# Алгоритмы и программирование

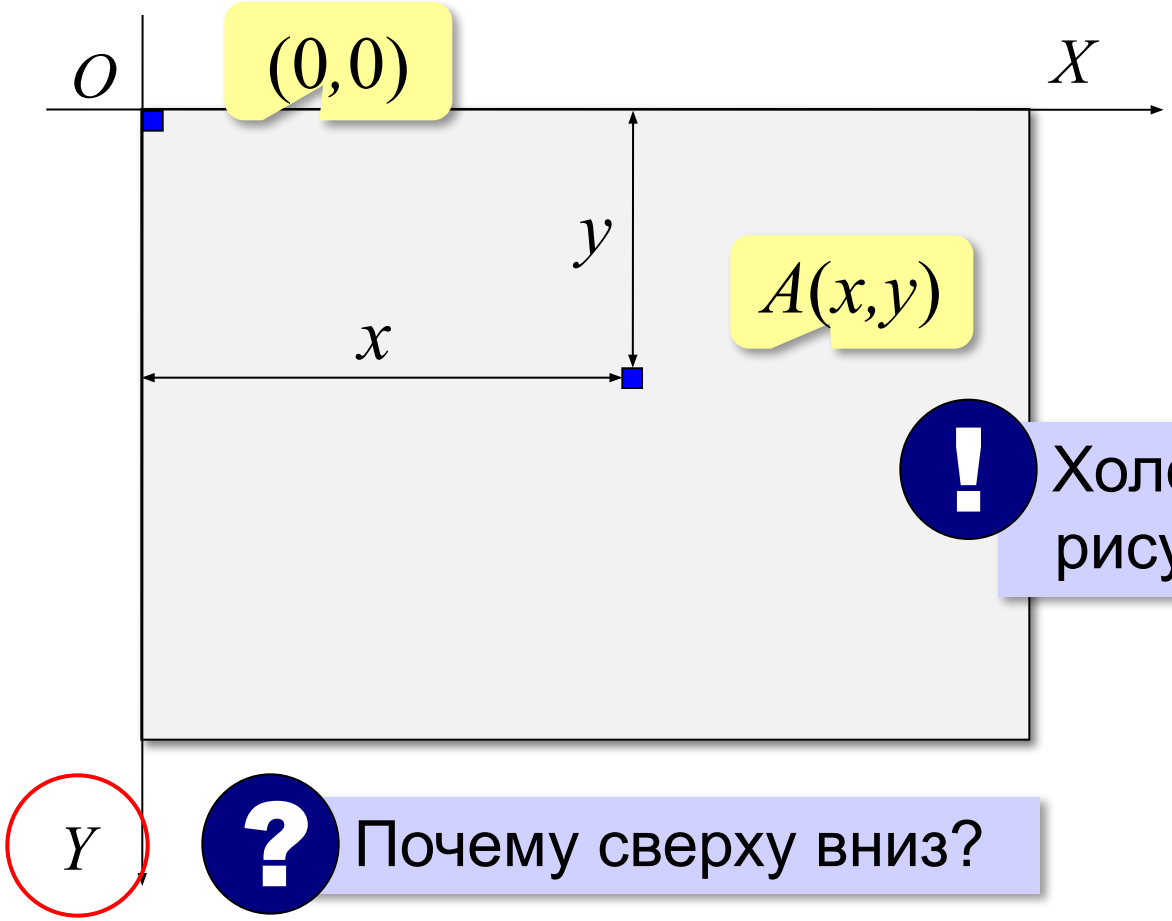
# Алгоритмы и программирование

## § 40. Компьютерная графика

# Графический режим монитора

**Холст** – это прямоугольная область экрана, доступная для рисования.




! Холст – это растровый рисунок!

? Почему сверху вниз?

# Исполнитель Рисователь

Задача – нарисовать что-то на холсте не с помощью мыши, а управляя исполнителем.


Как управлять?

Какие команды умеет выполнять Рисователь?

**Первая программа**:

```
использовать Рисователь
алг Холст
нач
   новый лист(500, 400, белый)
кон
```

программа (алгоритм)

начало

конец

аргументы

команда Рисователя

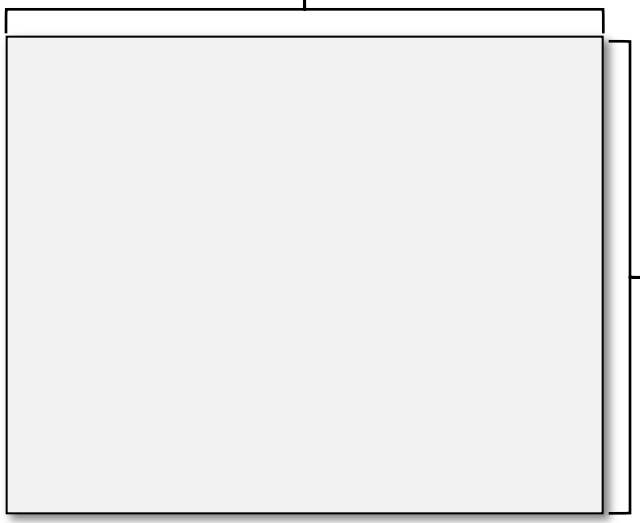
# Команда «новый лист»

```
использовать Рисователь
алг Холст
нач
   новый лист(500, 400, белый)
кон
```

ширина

высота

цвет



белый, чёрный,
серый, фиолетовый,
синий, голубой,
зелёный, жёлтый,
оранжевый, красный

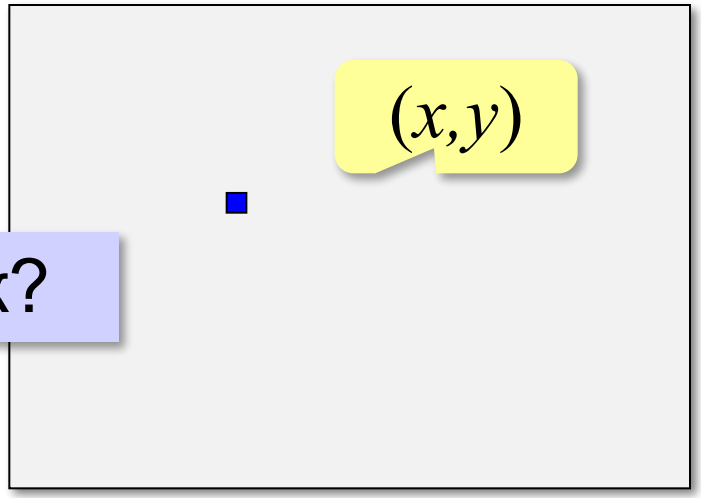
# Управление пикселями

? Что значит «управлять пикселем»?

аргументы (данные для работы)

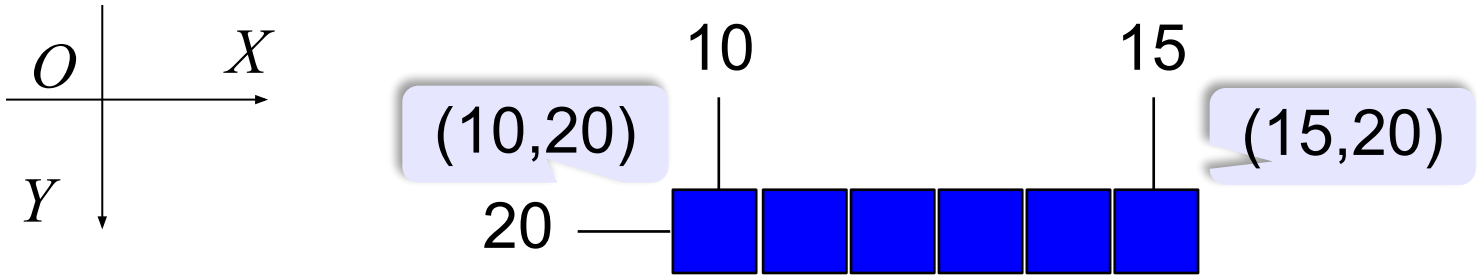
```
пиксель(10, 20, синий)
```

$x$ $y$ цвет




? Как рисовать отрезок?

# Отрезок




```
пиксель(10, 20, синий)
пиксель(11, 20, синий)
пиксель(12, 20, синий)
пиксель(13, 20, синий)
пиксель(14, 20, синий)
пиксель(15, 20, синий)
```

? Как сократить?

```
пиксель(x, 20, синий)
```

для **x** от 10 до 15

**x** – **переменная** (изменяемая величина)

# Отрезок

**Цикл** – это многократное выполнение одинаковых действий.

**X** – целая величина

```
цел X
X:=10
нц пока X <= 15
  пиксель(X, 20, синий)
  X:= X + 1
кц
```

присвоить **X** значение 10

тело цикла

начало цикла

конец цикла

присвоить **X** значение **X+1**

10 → 11 → 12 → 13 → …

? При каком **X** остановится цикл?

16

# Когда остановится цикл?

```
цел X
X:=10
нц пока X <= 15
   пиксель(X, 20, синий)
   X:= X + 1
кц
```

выполняется до тех пор, пока условие не станет ложным

это **цикл с условием**

? При каком **X** остановится цикл?

16

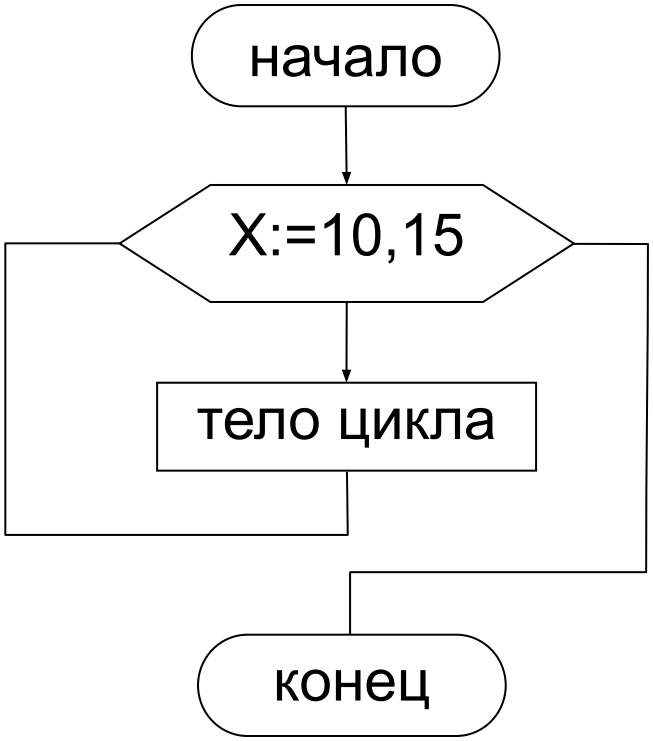
# Ещё один цикл

```
цел X
нц X от 10 до 15
   пиксель(X, 20, синий)
кц
```

цикл по переменной

блок-схема

**X** автоматически увеличится на 1 после каждого повторения!



# Что дальше?

Как нарисовать вертикальный отрезок?

… наклонный? в чём сложность?

… окружность?

# Алгоритмы и программирование

## § 41. Графические примитивы

# Что такое графический примитив?

**Графический примитив** — это элемент рисунка, который добавляется с помощью одной команды.

- пиксель
- линия
- прямоугольник
- окружность

# Линия (=отрезок)

```
линия(10, 20, 15, 20)
```

($x$,$y$) первой точки

($x$,$y$) второй точки

одна команда заменяет цикл

```
цел X
нц X от 10 до 15
  пиксель(X, 20, синий)
кц
```

? Какие вопросы возникли?

# Линия

толщина линии (px)

цвет

цвет сохраняется, пока не сменят

```
перо(1, синий)
линия(10, 20, 15, 20)
линия(15, 20, 15, 30)
линия(15, 30, 10, 30)
линия(10, 30, 10, 20)
```

? Что это?

(10,20) (15,20)

? Какого цвета остальные линии?


(10,30) (15,30)

# Прямоугольник

```
кисть(прозрачный)
```

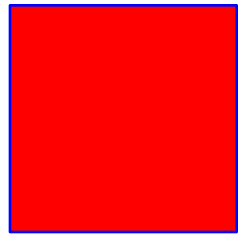
только рамка

```
перо(1, синий)
кисть(красный)
прямоугольник(20, 10, 40, 30)
```

цвет заливки

(20,10) — $(x,y)$ левого верхнего угла

$(x,y)$ правого нижнего угла




(40,10)

(20,30)

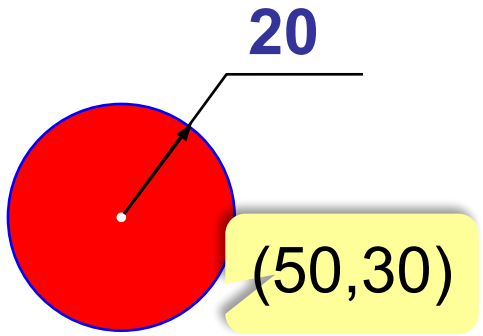
(40,30)

? Координаты остальных углов? размеры?

# Окружность

? Какие данные нужны?

```
перо(1, синий)
кисть(красный)
окружность(50, 30, 20)
```

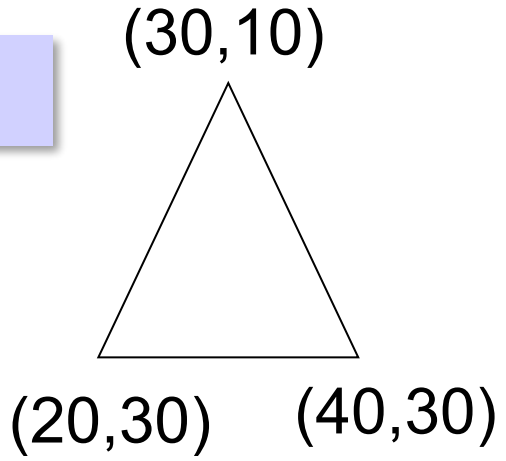



$(x,y)$ центра

радиус

# Ломаная

? Из каких примитивов состоит?



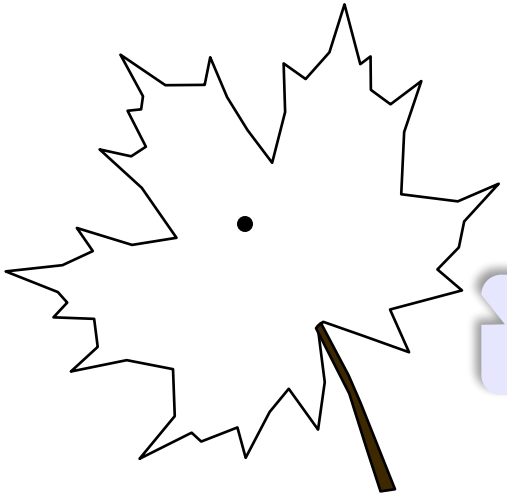

```
линия(20, 30, 30, 10)
линия(30, 10, 40, 30)
линия(40, 30, 20, 30)
```

или так

```
в точку(20, 30)
линия в точку(30, 10)
линия в точку(40, 30)
линия в точку(20, 30)
```

? Что лучше?

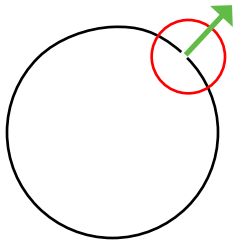
# Заливка области



```
кисть(оранжевый)
залить(50, 60)
```

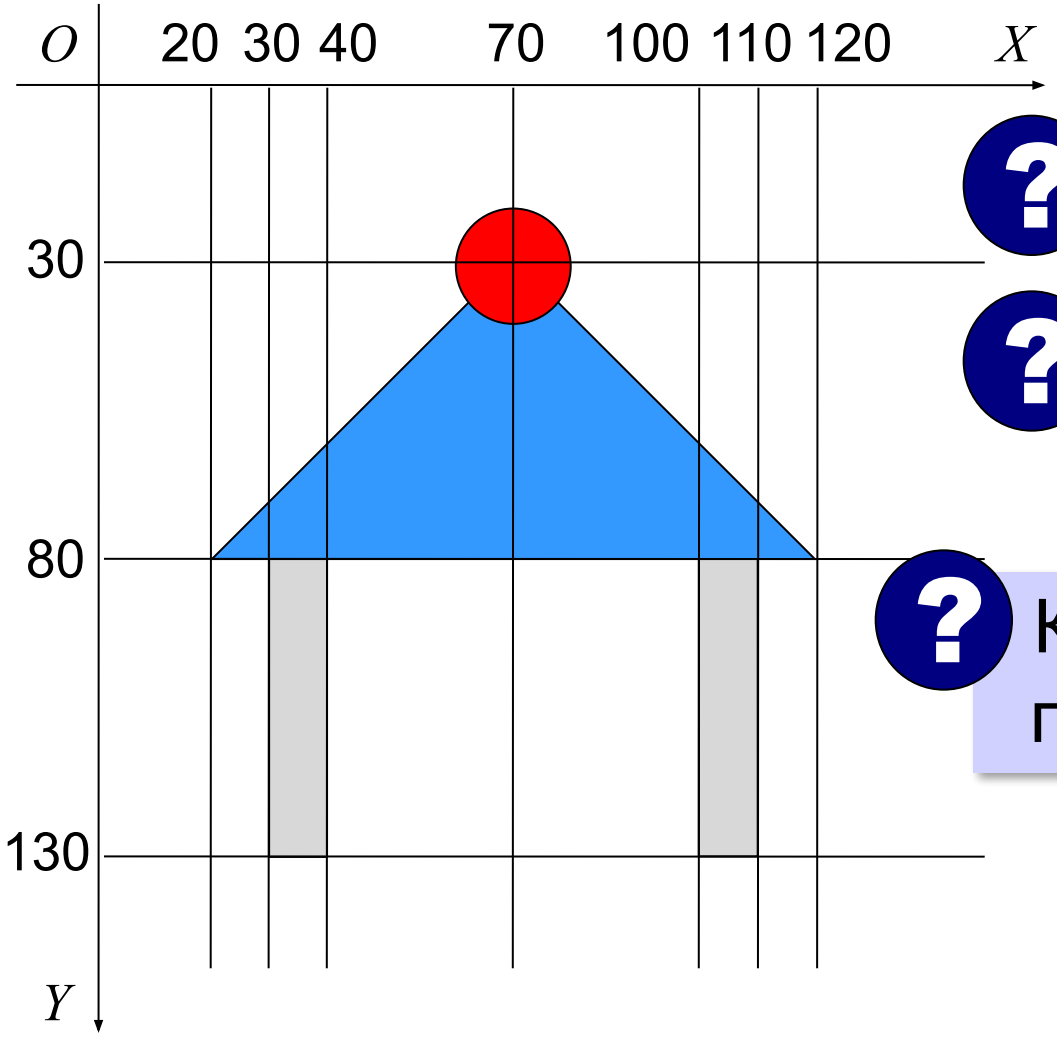
(50,60)

- можно начать с любой точки внутри области
- заливаются все соседние пиксели одного цвета
- заливка прекращается на границе другого цвета



? Если линия не замкнута?

# Пример



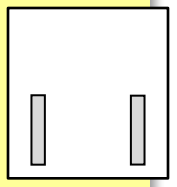

? Из каких фигур?

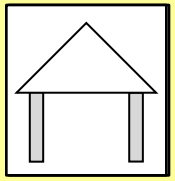
? В каком порядке рисовать?

? Координаты углов прямоугольников?

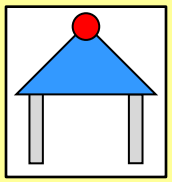
# Пример

```
алг Беседка
нач
   новый лист(200, 200, белый)          | 1
   перо(1, черный)                      | 2
   кисть(серый)                         | 3
   прямоугольник(30, 80, 40, 130)       | 4
   прямоугольник(100, 80, 110, 130)     | 5
   в точку(20, 80)                      | 6
   линия в точку(70, 30)                | 7
   линия в точку(120, 80)               | 8
   линия в точку(20, 80)                | 9
   кисть(синий)
   залить(70, 70)
   кисть(красный)                       | 12
   окружность(70, 30, 10)               | 13
кон
```





? Можно переставить?



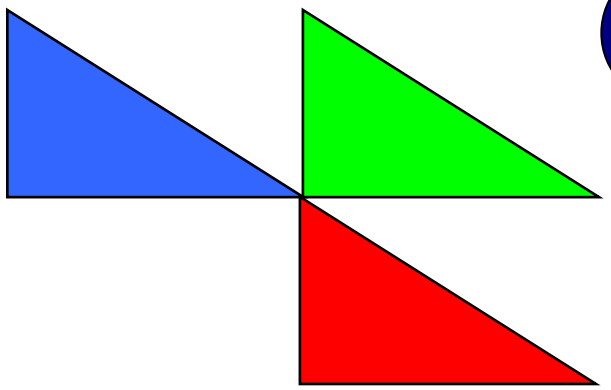
# Алгоритмы и программирование

## § 42. Вспомогательные алгоритмы

# Зачем это нужно?

Задача:



? Можно ли решить известными методами?

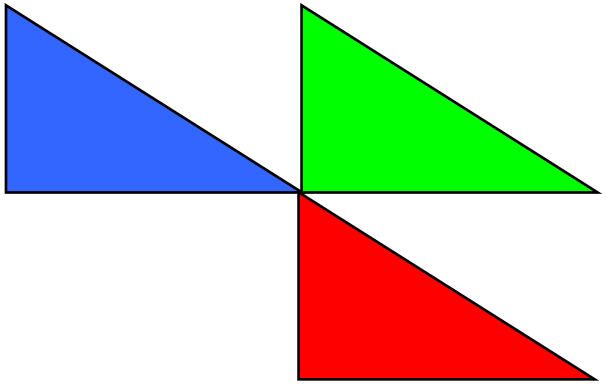
? Что особенного?

Особенность: три похожие фигуры

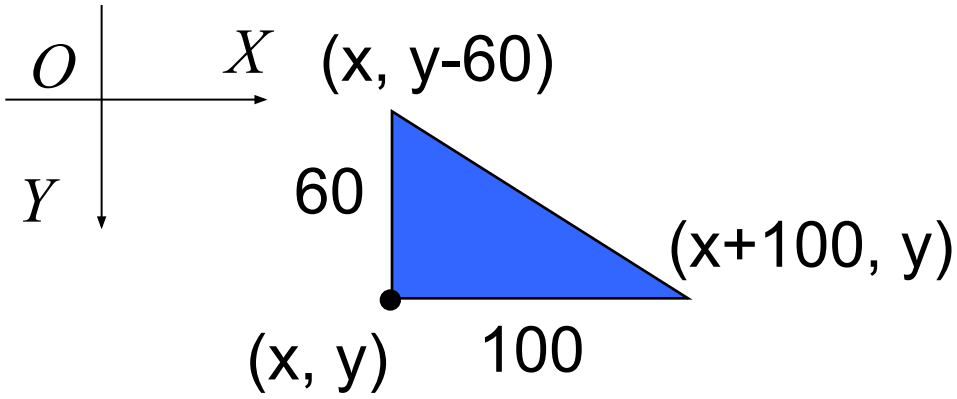
! Идея: научить Рисователя рисовать такие треугольники!

# Что такое вспомогательный алгоритм?

**Вспомогательный алгоритм (процедура)** — это новая команда, которую мы «учим» выполнять исполнителя.



? Координаты углов?



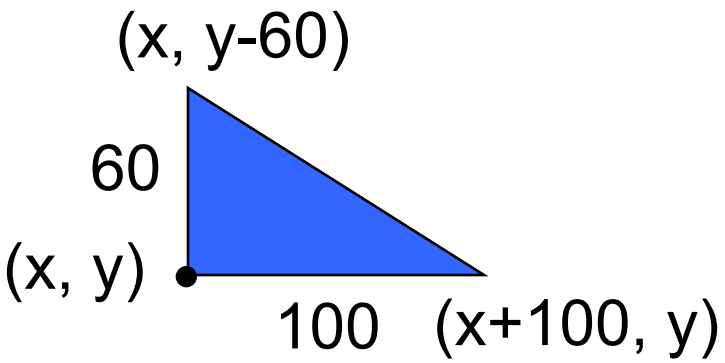

? Что общего?
Что отличается?

общее: размеры, угол поворота

отличия: **координаты**, **цвет**

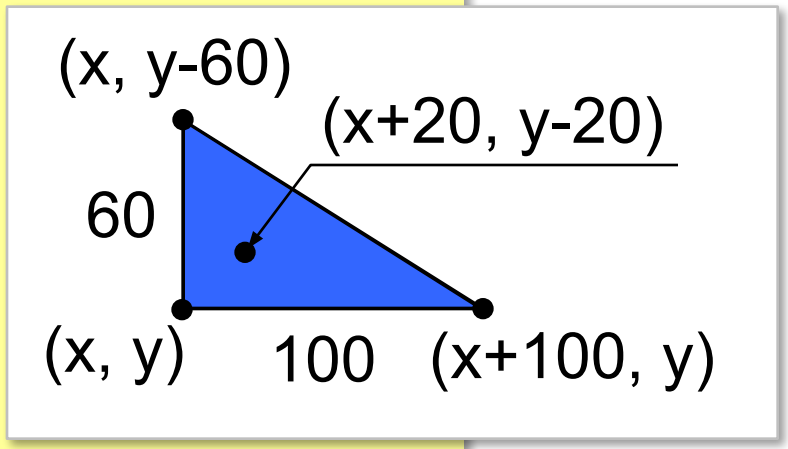
базовая точка

# Составляем вспомогательный алгоритм



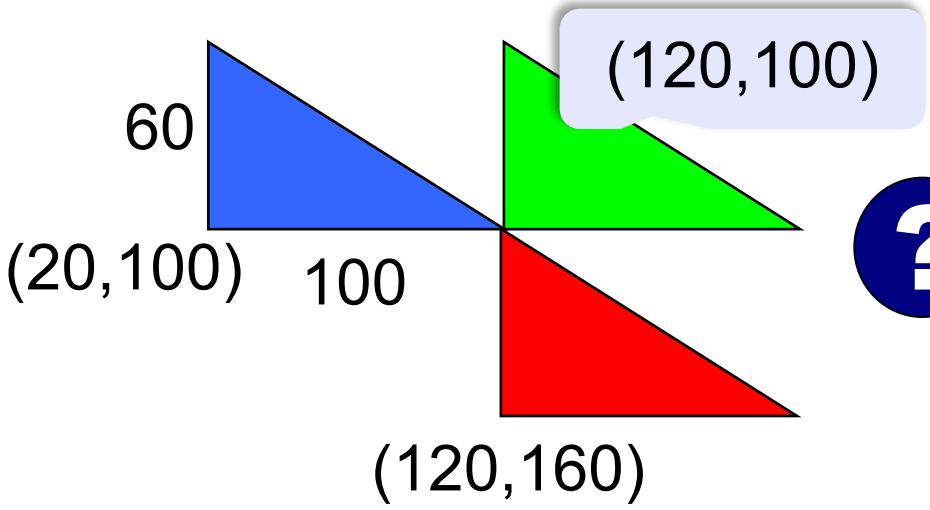

**Параметры** — это данные необходимые для работы процедуры.

```
алг треугольник
нач
   в точку(x, y)
   линия в точку(x, y-60)
   линия в точку(x+100, y)
   линия в точку(x, y)
   кисть(ц)
   залить(x+20, y-20)
кон
```




? Какую строку можно изменить?

# Основная программа




? Какие данные нужны для вызова процедуры?

? Что если запустить?

```
алг Трио
нач
   треугольник(20, 100, синий)
   треугольник(120, 100, зеленый)
   треугольник(120, 160, красный)
кон
```

# Полная программа

```
использовать Рисователь
алг Трио
нач
  треугольник(20, 100, синий)
  треугольник(120, 100, зеленый)
  треугольник(120, 160, красный)
кон
```

основная программа

```
алг треугольник (цел x, y, цвет ц)
нач
  в точку(x, y)
  линия в точку(x, y-60)
  линия в точку(x+100, y)
  линия в точку(x, y)
  кисть(ц)
  залить(x+20, y-20)
кон
```

вспомогательный алгоритм (процедура)

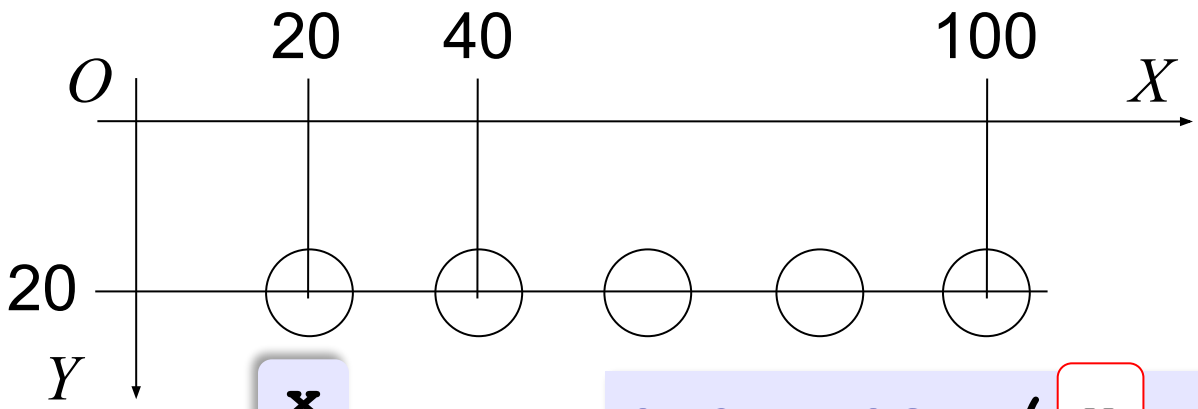
# Алгоритмы и программирование

## § 43. Применение циклов

# Рисование с помощью циклов

Задача:




x

```
окружность( x , 20, 10)
```

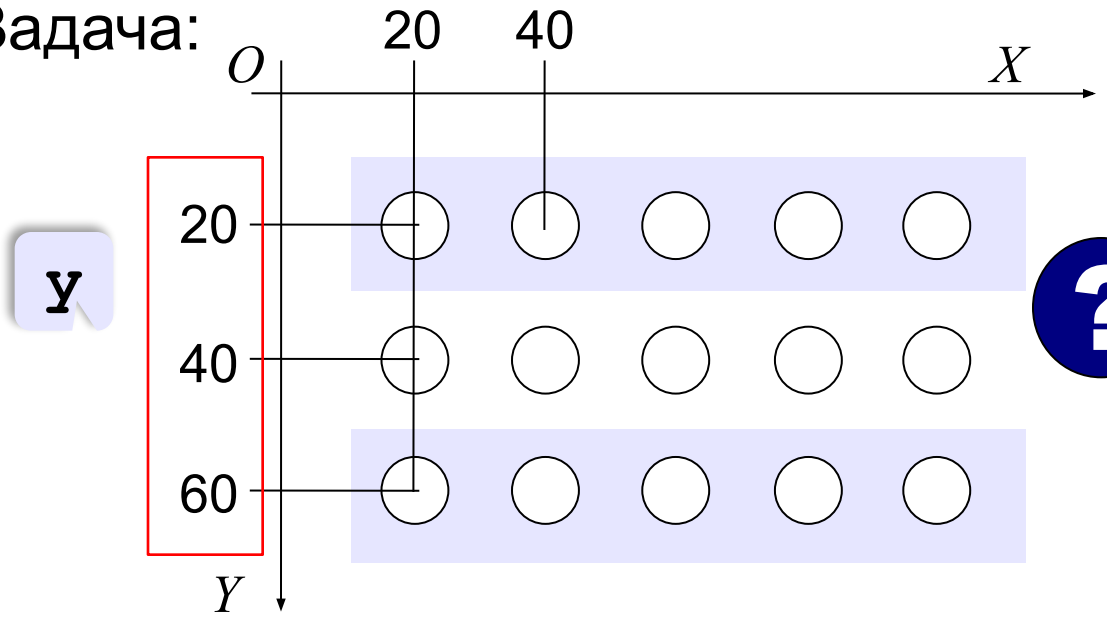
```
окружность( 20, 20, 10)
окружность( 40, 20, 10)
окружность( 60, 20, 10)
окружность( 80, 20, 10)
окружность(100, 20, 10)
```

шаг изменения переменной цикла

```
цел x
нц для x от 20 до 100 шаг 20
   окружность(x, 20, 5)
кц
```

# Рисование с помощью циклов

Задача:




? Чем отличаются ряды?

```
Ряд(20)
Ряд(40)
Ряд(60)
```

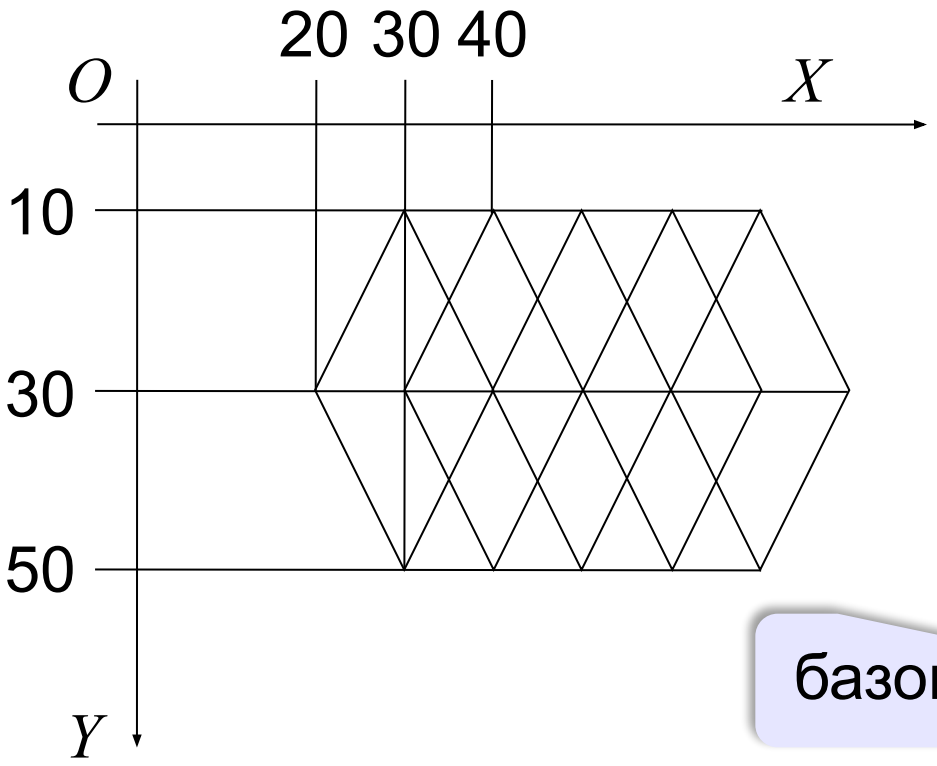
→

```
цел y
нц для y от 20 до 60 шаг 20
   Ряд(y)
кц
```

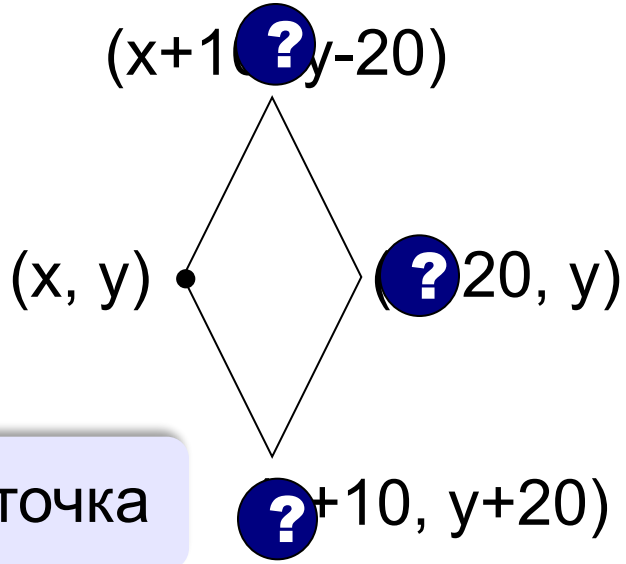
! Нужно добавить процедуру Ряд!

# Использование процедур

Задача:

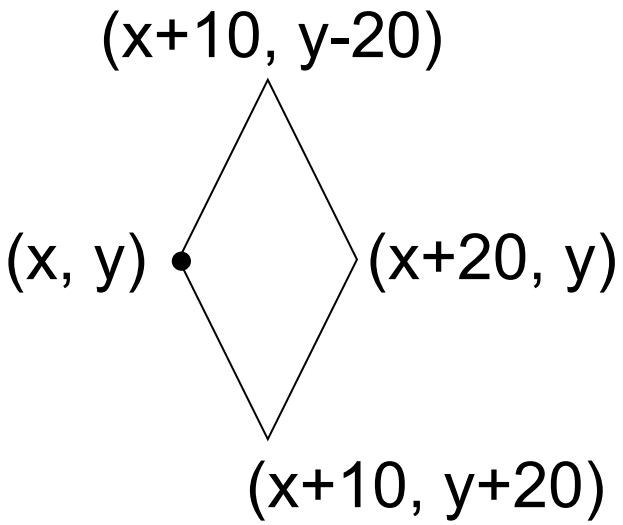



? Из каких фигур состоит?




базовая точка

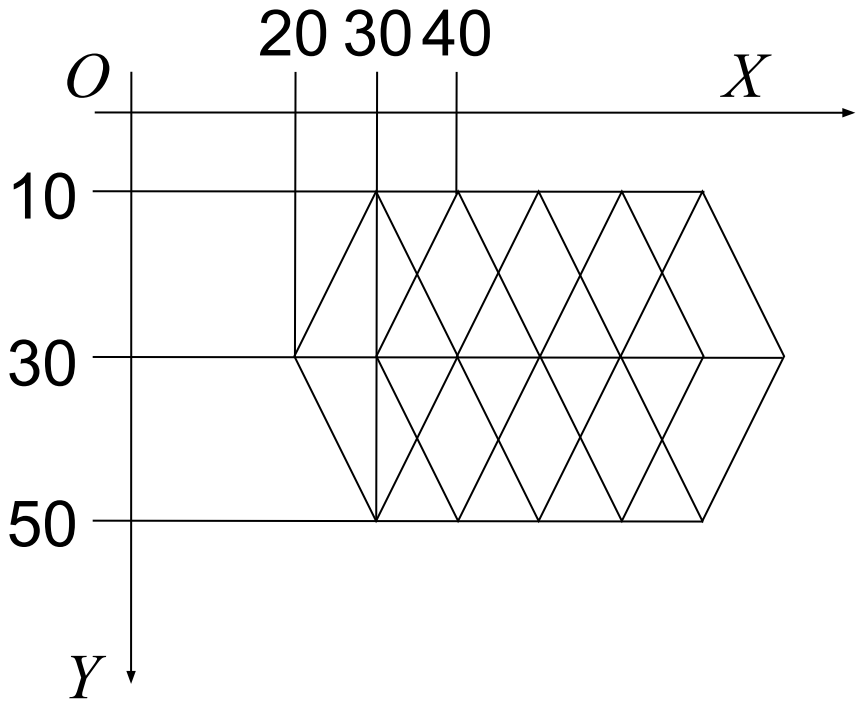
# Процедура Ромб




```
алг Ромб(цел x, y)
нач
  в точку(x, y)
  линия в точку(x+10, y-20)
  линия в точку(x+20, y)
  линия в точку(x+10, y+20)
  линия в точку(x, y)
кон
```

? Зачем 2 параметра (**y** не изменяется)?

# Вызов процедуры в цикле

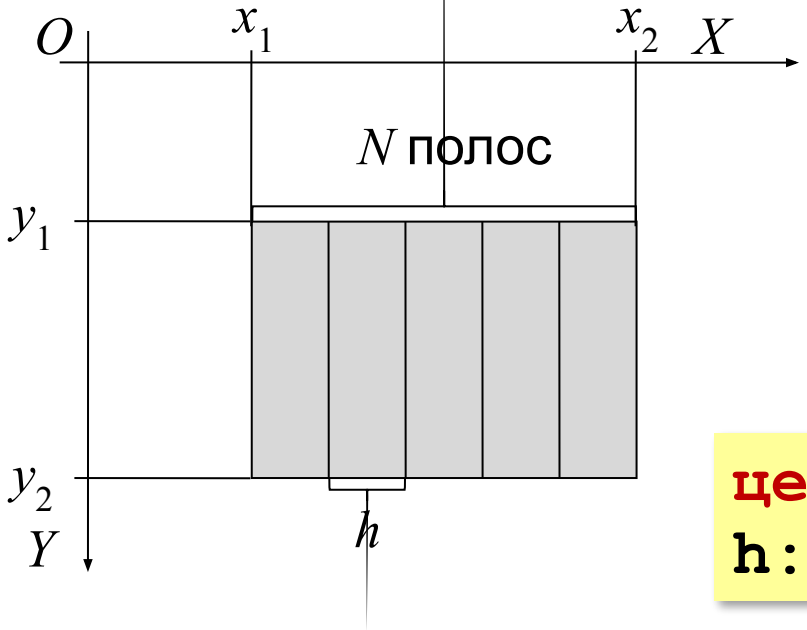



```
Ромб(20, 30)
Ромб(30, 30)
Ромб(40, 30)
Ромб(50, 30)
Ромб(60, 30)
```

```
цел y
нц для y от 20 до 60 шаг 10
   Ромб(y, 30)
кц
```

# Штриховка

Задача:




? Как найти $h$?

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

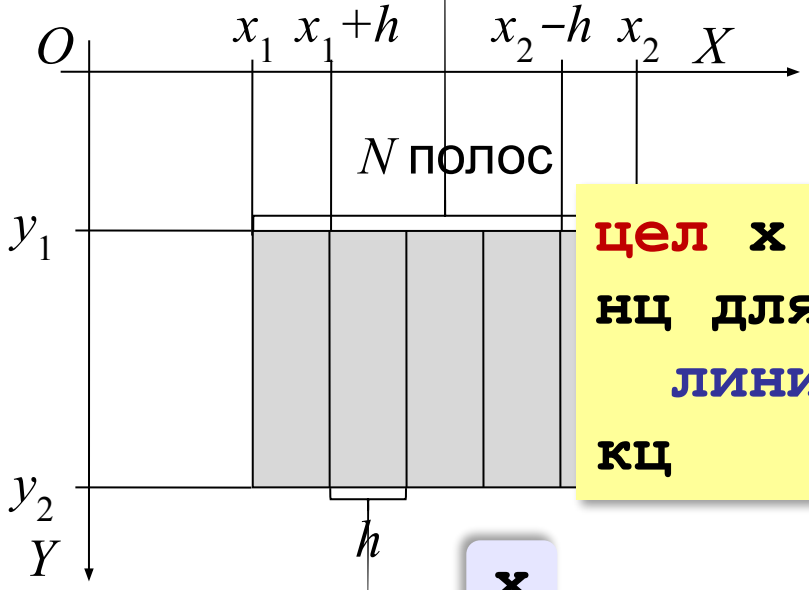
```
цел h
h:= div(x2-x1, N)
```

деление нацело

```
цел x1=100, x2=300
цел y1=100, y2=200
кисть(серый)
прямоугольник(x1, y1, x2, y2)
```

# Штриховка

Задача:




```
цел x
нц для x от x1+h до x2-h шаг h
   линия(x, y1, x, y2)
кц
```

x x

```
линия(x1+h,   y1, x1+h,   y2)
линия(x1+2*h, y1, x1+2*h, y2)
...           ...
линия(x2-h,   y1, x2-h,   y2)
```

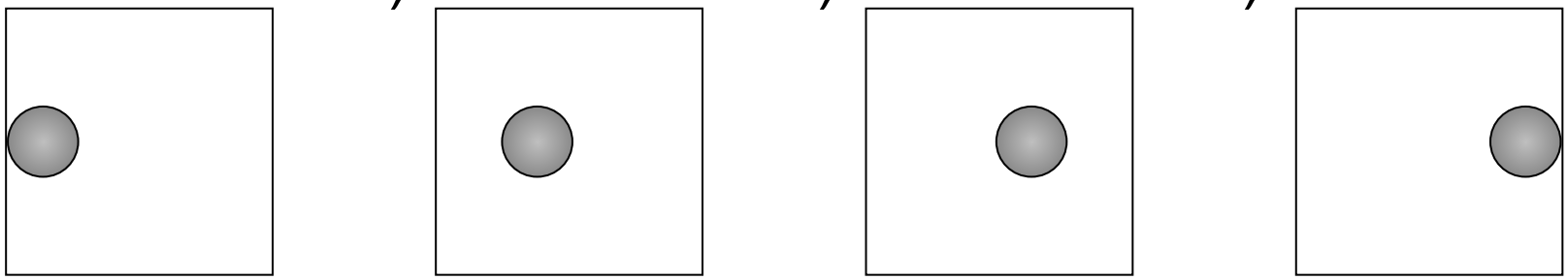
# Алгоритмы и программирование

## § 44. Анимация

# Как сделать анимацию?

**Анимация** — это быстрая смена изображений («кадров») на экране.

а) б) в) г)

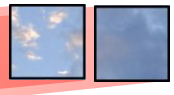


? Насколько быстрая?

≥ 16 кадров/с

# Перемещение шарика на фоне


Где-то сохранить:



- запомнить фон под шариком
- нарисовать шарик

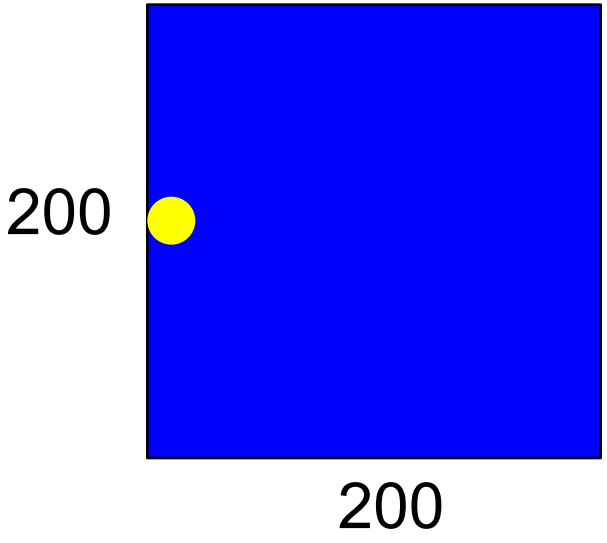
**Переместить**:

- восстановить фон под шариком (стереть)
- изменить координаты
- запомнить фон под новым местом
- нарисовать шарик

? Если фон одноцветный?

не нужно запоминать!

# Начало программы




```
использовать Рисователь
алг Движение
нач
   новый лист(200, 200, синий)
   перо(1, прозрачный)
   ...
кон
```

шарик без контура!

# Новая команда

параметры

```
алг Шарик(цел x, y, цвет ц)
нач
   цел R=10
   кисть(ц)
   окружность(x, y, R)
кон
```

? Что такое 10?

показываем!

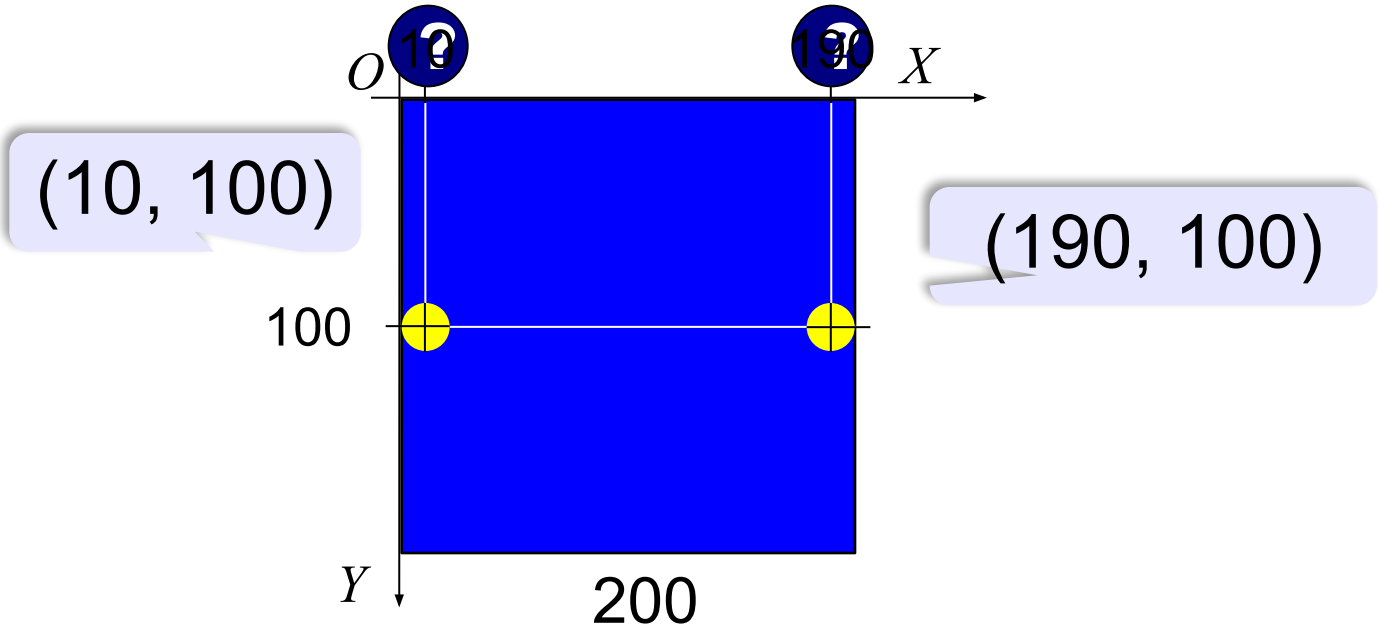
```
Шарик(20, 100, желтый)
```

? В чём отличие?

```
Шарик(20, 100, синий)
```

стираем!

# Основной цикл




**Начальное положение**:

```
цел x=10, y=100
```

**Цикл**:

```
нц пока x < 190
   ... | двигаем шарик
кц
```

? Когда закончится?

# Как двигать шарик?

```
нц пока x < 190
                                  | рисуем шарик
                                  | стираем
                                  | перемещаем
кц
```


Что плохо?

```
нц пока x < 190
   Шарик(x, y, жёлтый)    | рисуем шарик
   ждать(20)              | смотрим 20 мс
   Шарик(x, y, синий)     | стираем
   x:= x + 2              | перемещаем
кц
```

# В чём проблема?

```
нц пока x < 190
   Шарик(x, y, жёлтый)
   Шарик(x, y, синий)
   ждать(20)
   x:= x + 2
кц
```

почти не видно!

```
нц пока x < 190
   Шарик(x, y, жёлтый)
   ждать(20)
   x:= x + 2
   Шарик(x, y, синий)
кц
```

стирает в другом месте!

# Алгоритмы и программирование

## § 45. Управление с помощью клавиатуры

# Команды для работы с клавиатурой

## Определить, нажата ли клавиша:

да/нет

Это **условный оператор**!

```
если сигнал клав то
   | клавиша нажата, что-то сделать
все
```

## Определить, какая клавиша нажата:

```
цел с
с:= код клав
если с = КЛ_ВВЕРХ то
   | передвинуть объект вверх
все
```

ждать, если не нажата

символьное имя

**константы:**
КЛ_ВВЕРХ
КЛ_ВНИЗ
КЛ_ВПРАВО
КЛ_ВЛЕВО

# Основной цикл

всегда верно!

```
нц пока да
   Шарик(x, y, жёлтый)
   с:= код клав
   Шарик(x, y, синий)
   | переместить шарик
кц
```

бесконечный цикл

? Что плохо?

# Вся программа

```
использовать Рисователь
алг Управление клавишами
нач
  новый лист(200, 200, синий)
  перо(1, прозрачный)
  цел x=100, y=100, c
  нц пока да
    Шарик(x, y, жёлтый)
    c:= код клав
    Шарик(x, y, синий)
    если c=КЛ_ВЛЕВО  то x:=x-5 все
    если c=КЛ_ВПРАВО то x:=x+5 все
    если c=КЛ_ВВЕРХ  то y:=y-5 все
    если c=КЛ_ВНИЗ   то y:=y+5 все
  кц
кон
```

? Чего не хватает?

# Управление по требованию

События на экране развиваются и без действий игрока, но он может вмешаться.

```
нц пока да
  | если нажата клавиша, то
  |   изменить направление движения
  | нарисовать шарик
  | ждать 20 мс
  | стереть шарик
  | переместить шарик
кц
```

работает и без игрока!

# Обработка нажатия клавиши

если нажата какая-нибудь клавиша

```
если сигнал клав то
   с:= код клав
   | изменить направление движения
все
```

? Как изменить направление?

**Движение**:

```
x:= x + dx
y:= y + dy
```

сдвиг по x

сдвиг по y

? Как остановить?

```
если с=КЛ_ВЛЕВО то dx:=-5; dy:=0 все
если с=КЛ_ВПРАВО то dx:=5; dy:=0 все
...
если с=КЛ_ПРОБЕЛ то dx:=0; dy:=0 все
```

# Конец фильма

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

**ЕРЕМИН Евгений Александрович**
к.ф.-м.н., доцент кафедры мультимедийной дидактики и ИТО ПГГПУ, г. Пермь
eremin@pspu.ac.ru

# Источники иллюстраций

1. авторские материалы