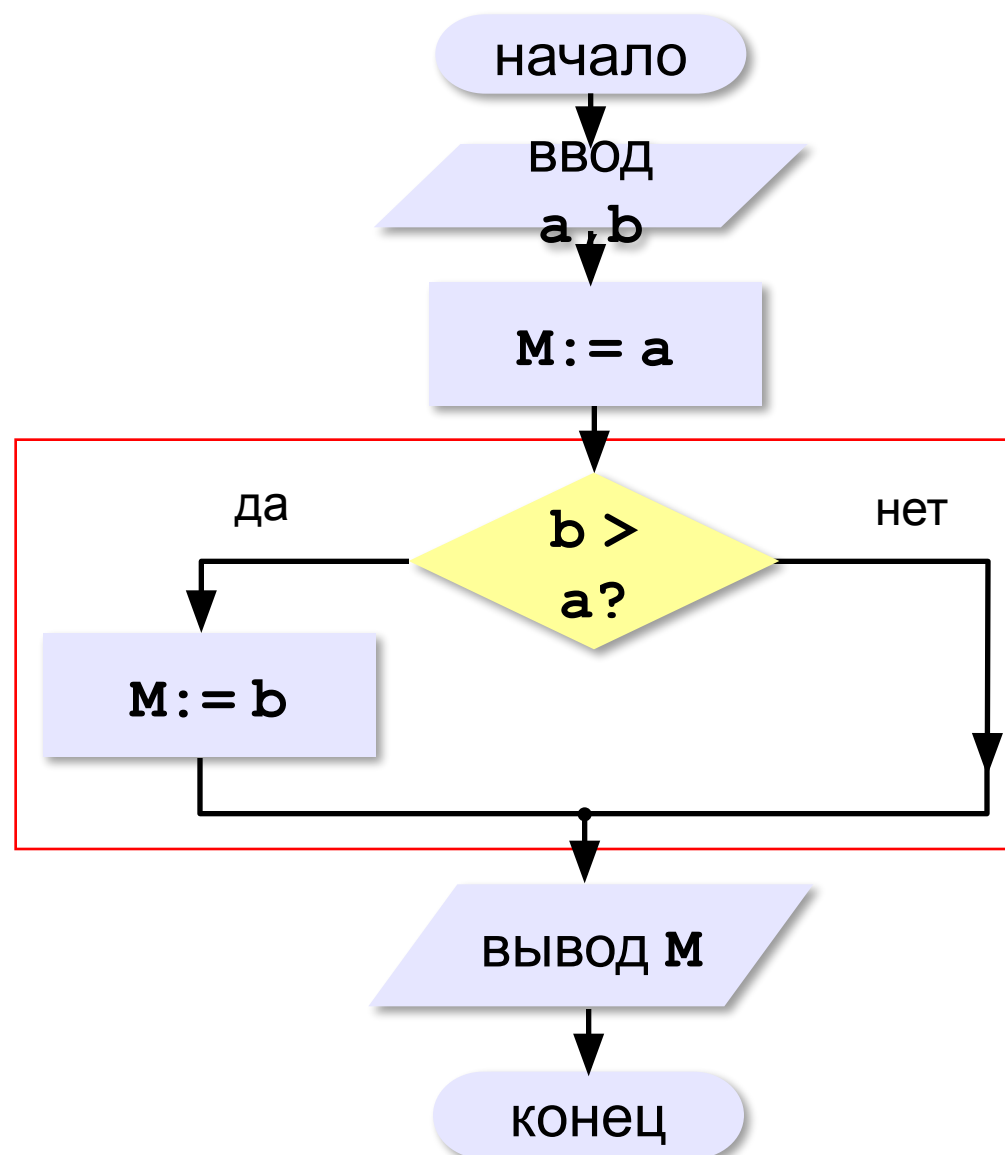
| что делать, если условие неверно

все



Вторая часть (иначе) может отсутствовать!

Вариант 2. Блок-схема



неполная
форма
ветвления

Вариант 2. Программа

алг **Максимум 2**

нач

цел a, b, M

вывод "Введите два целых числа", **нс**

ввод a, b

$M := a$

если $b > a$ **то**

$M := b$

все

вывод "Наибольшее число ", M

кон

неполная
форма
условного
оператора

Вариант 26. Программа

алг **Максимум 26**

нач

цел a, b, M

вывод "Введите два целых числа", **нс**

ввод a, b

$M := b$

если $a > b$ **то**

$M := a$

все

вывод "Наибольшее число ", M

кон

Задания

«3»: Ввести два числа и вывести их в порядке возрастания.

Пример:

Введите два числа:

15 9

Ответ: 9 15

«4»: Ввести три числа и найти наибольшее из них.

Пример:

Введите три числа:

4 15 9

Наибольшее число 15

Задания

«5»: Ввести пять чисел и найти наибольшее из них.

Пример:

Введите пять чисел:

4 15 9 56 4

Наибольшее число 56

Программирование на алгоритмическом языке

Тема 3. Сложные условия

Сложные условия

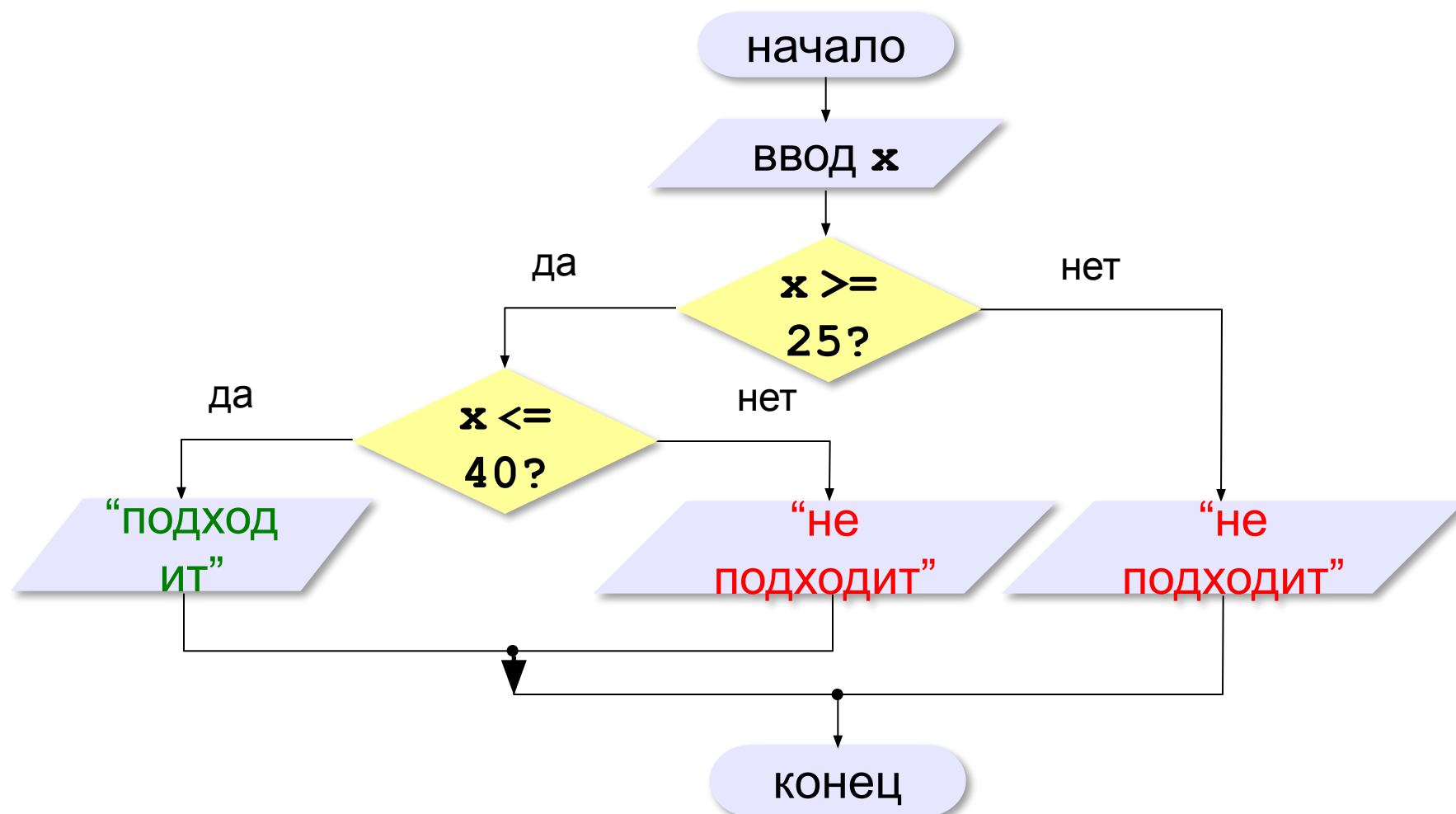
Задача. Фирма набирает сотрудников от 25 до 40 лет включительно. Ввести возраст человека и определить, подходит ли он фирме (вывести ответ «подходит» или «не подходит»).

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 \geq 25 то

 если x \leq 40 то

 вывод "Подходит!"

 иначе

 вывод "Не подходит."

 все

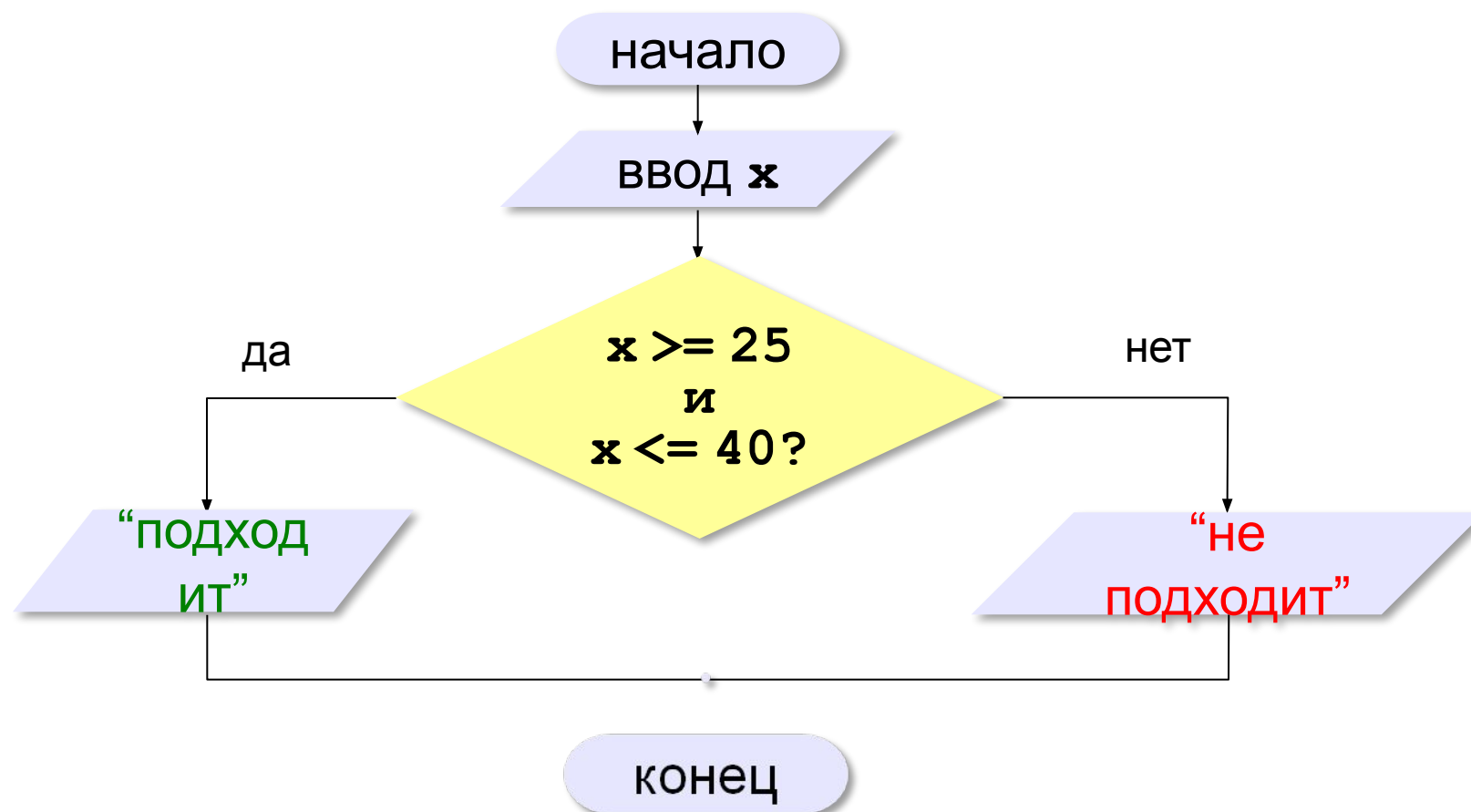
иначе

 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

кон

сложное
условие

Сложные условия

Простые условия (отношения)

равно

< <= > >= = <>

не равно

Сложное условие – это условие, состоящее из нескольких простых условий (отношений), связанных с помощью **логических операций**:

- **И** – одновременное выполнение условий

$x \geq 25 \text{ И } x \leq 40$

- **ИЛИ** – выполнение хотя бы одного из условий

$x \leq 25 \text{ ИЛИ } x \geq 40$

- **НЕ** – отрицание, обратное условие

$\text{НЕ } (x > 25) \quad \Leftrightarrow$

$x \leq 25$

Сложные условия

Порядок выполнения (приоритет = старшинство)

- выражения в скобках
- НЕ
- <, <=, >, >=, =, <>
- И
- ИЛИ

Пример

2 1 6 3 5 4
если не (a > 2) или c <> 5 и b < a то
...
все

Сложные условия

Истинно или ложно при $a := 2; b := 3; c := 4;$

не $(a > b)$

Да

$a < b$ **и** $b < c$

Да

$a > c$ **или** $b > c$

Нет

$a < b$ **и** $b > c$

Нет

$a > c$ **и** $b > d$

Нет

не $(a \geq b)$ **или** $c = d$

Да

$a \geq b$ **или не** $(c < b)$

Да

$a > c$ **или** $b > c$ **или** $b > a$

Да

Сложные условия

Для каких значений x истинны условия:

$x < 6$ и $x < 10$

$x < 6$

$x < 6$ и $x > 10$

нет таких

$x > 6$ и $x < 10$

$6 < x < 10$

$x > 6$ и $x > 10$

$x > 10$

$x < 6$ или $x < 10$

$x < 10$

$x < 6$ или $x > 10$

$x > 6$ или $x < 10$

все x

$x > 6$ или $x > 10$

$x > 6$

Задания

«3»: Ввести три числа и определить, верно ли, что они вводились в порядке возрастания.

Пример:

Введите три числа:

4 5 17

да

«4»: Ввести номер месяца и вывести название времени года.

Пример:

Введите номер месяца:

4

весна

Задания

«5»: Ввести возраст человека (от 1 до 150 лет) и вывести его вместе с последующим словом «год», «года» или «лет».

Пример:

Введите возраст:

24

Вам 24 года

Введите возраст:

57

Вам 57 лет

Программирование на алгоритмическом языке

Тема 4. Циклы

Циклы

Цикл – это многократное выполнение одинаковых действий.

- цикл с **известным** числом шагов
- цикл с **неизвестным** числом шагов (цикл с условием)

Задача. Вывести на экран 5 раз слово «Привет».

Особенность: одинаковые действия выполняются 5 раз.



Можно ли решить известными методами?

Циклы

алг **Привет**

нач

вывод "Привет" , нс

вывод "Привет" , нс

вывод "Привет" , нс

вывод "Привет" , нс

вывод "Привет" , нс

кон



Что плохо?

Циклы

начало цикла

алг **Привет**

нач

тело цикла

нц **5** раз

вывод **"Привет!"**, нс

конец цикла

кц

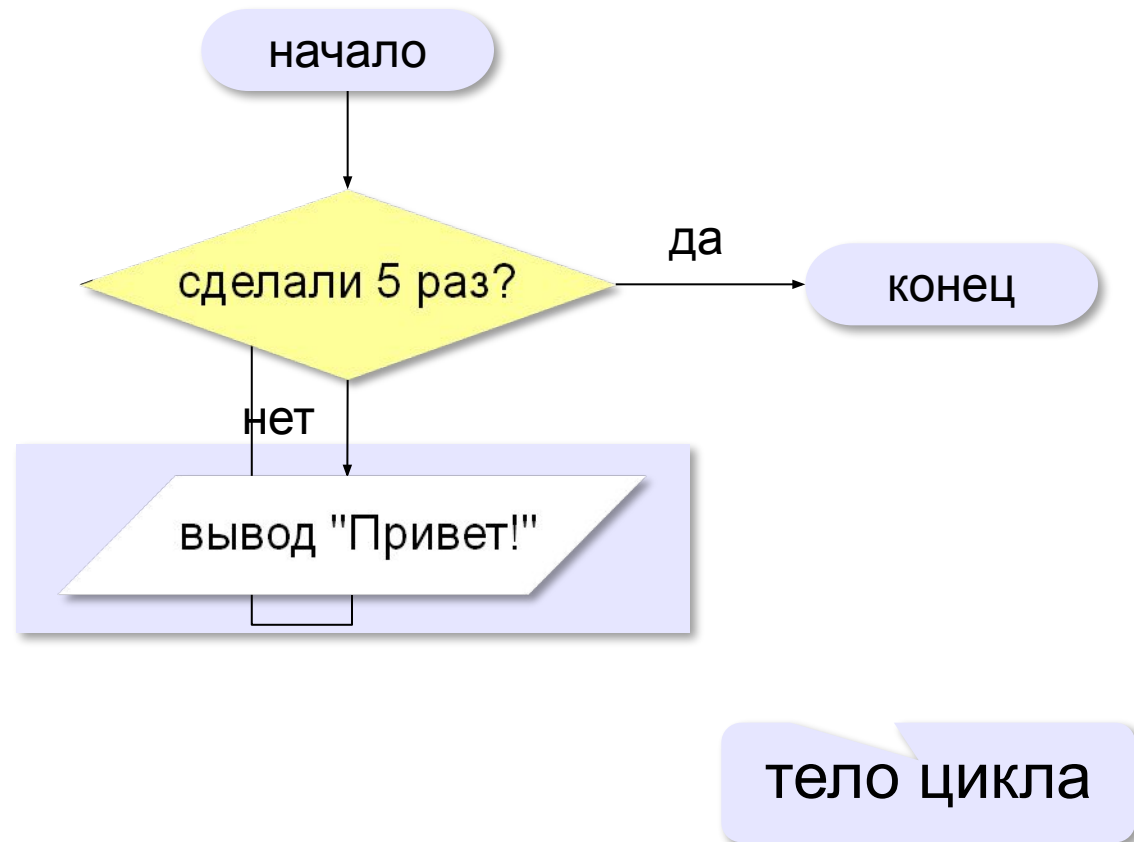
кон



Как выглядит блок-схема?

Циклы

Блок-схема:



Число шагов – переменная

Задача: ввести количество повторения с клавиатуры.

алг **Привет**

нач

цел N

вывод "**Сколько раз?**", нс

ввод N

нц N раз

 вывод "**Привет!**", нс

кц

кон

Задания

«3»: Ввести натуральное число и вывести в строчку все числа от 1 до этого числа.

Пример:

Введите натуральное число:

4

Ответ: 1 2 3 4

«4»: Ввести два целых числа, найти их произведение, не используя операцию умножения.

Пример:

Введите два числа:

4 15

4*15=60

Задания

«5»: Ввести натуральное число N и найти сумму всех чисел от 1 до N ($1+2+3+\dots+N$).

Пример:

Введите число слагаемых:

100

Сумма чисел от 1 до 100 равна 5050

Циклы

алг **Привет**

нач

нц **5** раз

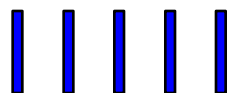
вывод "**Привет!**", нс

кц

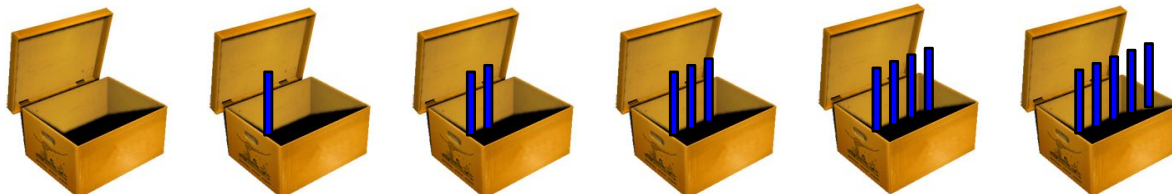
кон



Как отсчитать ровно 5 раз?

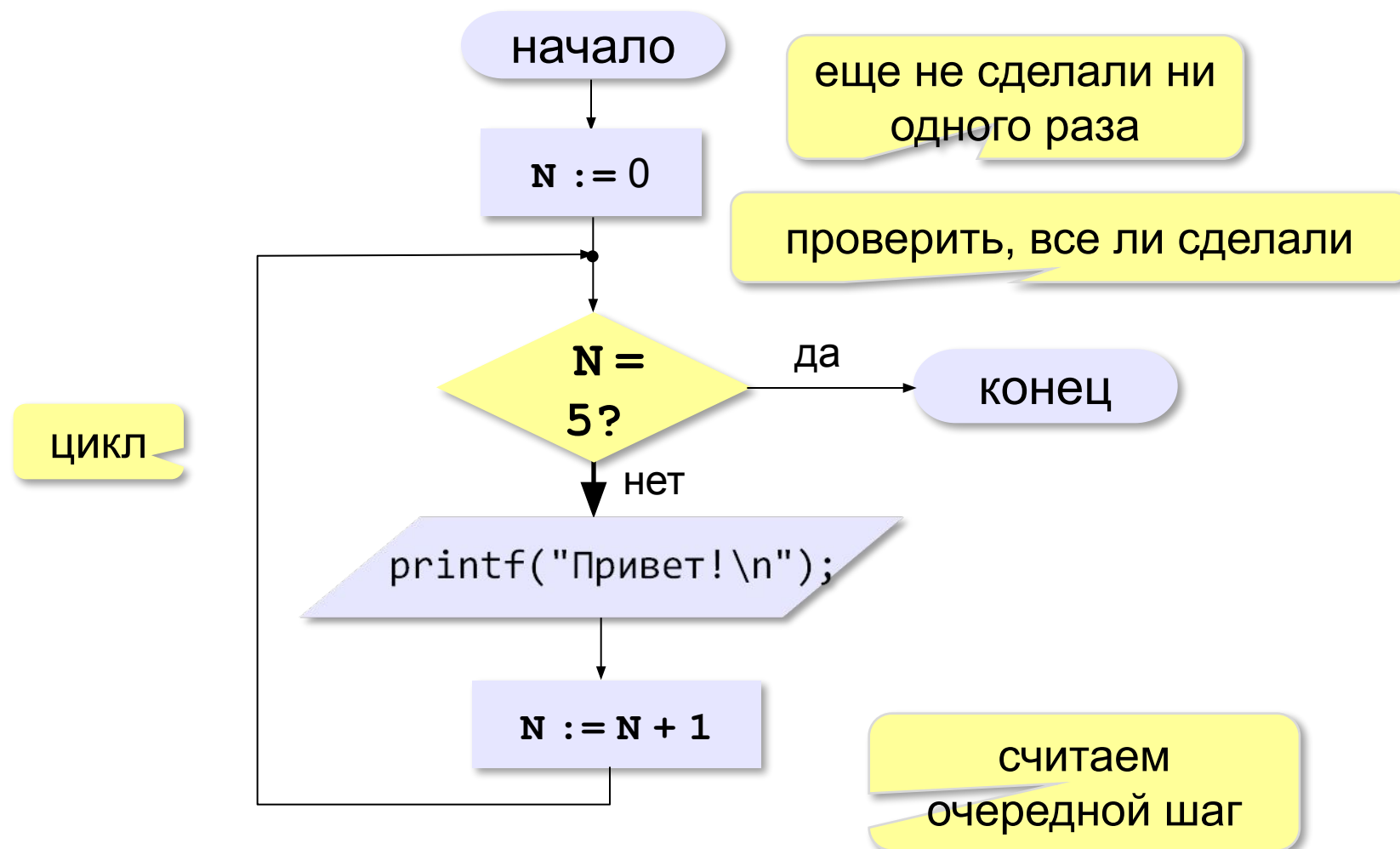


Как запоминать, сколько раз уже сделали?



$N := N + 1$

Блок-схема алгоритма



Цикл с условием

```
алг Привет 2
нач
  цел N
  N := 0
  нц пока N <> 5
    вывод "Привет! ", нс
    N := N + 1
  кц
кон
```

Цикл с условием

Вместо знаков вопроса добавьте числа и операторы так, чтобы цикл выполнялся ровно 5 раз:

```
алг Привет 3
нач
  цел N
  N := 5
  нц пока N <> 0
    вывод "Привет! ", нс
    N := N - 1
  кц
кон
```


Что получим?

алг **Пример 1**

нач

цел N

N := 1

нц пока N ≤ 5

вывод N, нс

 N := N + 1

кц

кон



1

2

3

4

5

Что получим?

алг **Пример 2**

нач

цел N

N := 1

нц пока N ≤ 5

вывод N, нс

 N := N + 2

кц

кон

→
1
3
5

Что получим?

алг **Пример 3**

нач

цел N

N := 2

нц пока N <> 5

 вывод N, нс

 N := N + 2

кц

кон



2

4

6

8

10

12

14

16

...



Условие цикла никогда не станет ложным – это **зацикливание!**

Что получим?

алг **Пример 4**

нач

цел N

N := 1

нц пока N ≤ 5

вывод N*N*N, нс

 N := N + 1

кц

кон



1

8

27

64

125

Что получим?

алг **Пример 5**

нач

цел N

N := 5

нц пока N >= 1

вывод N*N*N, нс

 N := N - 1

кц

кон



125

64

27

8

1

Задания

«3»: Ввести натуральное число вывести квадраты и кубы всех чисел от 1 до этого числа.

Пример:

Введите натуральное число:

3

1: 1 1

2: 4 8

3: 9 27

«4»: Ввести два целых числа a и b ($a \leq b$) и вывести квадраты все чисел от a до b .

Пример:

Введите два числа:

4 5

4*4=16

5*5=25

Задания

«5»: Ввести два целых числа a и b ($a \leq b$) и вывести сумму квадратов всех чисел от a до b .

Пример:

Введите два числа:

4 10

Сумма квадратов 371

Циклы с условием

Пример: Отпилить полено от бревна. Сколько раз надо сделать движения пилой?

Задача: Ввести целое число (< 2000000) и определить число цифр в нем.

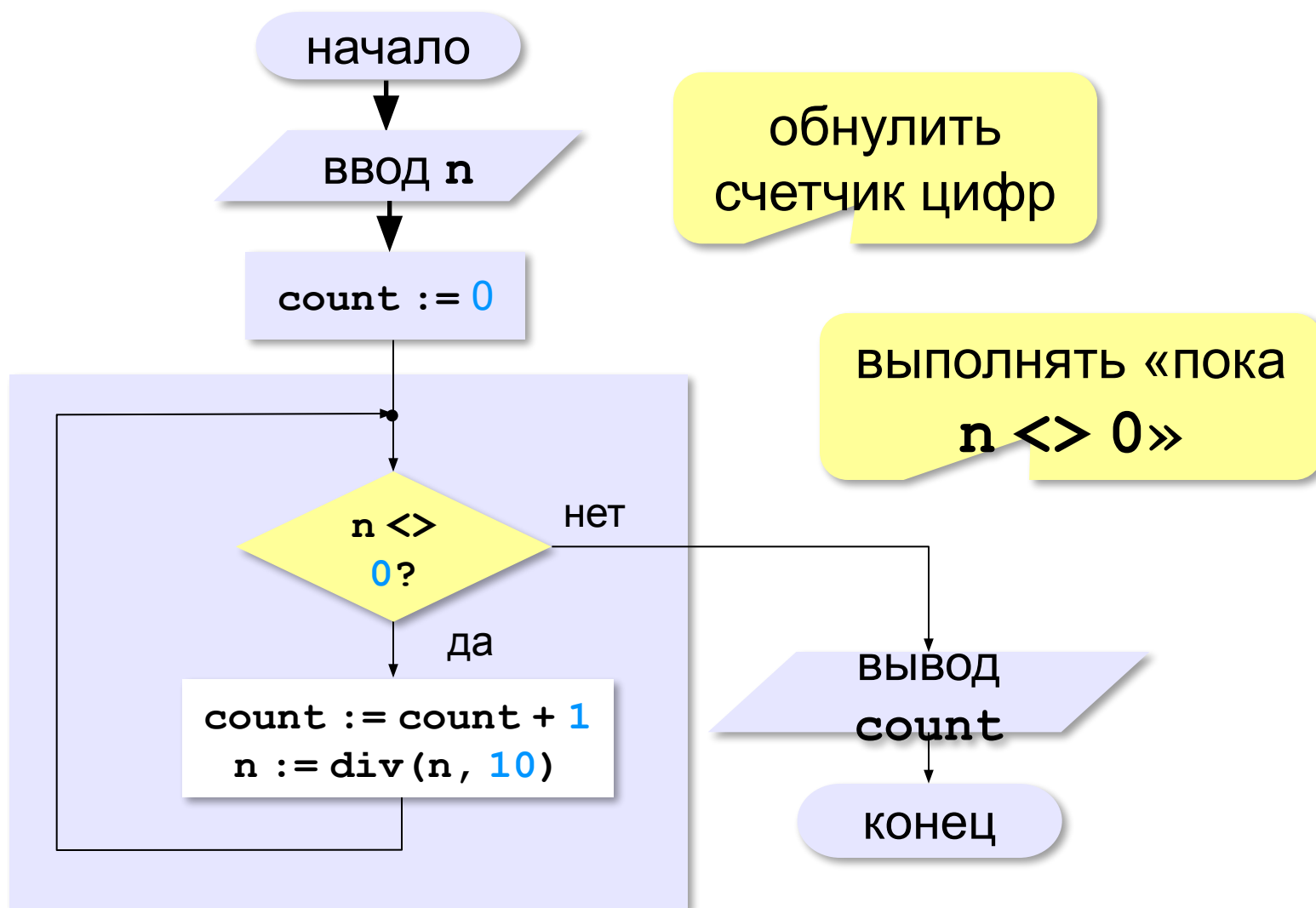
Идея решения: Отсекаем последовательно последнюю цифру, увеличиваем счетчик.

n	count
123	0
12	1
1	2
0	3

Проблема: Неизвестно, сколько шагов надо сделать.

Решение: Надо остановиться, когда $n = 0$, т.е. надо делать «пока $n \neq 0$ ».

Блок-схема алгоритма



Программа

алг Число цифр

нач

цел n , count, $n1$

вывод "Введите целое число", нс

ввод n ; $n1 := n$

count := 0

нц пока $n \neq 0$

count := count + 1

$n := \text{div}(n, 10)$

кц

вывод "В числе ", $n1$, " нашли ", count, " цифр"

кон



Что плохо?

Цикл с условием

Особенности:

- МОЖНО ИСПОЛЬЗОВАТЬ СЛОЖНЫЕ УСЛОВИЯ:

```
нц пока a < 10 и b > 5  
    a := a + 5; b := b - 2  
кц
```

- МОЖНО ЗАПИСЫВАТЬ В ОДНУ СТРОЧКУ, РАЗДЕЛЯЯ КОМАНДЫ ТОЧКОЙ С ЗАПЯТОЙ:

```
нц пока a < b ; b := b - 2 кц
```

Цикл с условием

Особенности:

- условие пересчитывается при **каждом** входе в цикл
- если условие на входе в цикл ложно, цикл не выполняется ни разу

```
а := 4;  b := 6  
нц пока а > b;  а := а - b  кц
```

- если условие никогда не станет ложным, программа **зацикливается**

```
а := 4;  b := 6  
нц пока а < b;  d := а + b  кц
```

Сколько раз выполняется цикл?

$a := 4; b := 6$

нц пока $a < b$; $a := a + 1$ кц

2 раза

$a = 6$

$a := 4; b := 6$

нц пока $a < b$; $a := a + b$ кц

1 раз

$a = 10$

$a := 4; b := 6$

нц пока $a > b$; $a := a + 1$ кц

0 раз

$a = 4$

$a := 4; b := 6$

нц пока $a < b$; $b := a - b$ кц

1 раз

$b = -2$

$a := 4; b := 6$

нц пока $a < b$; $a := a - 1$ кц

зацикливание

Задания

«3»: Ввести целое число и определить, верно ли, что в нём ровно 3 цифры.

Пример:

Введите число: Введите число:

123

1234

Да.

Нет.

«4»: Ввести целое число и найти сумму его цифр.

Пример:

Введите целое число:

1234

Сумма цифр числа 1234 равна 10.

Задания

«5»: Ввести целое число и определить, верно ли, что в его записи есть две одинаковые цифры, стоящие **рядом**.

Пример:

Введите целое число:

1232

Нет.

Введите целое число:

1224

Да.

«6»: Ввести целое число и определить, верно ли, что в его записи есть две одинаковые цифры, **НЕ** обязательно стоящие рядом.

Пример:

Введите целое число:

1234

Нет.

Введите целое число:

1242

Да.

Задания-2

«3»: Ввести целое число и определить, верно ли, что в нём ровно 1 цифра «9».

Пример:

Введите число: Введите число:

193

1994

Да .

Нет .

«4»: Ввести целое число и определить, верно ли, что все его цифры четные.

Пример:

Введите число: Введите число:

2684

2994

Да .

Нет .

Задания-2

«5»: Ввести целое число и определить, верно ли, что все его цифры расположены в порядке возрастания.

Пример:

Введите целое число:

1238

Да.

Введите целое число:

1274

Нет.

«6»: Ввести целое число и «перевернуть» его, так чтобы первая цифра стала последней и т.д.

Пример:

Введите целое число:

1234

4321

Введите целое число:

782

287

Вычисление НОД

НОД = наибольший общий делитель двух натуральных чисел – это наибольшее число, на которое оба исходных числа делятся без остатка.

Перебор:

1. Записать в переменную **k** минимальное из двух чисел.
2. Если **a** и **b** без остатка делятся на **k**, то стоп.
3. Уменьшить **k** на 1.
4. Перейти к шагу 2.



Где будет НОД?



Почему алгоритм обязательно закончится?

это цикл с условием!

Алгоритм Евклида

Надо: вычислить наибольший общий делитель (НОД) чисел a и b .

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

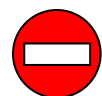
$$\begin{aligned}\text{НОД}(a, b) &= \text{НОД}(a - b, b) \\ &= \text{НОД}(a, b - a)\end{aligned}$$



Евклид
(365-300 до. н. э.)

Пример:

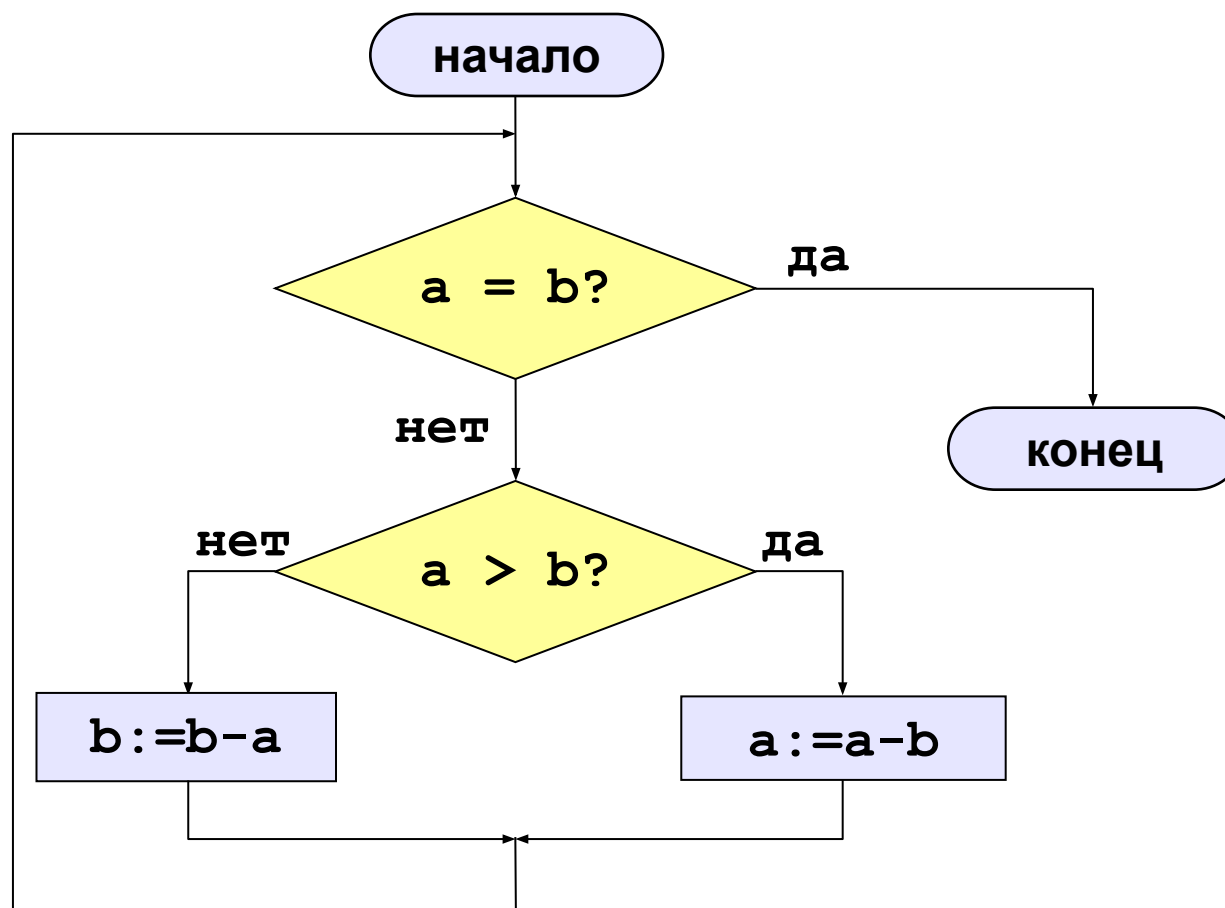
$$\begin{aligned}\text{НОД}(14, 21) &= \text{НОД}(14, 21 - 14) = \text{НОД}(14, 7) \\ &= \text{НОД}(7, 7) = 7\end{aligned}$$



много шагов при большой разнице чисел:


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


Блок-схема алгоритма



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:

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

Еще ⁷ один вариант:

$$\text{НОД}(2 \cdot a, 2 \cdot b) = 2 \cdot \text{НОД}(a, b)$$

$$\text{НОД}(2 \cdot a, b) = \text{НОД}(a, b) \quad | \quad \text{при нечетном } b$$

Алгоритм Евклида

«3»: Составить программу для вычисления НОД с помощью алгоритма Евклида.

«4»: Составить программу для вычисления НОД с помощью **модифицированного** алгоритма Евклида и заполнить таблицу:

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

Алгоритм Евклида

«5»: Выполнить задание на «4» и подсчитать число шагов алгоритма для каждого случая.

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

Программирование на алгоритмическом языке

Тема 5. Циклы с переменной

Цикл с переменной

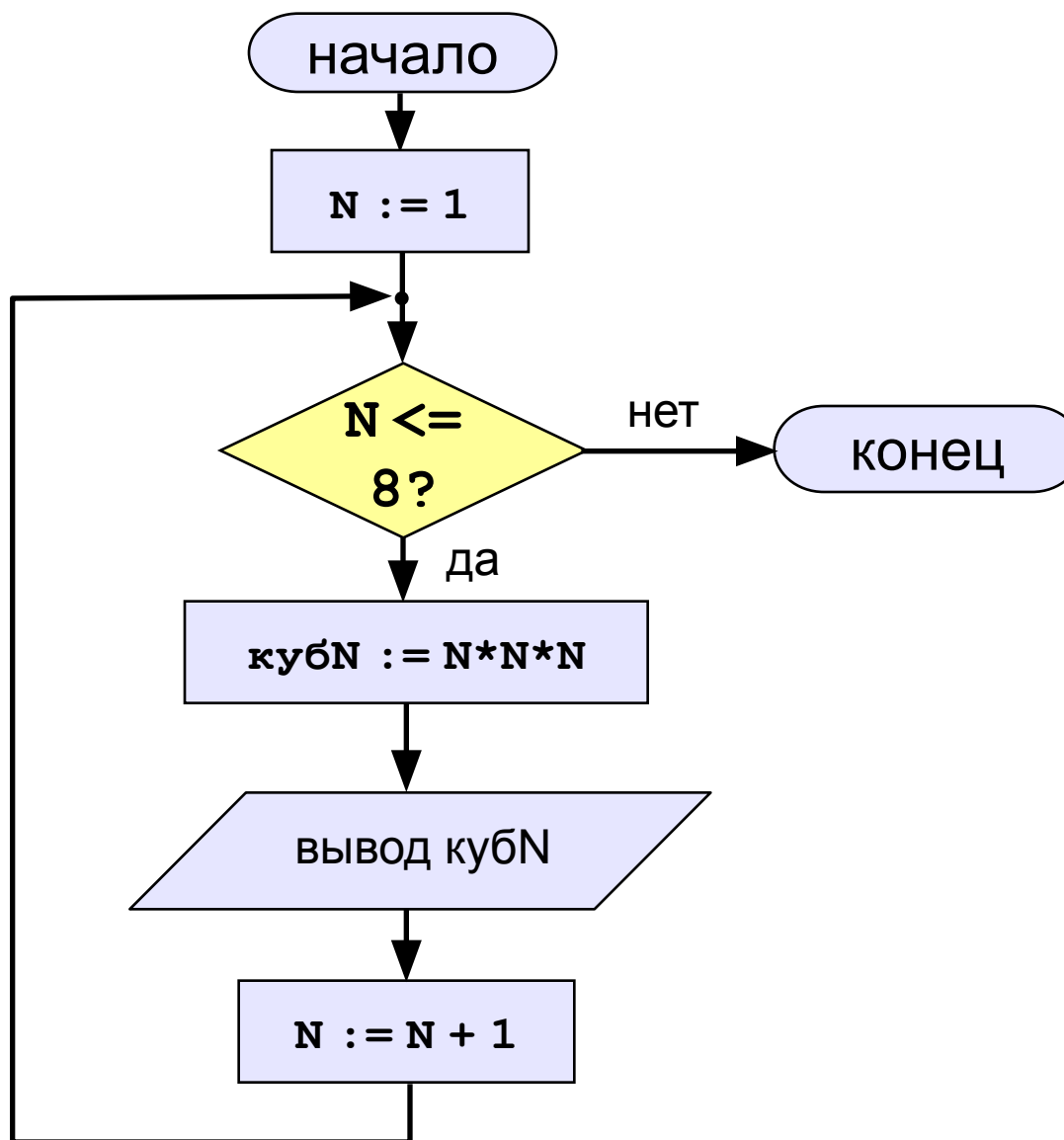
Задача: вывести кубы чисел от 1 до 8.



Можно ли решить известными способами?

1. Нужны ли переменные? Сколько?
2. Как они должны изменяться?
3. Нужен ли цикл?

Блок-схема алгоритма



Цикл с переменной

Задача: вывести кубы натуральных чисел от 1 до 8.

алг Кубы

нач

цел N, кубN

N :=

1

нц пока N <= 8

кубN := N * N * N

вывод кубN, нс

N := N +

1

кц

кон

3 действия с N

Цикл с переменной

Задача: вывести кубы натуральных чисел от 1 до 8.

алг Кубы

нач

для 1, 2, 3, ..., 8

цел N, кубN

нц для N от 1 до 8

кубN := N*N*N

вывод кубN, нс

кц

кон



Как обойтись без переменной кубN?

Цикл с переменной

Задача: вывести кубы чётных чисел от 2 до 8.

алг Кубы

нач

для 2, 4, 6, 8

цел N, кубN

нц для N от 2 до 8 шаг 2

кубN := N*N*N

вывод кубN, нс

только целые!

кц

кон

Сколько раз выполняется цикл?

$a := 1$

нц для i от 1 до 3; $a := a + 1$ кц

$a = 4$

$a := 1$

нц для i от 3 до 1; $a := a + 1$ кц

$a = 1$

$a := 1$

нц для i от 1 до 3 шаг -1; $a := a + 1$ кц

$a = 1$

$a := 1$

нц для i от 3 до 1 шаг -1; $a := a + 1$ кц

$a = 4$

Цикл с переменной

Особенности:

- переменная цикла может быть только целой (**цел**)
- начальное и конечное значения и шаг – целые
- можно записывать в одну строчку, разделяя команды точкой с запятой:

```
нц для n от 1 до 4; вывод n кц
```

- если шаг > 0 и конечное значение $<$ начального, цикл не выполняется ни разу (проверка условия в начале цикла, цикл с предусловием)
- если шаг < 0 и конечное значение $>$ начального, цикл не выполняется ни разу

Замена одного вида цикла на другой

```
нц для i от 1 до 10
  | тело цикла
кц
```

```
i := 1
нц пока i <= 10
  | тело цикла
  i := i + 1
кц
```

```
нц для i от a до b шаг -1
  | тело цикла
кц
```

```
i := a
нц пока i >= b
  | тело цикла
  i := i - 1
кц
```

Замена цикла **для** на **пока** возможна **всегда**.

Замена **пока** на **для** возможна только тогда, когда можно заранее **вычислить число шагов цикла**.

Задания

«3»: Ввести натуральное число N и вывести числа от N до 1 (через одно) в порядке убывания.

Пример:

Введите натуральное число:

8

Ответ: 8 6 4 2

Задания

«4»: Ввести два целых числа a и b ($a \leq b$) и вывести кубы всех чисел от a до b .

Пример:

Введите два числа:

4 6

$4 * 4 * 4 = 64$

$5 * 5 * 5 = 125$

$6 * 6 * 6 = 216$

«5»: Ввести целое число a и вывести сумму квадратов всех чисел от 1 до a с шагом 0.1.

Пример:

Введите последнее число:

3

Сумма 91.7

$$1^2 + 1.1^2 + 1.2^2 + \dots + a^2$$

Задания-2

«4»: Ввести *a* и *b* и вывести квадраты и кубы чисел от *a* до *b*.

Пример:

Введите границы интервала:

4 6

4: 16 64

5: 25 125

6: 36 216

«5»: Вывести квадраты и кубы 10 чисел следующей последовательности: 1, 2, 4, 7, 11, 16, ...

Пример:

1: 1 1

2: 4 8

4: 16 64

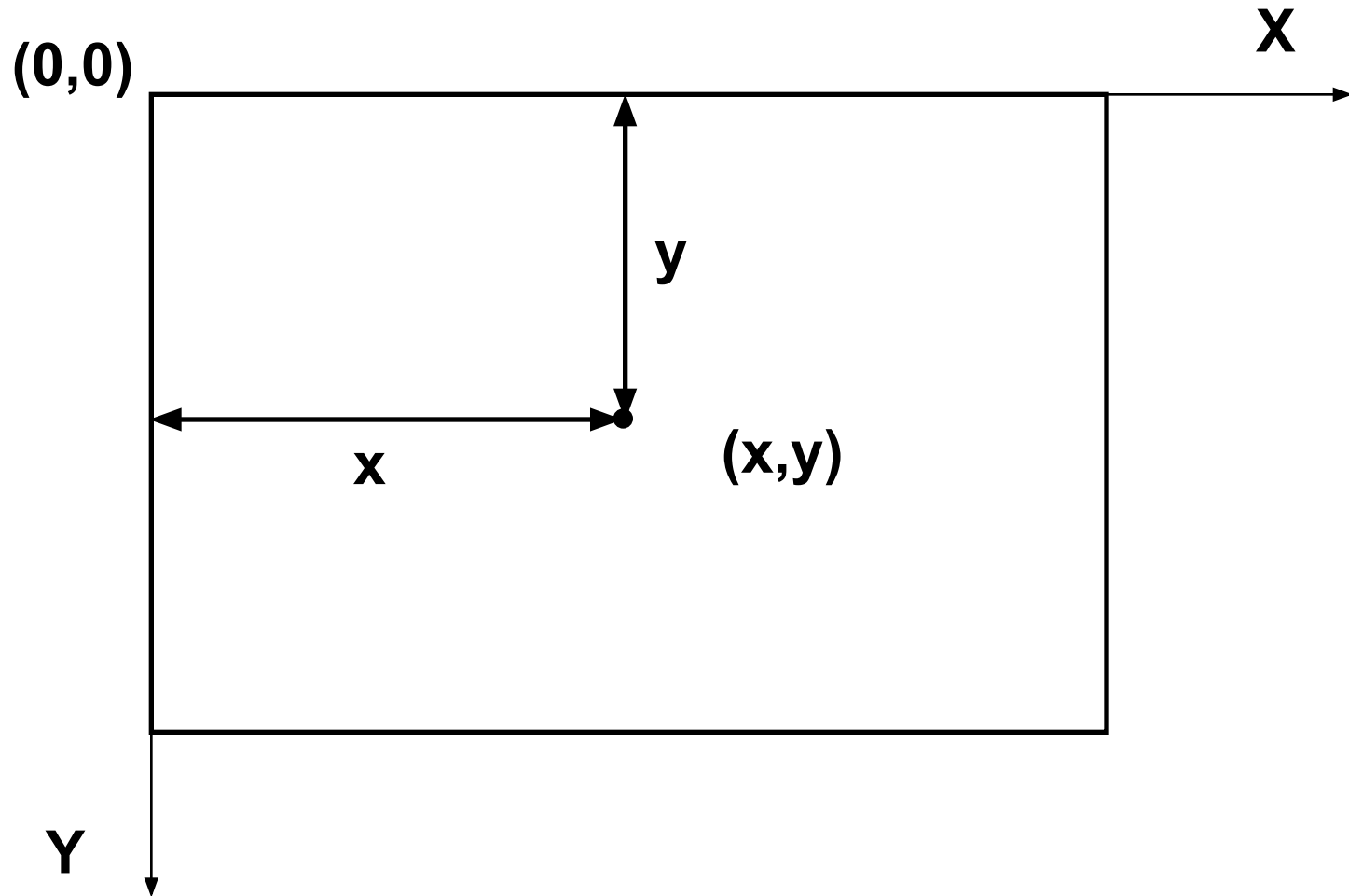
...

46: 2116 97336

Программирование на алгоритмическом языке

Тема 6. Графика

Система координат



Исполнитель Рисователь

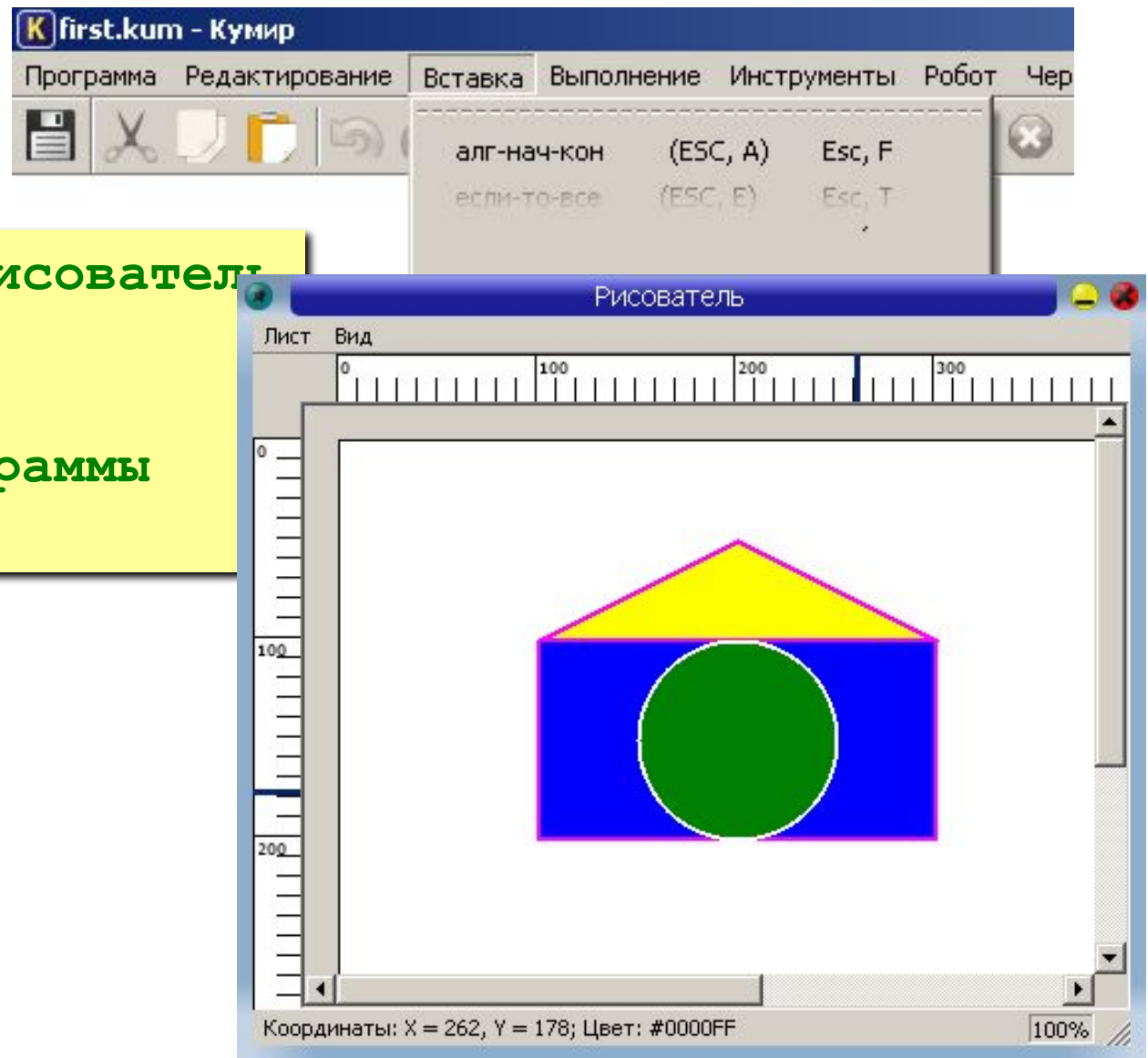
использовать Рисователь

алг

нач

| текст программы

кон



Линии

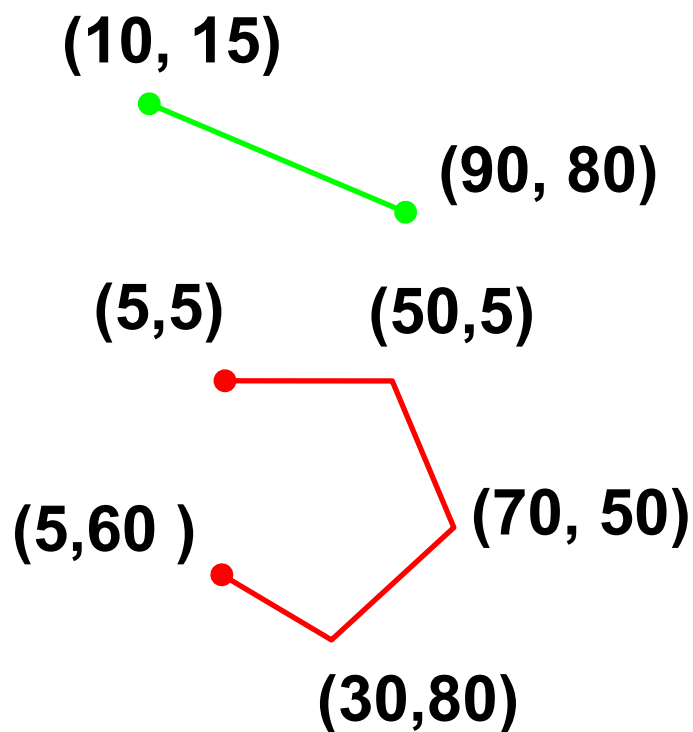
Цвет и толщина линий:

толщина линии

перо (2, "синий")

черный
белый
серый
фиолетовый
синий

голубой
зеленый
желтый
оранжевый
красный

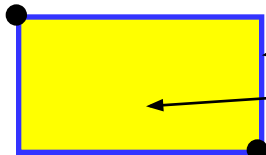


перо (1, "зеленый")
линия (10, 15, 90, 80)

перо (1, "красный")
в точку (5, 5)
линия в точку (50, 5)
линия в точку (70, 50)
линия в точку (30, 80)
линия в точку (5, 60)

Фигуры с заливкой

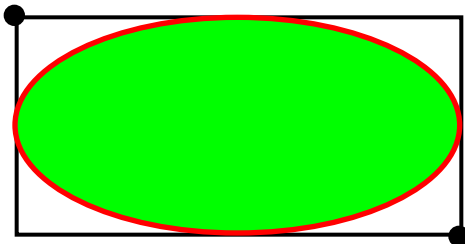
(0,0)



(80,40)

```
перо (1, "синий")  
кисть ("желтый")  
прямоугольник (0, 0, 80, 40)
```

(0,0)



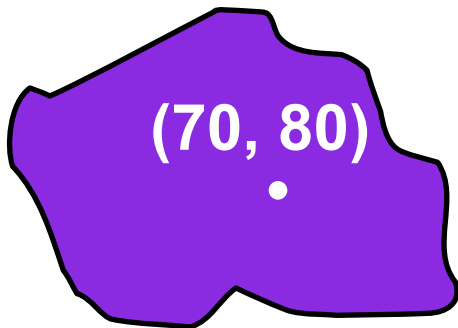
(100,50)

```
перо (1, "красный")  
кисть ("зеленый")  
эллипс (0, 0, 100, 50)
```



Как построить круг?

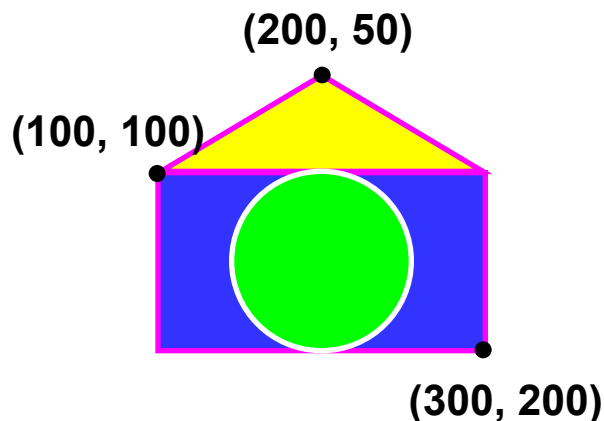
заливку



(70,80)

```
кисть ("фиолетовый")  
залить (70, 80)
```

Пример



использовать Рисователь

алг Домик

нач

перо (2, "фиолетовый")

кисть ("синий")

прямоугольник (100, 100, 300, 200)

в точку (100, 100)

линия в точку (200, 50)

линия в точку (300, 100)

кисть ("желтый")

залить (200, 75);

перо (2, "белый");

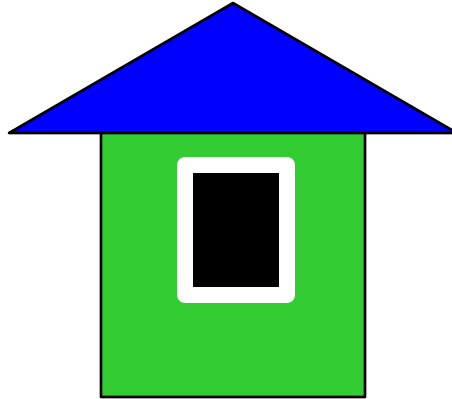
кисть ("зеленый");

эллипс (150, 100, 250, 200);

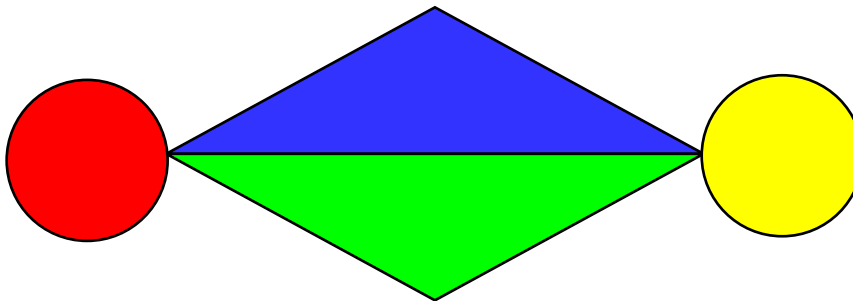
кон

Задания

«3»: «Домик»

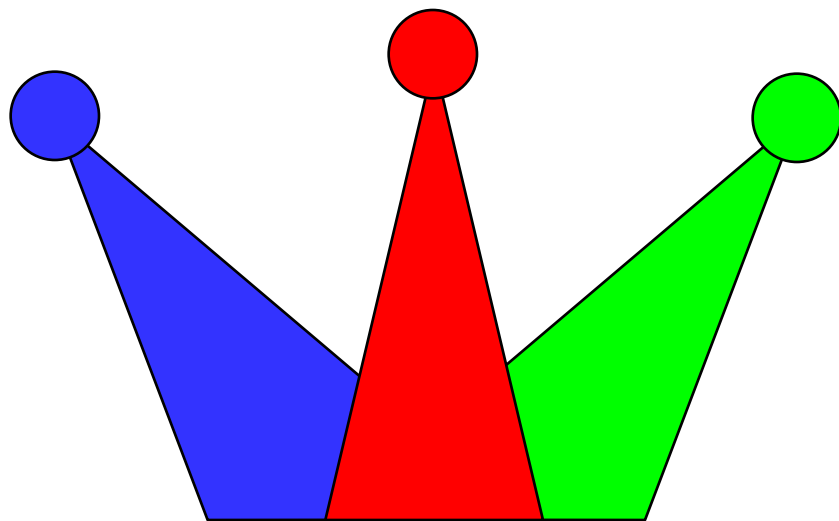


«4»: «Лягушка»



Задания

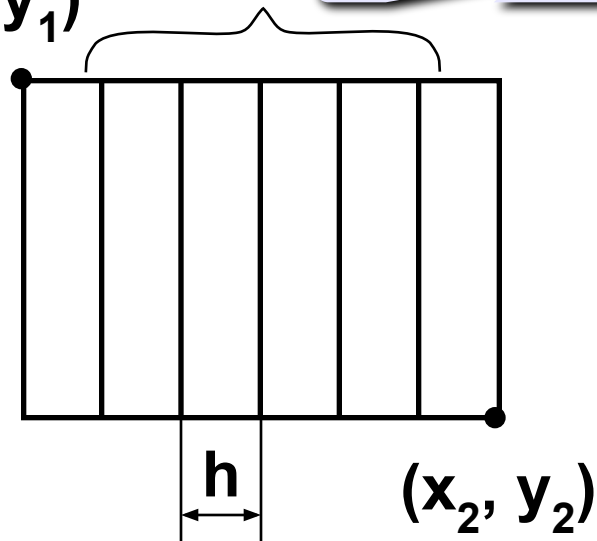
«5»: «Корона»



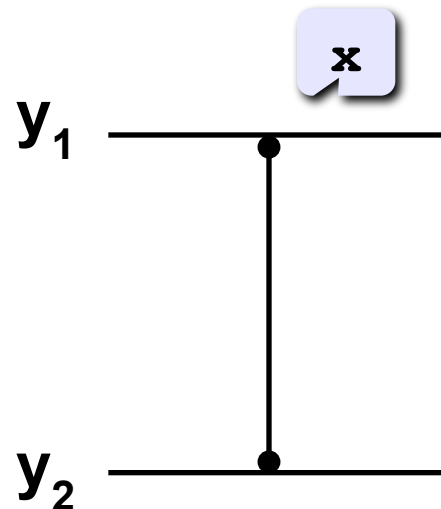
Штриховка

(x_1, y_1)

N линий (N=5)



$$h = \frac{x_2 - x_1}{N + 1}$$



прямоугольник (x_1, y_1, x_2, y_2)

$x := x_1 + h$

линия (x, y_1, x, y_2)

$x := x + h$

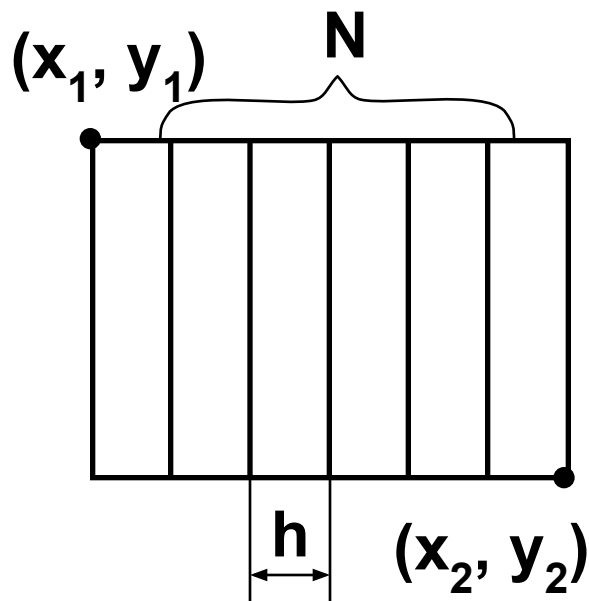
линия (x, y_1, x, y_2)

$x := x + h$

...

цикл N раз

Штриховка (программа)



использовать **Рисователь**
алг **Штриховка**

нач

цел $N = 5$ | **число** **линий**

цел $x1 = 100$, $x2 = 300$

цел $y1 = 100$, $y2 = 200$

вещ h, x

$h := (x2 - x1) / (N + 1)$

прямоугольник $(x1, y1, x2, y2)$

$x := x1 + h$

нц N раз

линия $(\text{int}(x), y1, \text{int}(x), y2)$

$x := x + h$

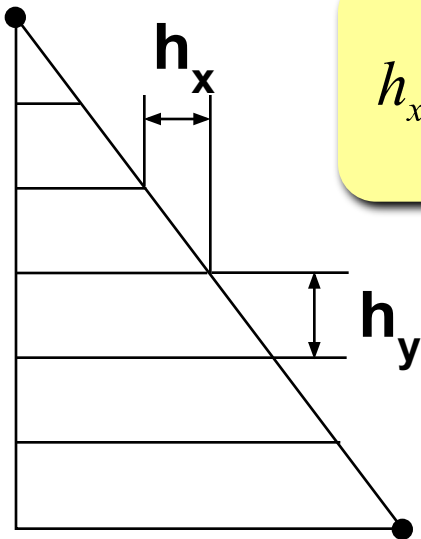
кц

кон

целая часть

Штриховка

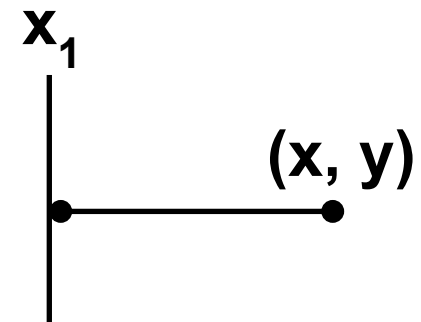
(x_1, y_1)



(x_2, y_2)

$$h_x = \frac{x_2 - x_1}{N + 1}$$

$$h_y = \frac{y_2 - y_1}{N + 1}$$



```
х := х1 + hх; у := у1 + hy
```

```
линия (х1, int (у), int (х), int (у))
```

```
х := х + hх; у := у + hy
```

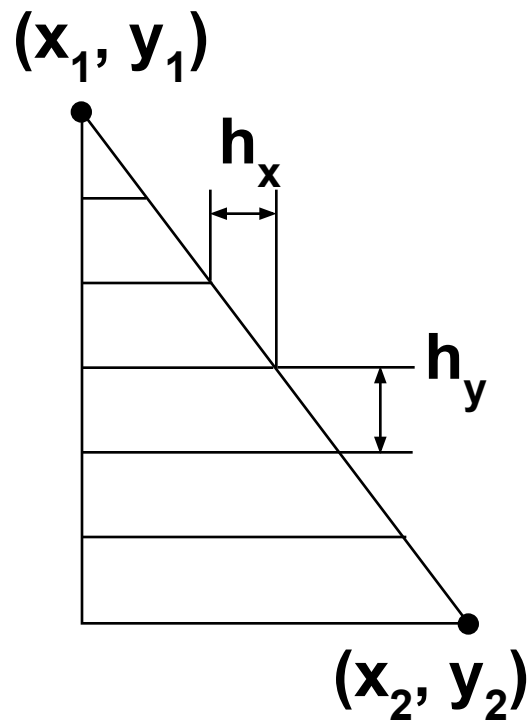
```
линия (х1, int (у), int (х), int (у))
```

```
х := х + hх; у := у + hy
```

```
...
```

цикл N раз

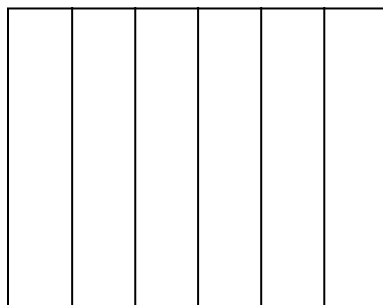
Штриховка



```
вещ  $h_x, h_y, x, y$   
 $h_x := (x_2 - x_1) / (N + 1)$   
 $h_y := (y_2 - y_1) / (N + 1)$   
в точку  $(x_1, y_1)$   
линия в точку  $(x_1, y_2)$   
линия в точку  $(x_2, y_2)$   
линия в точку  $(x_1, y_1)$   
 $x := x_1 + h_x; y := y_1 + h_y$   
нц N раз  
    линия  $(x_1, \text{int}(y), \text{int}(x), \text{int}(y))$   
     $x := x + h_x$   
     $y := y + h_y$   
кц
```


Задания

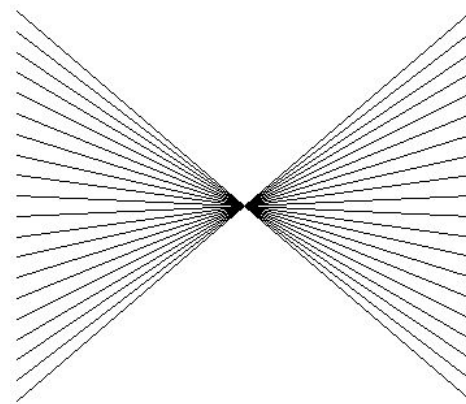
«3»: Ввести с клавиатуры количество линий,
построить фигуру и выполнить штриховку:



«4»: Ввести с клавиатуры количество линий,
построить фигуру и выполнить штриховку:

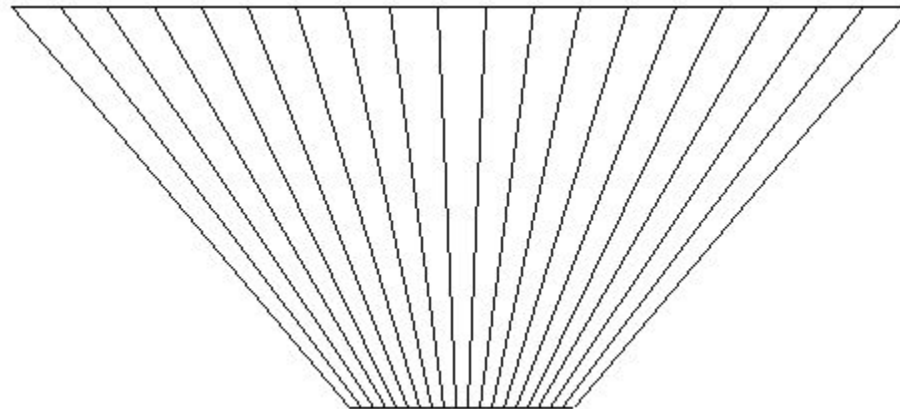


или



Задания

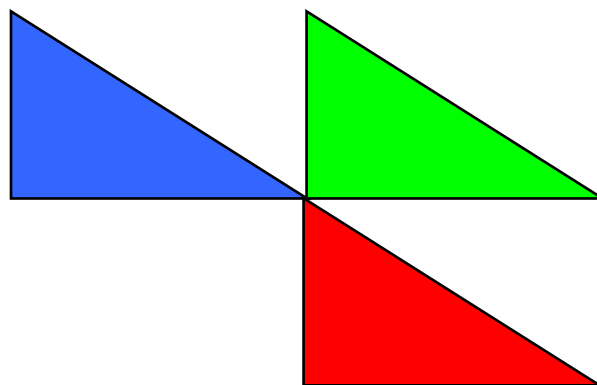
«5»: Ввести с клавиатуры количество линий и построить фигуру:



Программирование на алгоритмическом языке

Тема 7. Вспомогательные алгоритмы

Задача



Можно ли решить известными методами?

Особенность: три похожие фигуры.

общее: размеры, угол поворота

отличия: координаты, цвет

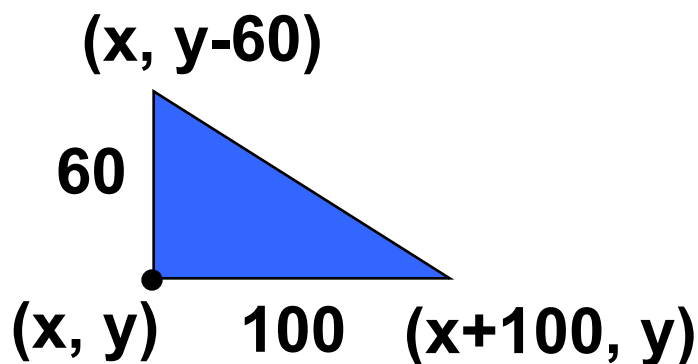


Сколько координат надо задать?

С чего начать?

- найти похожие действия (три фигуры)
- найти **общее** (размеры, форма, угол поворота) и **отличия** (координаты, цвет)
- отличия = **параметры** алгоритма (доп. данные)

цепочка символов



использовать **Рисователь**

алг **Тр** (**цел** x, y , **лит** цвет)

нач

в точку (x, y)

линия в точку ($x, y-60$)

линия в точку ($x+100, y$)

линия в точку (x, y)

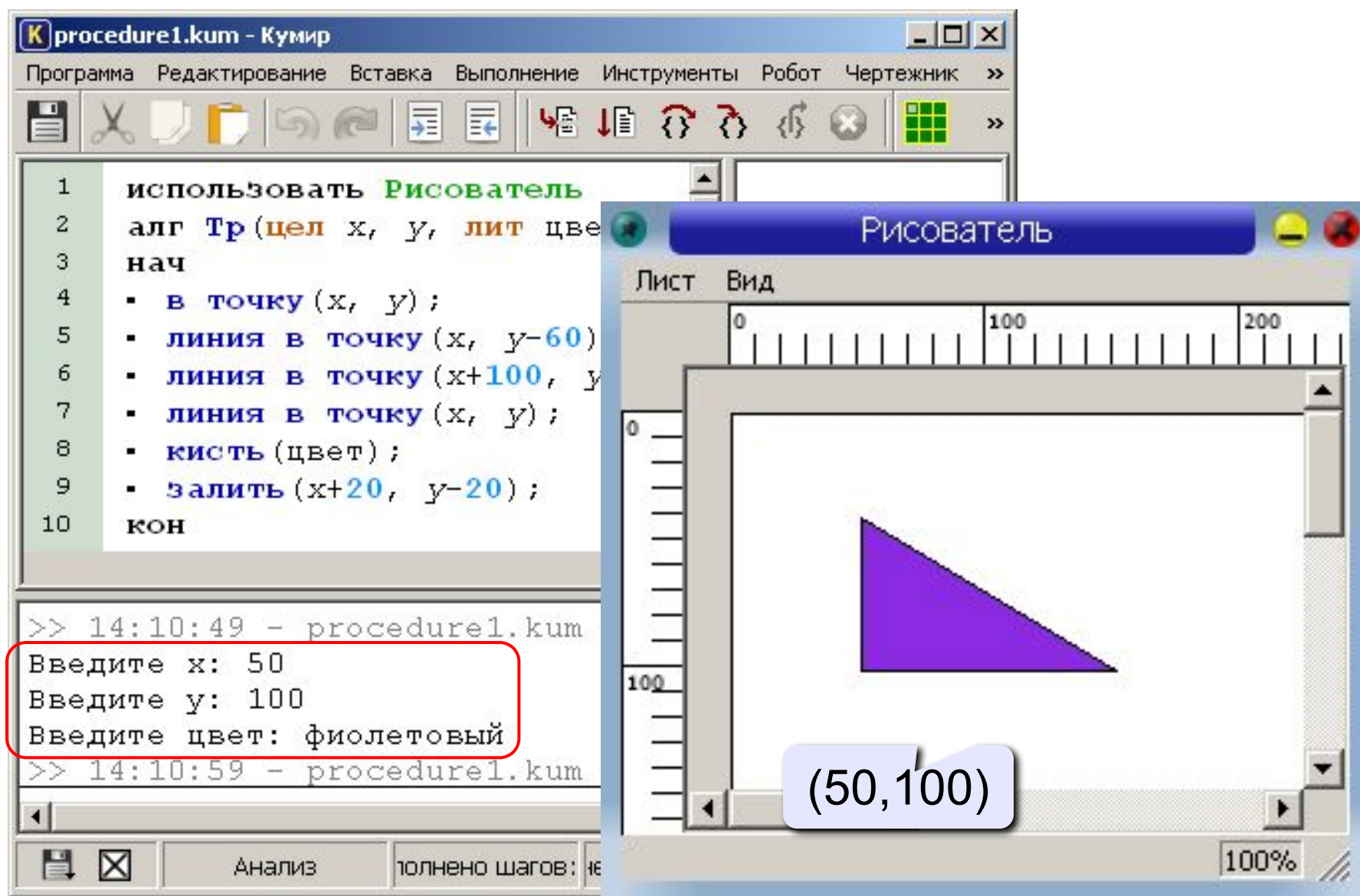
кисть (цвет)

залить ($x+20, y-20$)

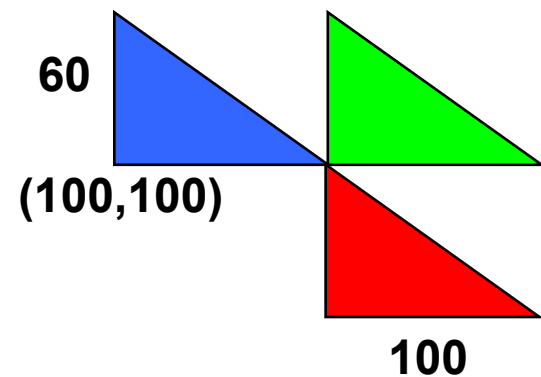
кон

параметры

Если запустить?



Как использовать?



ВЫЗОВЫ
алгоритма

использовать **Рисователь**

алг **Треугольники**

нач

перо (1, "черный")

Тр (100, 100, "синий")

Тр (200, 100, "зеленый")

Тр (200, 160, "красный")

кон

ОСНОВНОЙ
алгоритм

алг **Тр** (**цел** x, y, **лит** цвет)

нач

...

кон

ВСПОМОГАТЕЛЬНЫЙ
алгоритм

Вспомогательные алгоритмы

- расположены **ниже** основного
- в заголовке перечисляются **формальные** параметры, они обозначаются именами

```
алг Тр(цел х, у, лит цвет)
```

- для каждого параметра указывают тип
- однотипные параметры перечисляются через запятую
- при вызове в скобках указывают **фактические** параметры **в том же порядке**

```
Тр(200, 100, "зеленый")
```

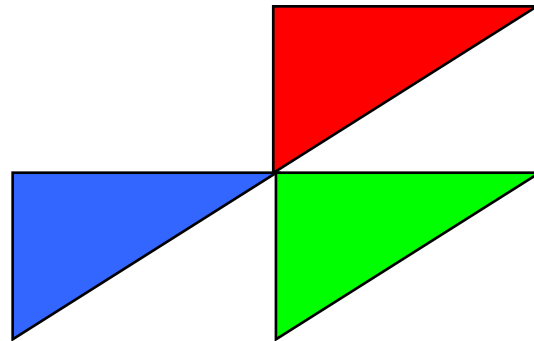
х

у

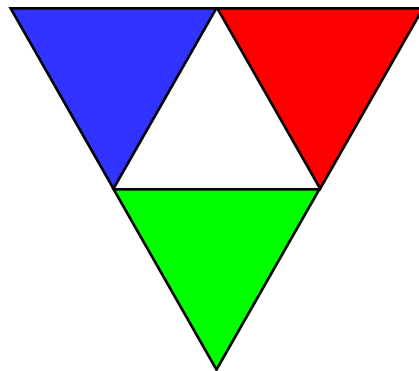
цвет

Задания

«3»: Используя одну процедуру, построить фигуру.

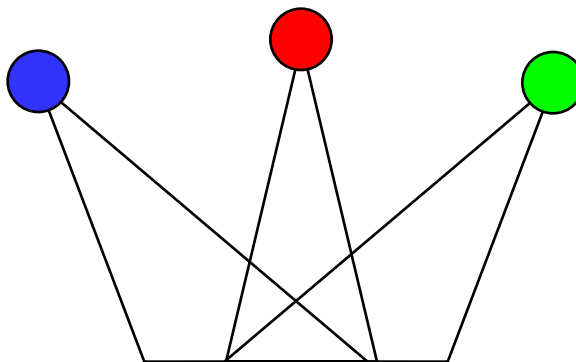


«4»: Используя одну процедуру, построить фигуру.



Задания

«5»: Используя одну процедуру, построить фигуру.



Рекурсивные объекты

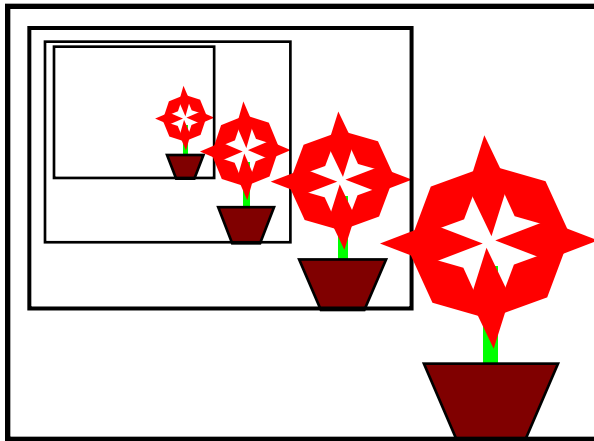
Примеры:

Сказка о попé и собаке:

У попа была собака, он ее любил.
Она съела кусок мяса, он ее убил.
В ямку закопал, надпись написал:

Сказка о попé и собаке

Рисунок с рекурсией:



Факториал:

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

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

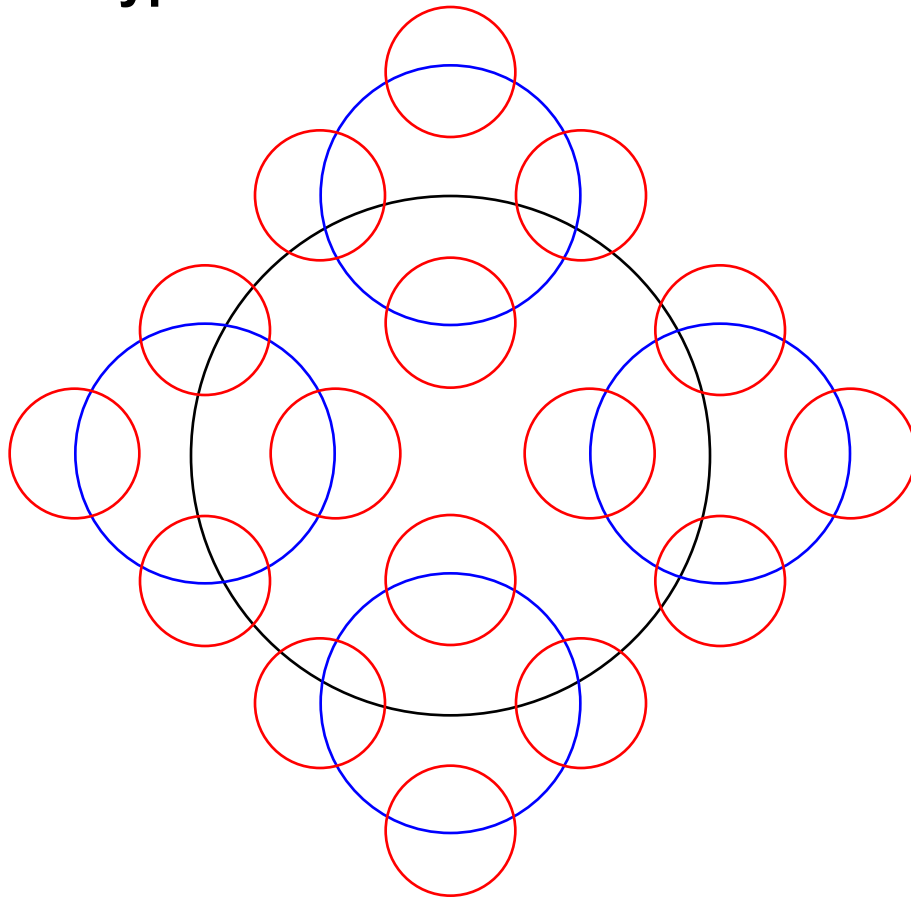
$$4! = 4 \cdot 3! = 4 \cdot 3 \cdot 2 \cdot 1$$

$$N! = N \cdot (N-1) \cdot \dots \cdot 2 \cdot 1$$

Рекурсивный объект – это объект, определяемый через один или несколько таких же объектов.

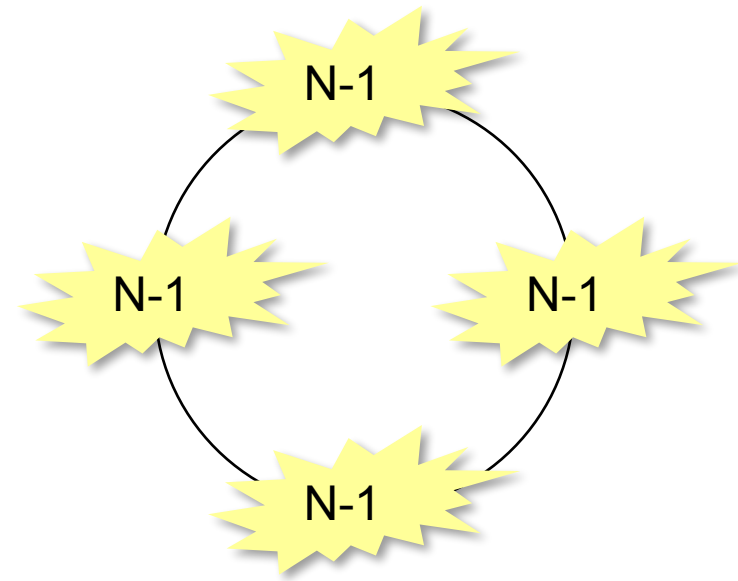
Рекурсивная фигура

3 уровня:



Где рекурсия?

- Фигура из N уровней – это
- окружность и
 - 4 фигуры из $N-1$ уровней

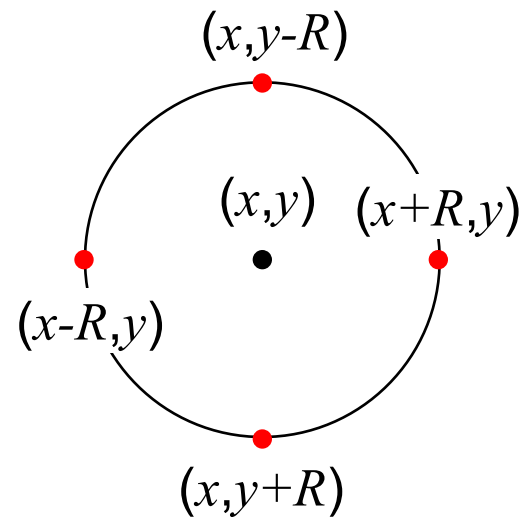


Рекурсивная фигура: алгоритм

центр

радиус

уровней

рекурсивные
вызовы**алг** РекОк (**цел** x, y, R, N)**нач**

окончание рекурсии

если $N \leq 0$ **то выход все****окружность** (x, y, R)РекОк ($x, y-R, \text{div}(R, 2), N-1$)РекОк ($x+R, y, \text{div}(R, 2), N-1$)РекОк ($x, y+R, \text{div}(R, 2), N-1$)РекОк ($x-R, y, \text{div}(R, 2), N-1$)**кон**

Рекурсивный алгоритм – это алгоритм, который вызывает сам себя (с другими параметрами!).

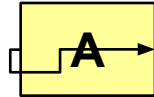
Рекурсивная фигура: программа

```
использовать Рисователь  
алг Рекурсия  
нач  
    РекОк (200, 200, 100, 3)  
кон
```

```
алг РекОк (цел x, y, R, N)  
нач  
    . . .  
кон
```

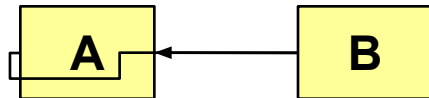
Рекурсивные алгоритмы

- вызывают сами себя прямо



прямая рекурсия

- ... или через другой алгоритм:

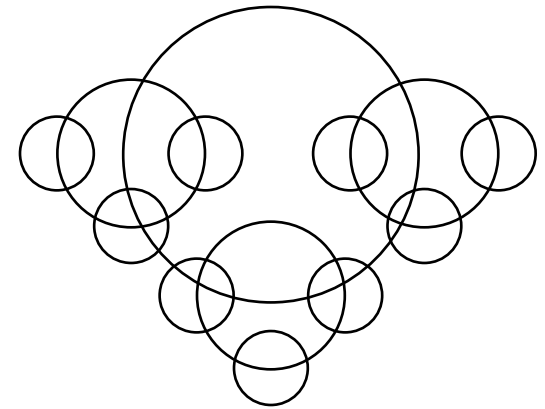


косвенная рекурсия

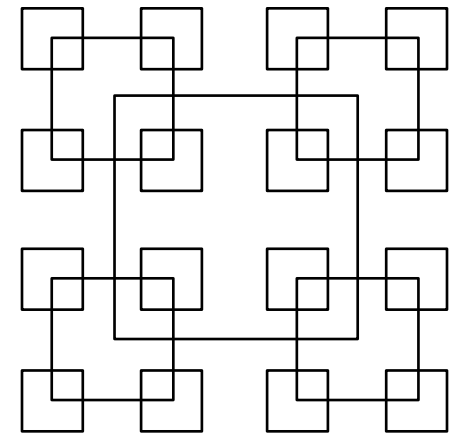
- должно быть условие окончания рекурсии (иначе?)
- рекурсия может стать бесконечной
- все задачи могут быть решены без рекурсии, но...
- часто рекурсивные алгоритмы проще и понятнее
- как правило, алгоритмы без рекурсии работают быстрее и требуют меньше памяти

Задания

«3»: Нарисовать рекурсивную фигуру, число уровней вводить с клавиатуры:

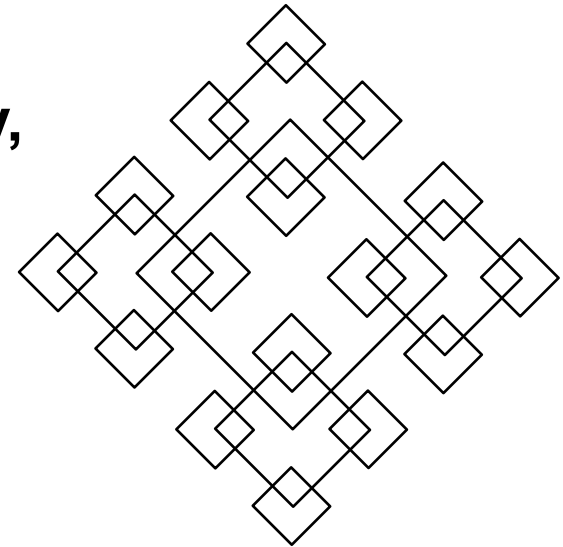


«4»: Нарисовать рекурсивную фигуру, число уровней вводить с клавиатуры:



Задания

«5»: Нарисовать рекурсивную фигуру, число уровней вводить с клавиатуры:



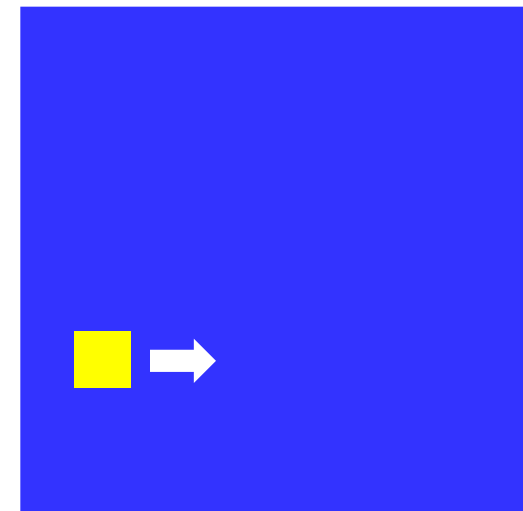
Программирование на алгоритмическом языке

Тема 8. Анимация

Анимация

Анимация (англ. *animation*) – оживление изображения на экране.

Задача: внутри синего квадрата 200 на 200 пикселей слева направо движется желтый квадрат 20 на 20 пикселей. Программа останавливается, если нажата клавиша **Esc** или квадрат дошел до границы синей области.



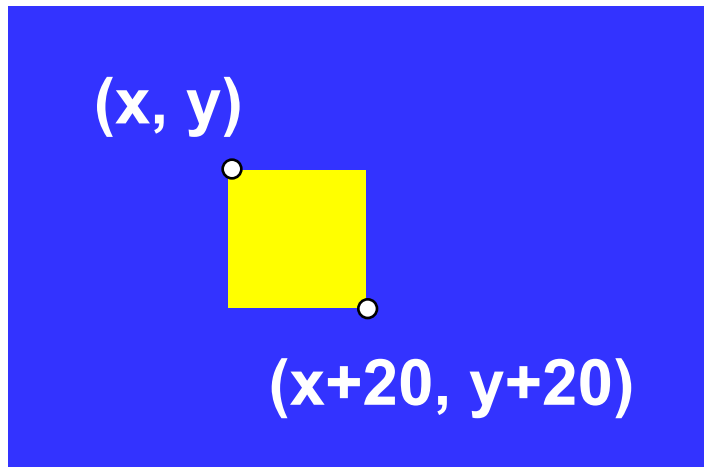
Проблема: как изобразить перемещение объекта на экране?

Привязка: состояние объекта задается координатами (x, y)

Принцип анимации:

1. рисуем объект в точке (x, y)
2. задержка на несколько миллисекунд
3. стираем объект
4. изменяем координаты (x, y)
5. переходим к шагу 1

Процедура (рисование и стирание)



- одна процедура рисует и стирает
- стереть = рисовать цветом фона
- границу квадрата отключить

рисуем: цвет кисти — желтый
стираем: цвет кисти — синий

```
алг фигура (цел x, y, лит цвет)
нач
    кисть (цвет)
    прямоугольник (x, y, x+20, y+20)
кон
```

Полная программа

использовать Рисователь

алг Анимация

нач

цел x , y | текущие координаты

кисть ("синий")

перо (1, "") | отключить контур

прямоугольник (0, 0, 200, 200) | синий фон

$x := 0$; $y := 100$ | начальные координаты

нц пока $x < 180$

Фигура (x , y , "желтый")

delay (50)

Фигура (x , y , "синий")

$x := x + 5$

кц

кон

пока не дошли до границы

алг Фигура (цел x , y , лит цвет)

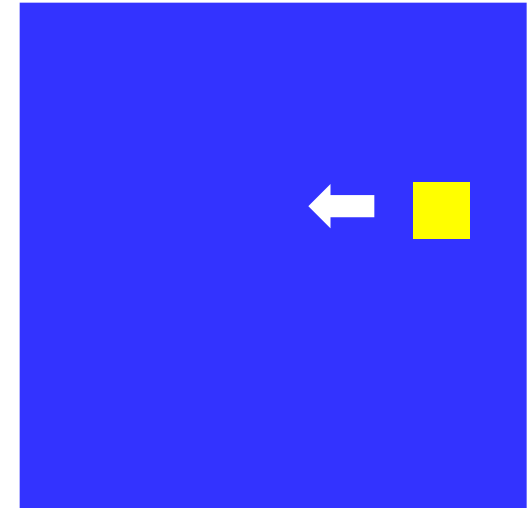
нач

...

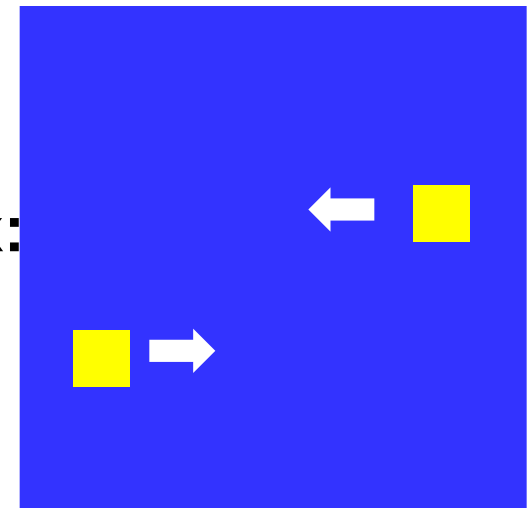
кон

Задания

«3»: Квадрат движется справа налево:

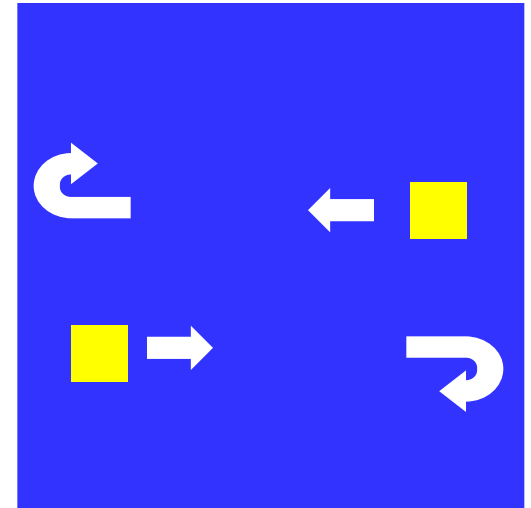


«4»: Два квадрата движутся в противоположных направлениях:



Задания

«5»: Два квадрата двигаются в противоположных направлениях и отталкиваются от стенок синего квадрата:



Управление клавишами

Задача: жёлтый квадрат внутри синего квадрата управляется клавишами-стрелками. Коды клавиш:

влево – **16777234**

вверх – **16777235**

вправо – **16777236**

вниз – **16777237**

Проблема: как изменять направление движения?

Решение:

с := клав

выбор

при **с = 16777234**: **х := х - 5** | **влево**

при **с = 16777235**: **у := у - 5** | **вверх**

при **с = 16777236**: **х := х + 5** | **вправо**

при **с = 16777237**: **у := у + 5** | **вниз**

все

ждать нажатия на клавишу, записать
ее код в переменную **с**



Когда стирать фигуру?

Программа

```
использовать Рисователь  
алг Управление клавишами  
нач
```

```
    цел x, y, c
```

```
        | нарисовать синий квадрат
```

```
x := 100; y := 100      | начальная точка
```

```
нц пока x < 180
```

```
    фигура (x, y, "желтый")    | рисуем фигуру
```

```
    c := клав                  | ждем нажатия клавиши
```

```
    фигура (x, y, "синий")     | стираем фигуру
```

```
выбор
```

```
    при c = 16777234: x := x - 5
```

```
    при c = 16777235: y := y - 5
```

```
    при c = 16777236: x := x + 5
```

```
    при c = 16777237: y := y + 5
```

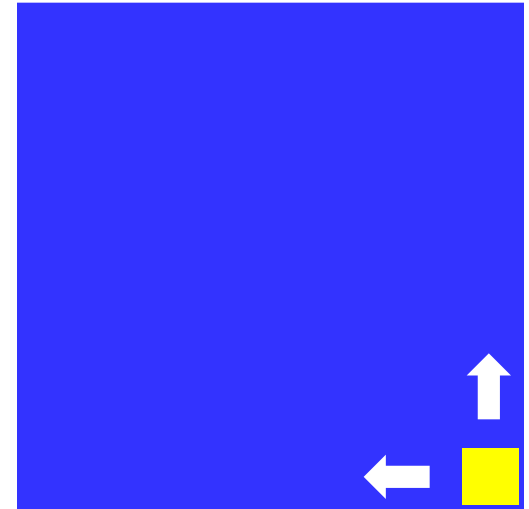
```
все
```

```
кц
```

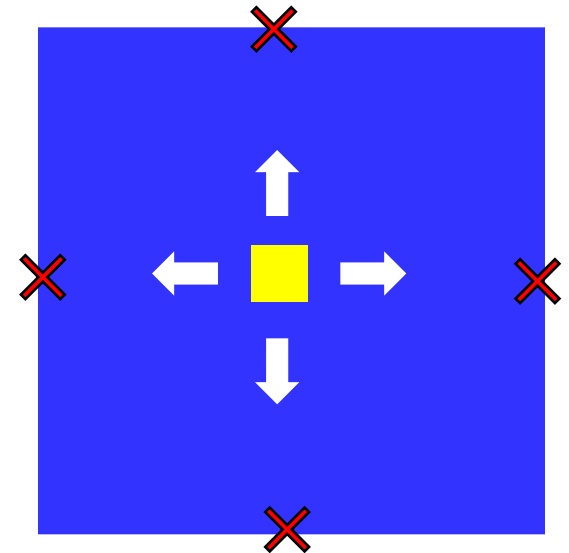
```
кон
```

Задания

«3»: Квадрат в самом начале стоит в правом нижнем углу, и двигается при нажатии стрелок только вверх или влево:

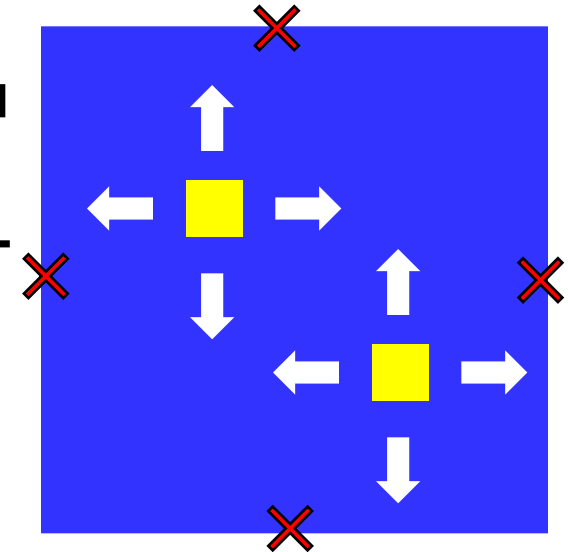


«4»: Квадрат двигается при нажатии стрелок, однако не может выйти за границы синего квадрата:



Задания

«5»: Два квадрата, один управляется стрелками, второй – любыми другими клавишами. Оба не могут выйти за границы синего поля.



Программирование на алгоритмическом языке

Тема 9. Случайные числа

Случайность и ее моделирование

Случайно...

- встретить друга на улице
- разбить тарелку
- найти 10 рублей
- выиграть в лотерею

Случайный выбор:

- жеребьевка на соревнованиях
- выигравшие номера в лотерее

Как получить случайность?



Случайные числа на компьютере

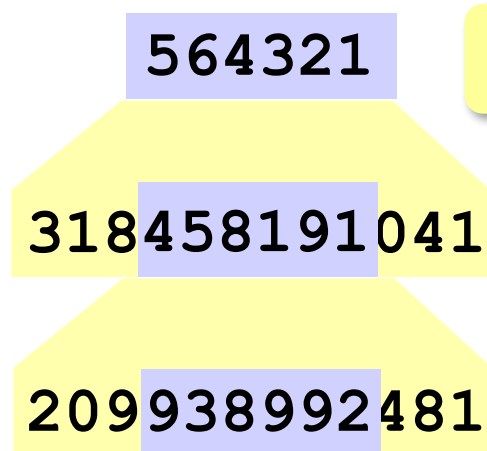
Электронный генератор



- 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)

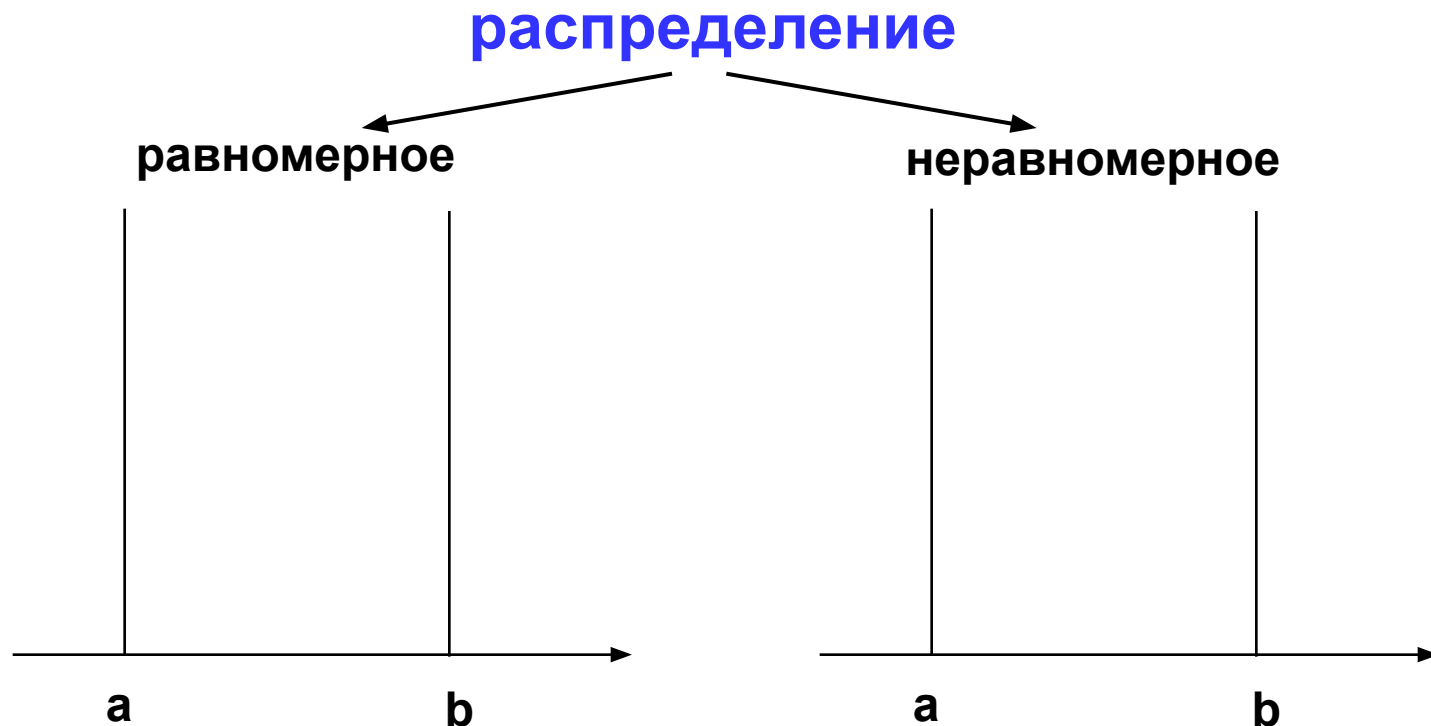


в квадрате

полный период
(последовательность
повторяется через 10^6 чисел)

Распределение случайных чисел

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

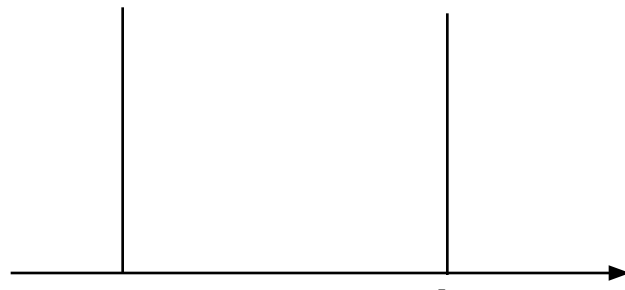



Сколько может быть разных распределений?

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

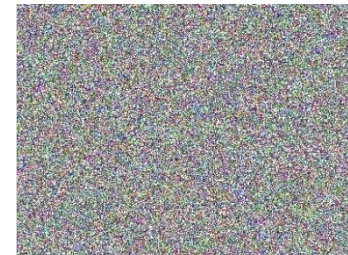
K := irand(0, 10) | интервал от 0 до 10 (≤ 10)

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

англ. *integer* – целый

Случайные числа

Задача: заполнить прямоугольник 200 на 150 пикселей равномерно точками случайного цвета



Как получить случайные координаты пикселя?

цел X, Y

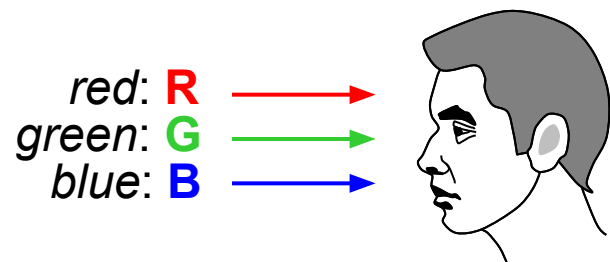
X := irand(0, 200)

Y := irand(0, 150)

Как добиться равномерности?

автоматически при использовании **irand**

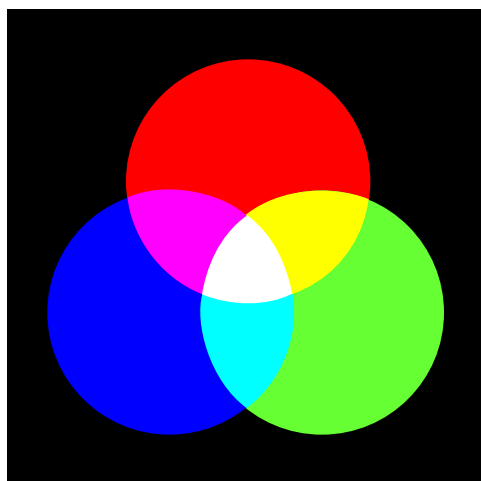
Цвет пикселя на мониторе


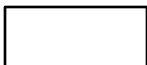









Красный, зелёный и синий лучи создают почти такое же ощущение, как луч «смешанного» цвета!

Вывод: цвет можно разложить на составляющие (каждая кодируется числом от 0 до 255).

Модель RGB:



	R	G	B
	0	0	0
	255	255	255
	255	0	0
	0	255	0
	0	0	255
	100	100	100

	RGB(255,0,255)
	RGB(255,255,0)
	RGB(0,255,255)

Случайный цвет пикселя

Случайные составляющие цвета:

```
цел r, g, b  
r := irand(0, 255)  
g := irand(0, 255)  
b := irand(0, 255)
```

это разные числа!

Управление цветом пикселя:

случайный цвет

```
пиксель (X, Y, RGB(r, g, b))
```

встроенные функции
Рисователя

Программа

```
использовать Рисователь
алг Случайные точки
нач
  цел x, y, r, g, b
  нц пока да
    x:=irand(0,200)
    y:=irand(0,100)
    r:=irand(0,255)
    g:=irand(0,255)
    b:=irand(0,255)
    пиксель(x,y,RGB(r,g,b))
  кц
кон
```

это бесконечный цикл:

нц пока да

...

кц

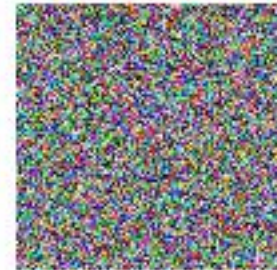
Задания

«3»: Заполнить квадрат точками случайного цвета.
размер квадрата ввести с клавиатуры:

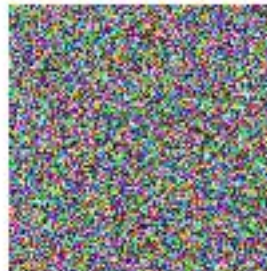
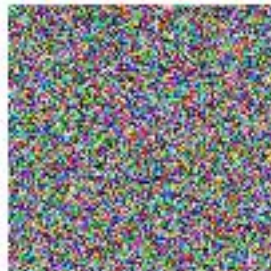
Пример:

Введите размер квадрата:

150

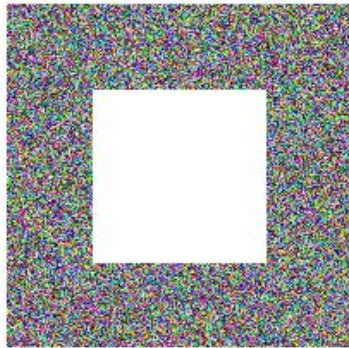


«4»: Заполнить область точками случайного цвета:

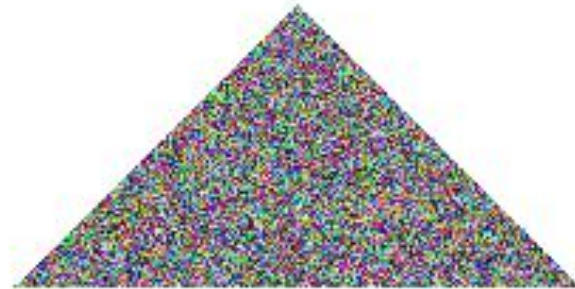


Задания

«5»: Заполнить область точками случайного цвета:



или



Конец фильма

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