

Начала информатики.

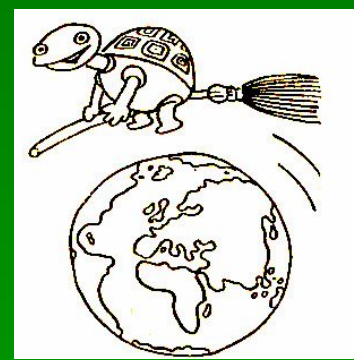


# Алгоритмический язык программирования ЛОГО.

Методическая разработка  
учителя информатики  
школы № 92  
Солодовниковой Т. Я.

# Цели:

1. Познакомить с понятиями алгоритм и программа.
2. Познакомить с командами алгоритмического языка ЛОГО.
3. Пробудить желание экспериментировать.



# Задачи:

1. Научить общению с компьютером в режиме диалога на языке ЛОГО.
2. Научить основам программирования на алгоритмическом языке ЛОГО.
3. Развивать алгоритмическое мышление.

# Темы уроков.

Урок 1. Введение. ▶

Урок 2. Углы и их измерение. ▶

Урок 3. «Черепашья графика». ▶

Урок 4. Циклы. ▶

Урок 5. Построение правильных многоугольников. ▶

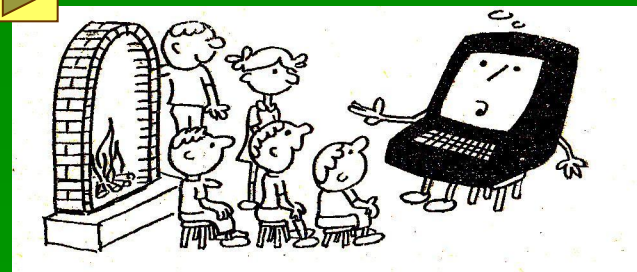
Урок 6. Построение окружности и дуги. ▶

Урок 7. Процедуры. ▶

Урок 8. Процедуры с параметрами. ▶

Урок 9. Рекурсивные процедуры. ▶

Урок 10. Координаты. ▶



# Думаем и исследуем на ЛОГО.



# Урок 1. Введение.



# Немного истории...

Язык программирования ЛОГО был разработан в конце 60-х годов американским исследователем в области искусственного интеллекта **С. Пейпертом**.

Название языка происходит от греческого слова «**LOGOS**» - «слово», «мысль», «смысл», «идея».

В языке используется специальный исполнитель команд  
«**черепашка**».



Она точно выполняет все команды учащегося.

# Алгоритм и программа - основные понятия информатики.

**Алгоритм** – конечная последовательность шагов в решении задачи, приводящая от исходных данных к требуемому результату.

*Алгоритм можно записать в виде нумерованного списка (словесная форма) или изобразить с помощью блок-схемы.*

**Программа** – это алгоритм, записанный на языке программирования.

Уметь **программировать** – значит строить алгоритм, т. е. последовательность команд, приводящих к нужному результату, и правильно записывать команды на языке программирования.

# Базовые команды черепашки.

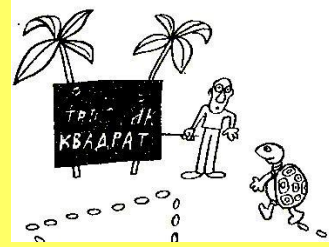
**Черепашку** можно научить многим командам, но она имеет и некоторый начальный уровень образованности и выполняет следующие базовые команды

**вперед, назад, направо, налево.**





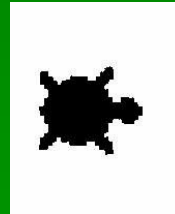
# Запомни:



- Для выполнения команд ***вперед*** и ***назад*** необходимо задать черепашке *количество шагов* (число).
- Для выполнения команд ***направо*** и ***налево*** необходимо указать черепашке, *на сколько градусов повернуться* (число), считая от её текущего положения.
- Между командой и числом ставится ***пробел***.

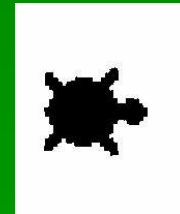
# Выполним:

**вперед 100**



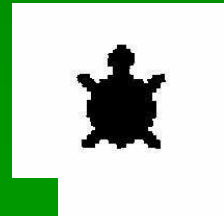
# Выполним:

назад 100



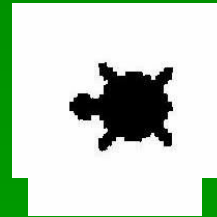
# Выполним:

направо 90



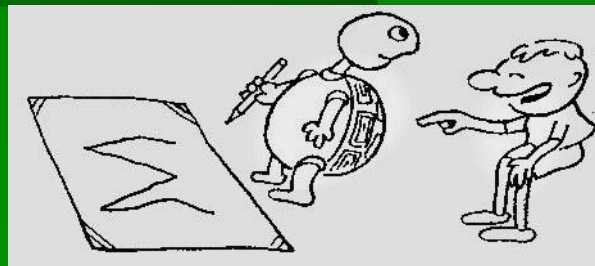
# Выполним:

налево 90

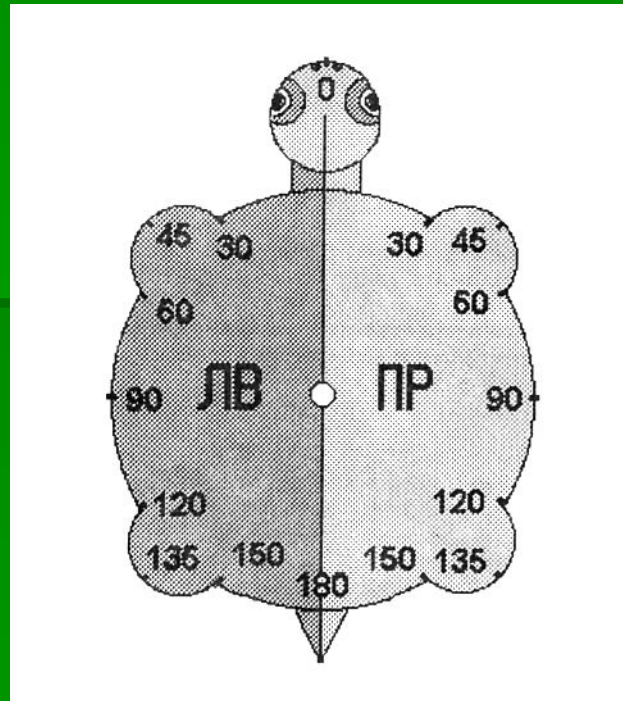


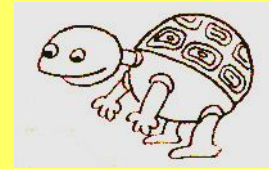
# Урок 2.

## УГЛЫ И ИХ ИЗМЕРЕНИЕ.



Прежде чем попросить черепашку двигаться, нужно обратить внимание на её **ориентацию** и **положение**.





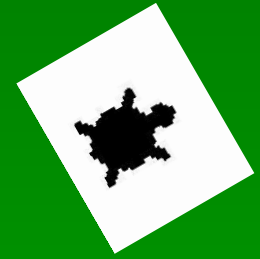
# Запомни!

Ориентацией черепашки называют то направление, куда устремлён её носик.

Положение черепашки – это та точка экрана, в которой она находится.

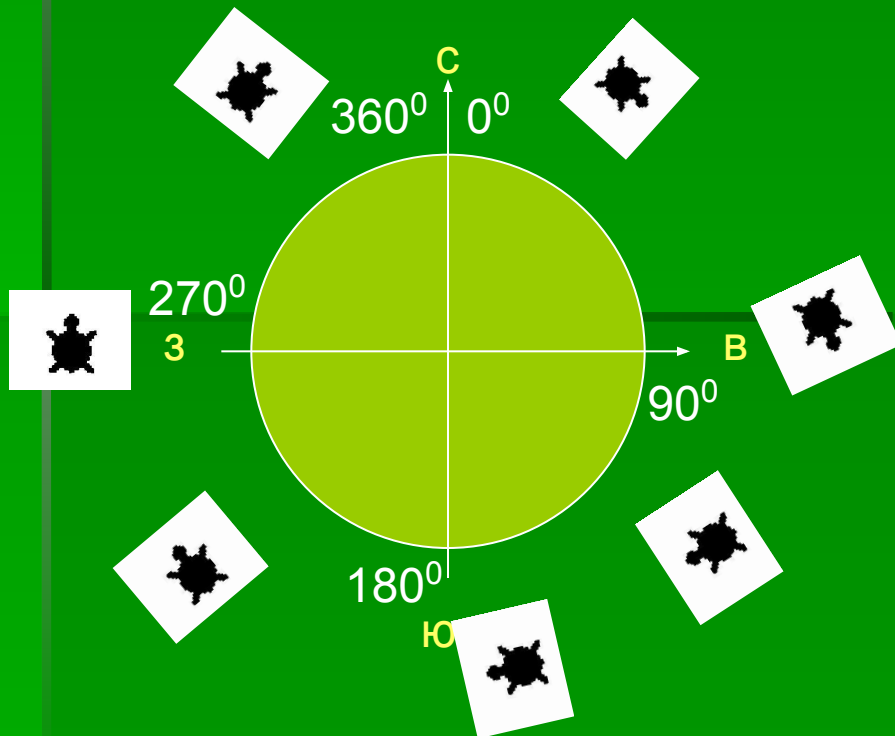


# Черепашка, повернись!

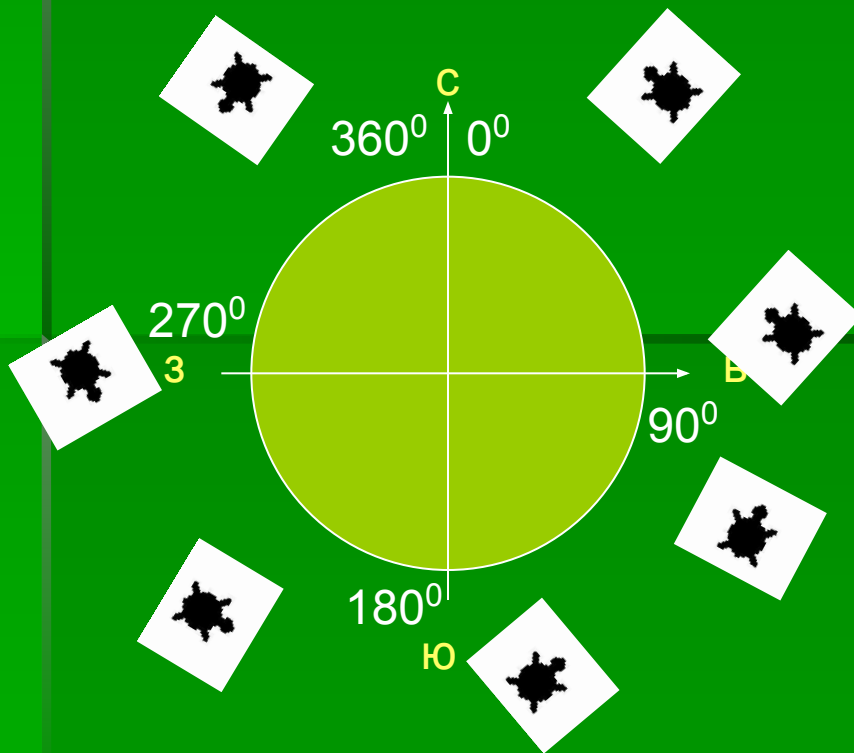


Черепашка двигаясь по экрану,  
должна выдержать заданное направление.  
Поэтому перед движением обязательно  
задаётся направление движения.

# Направо



# Налево



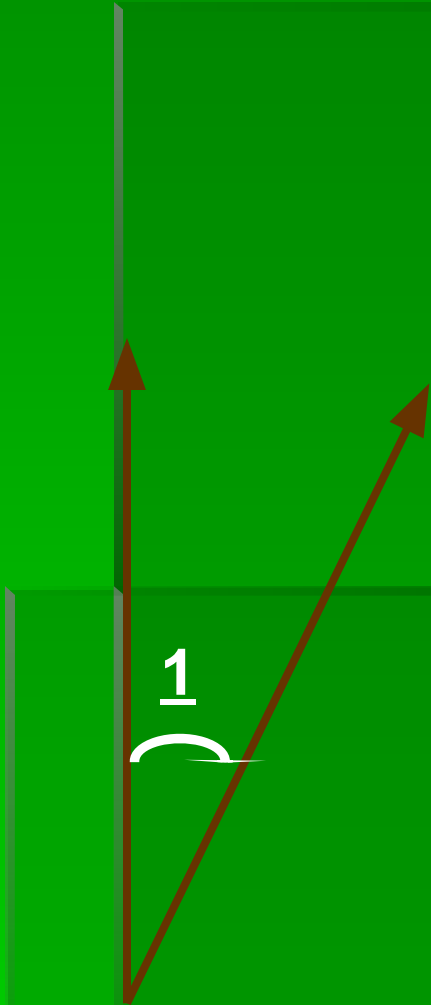
Измерить можно - длину отрезка на листе бумаги,

- расстояние между городами,
- время,
- скорость движения поезда,
- силу электрического тока,
- влажность воздуха,
- громкость звука,
- температуру окружающей среды,
- освещённость помещения и т. д.

**Сегодня мы научимся измерять угловые величины.**

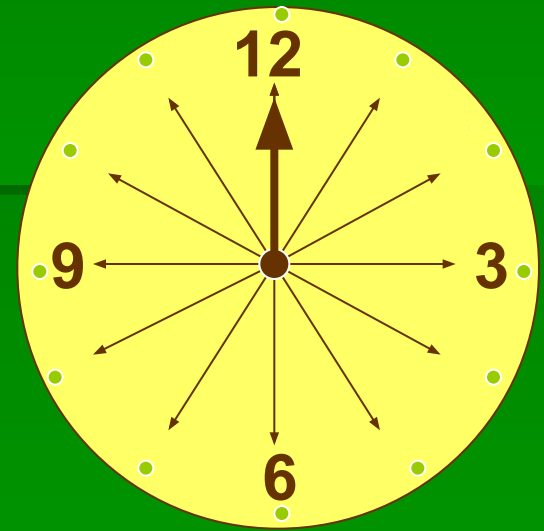
**Повернём стрелку направо и отметим её новое положение. Между исходной и повернутой стрелками образовался угол.**

**Этот угол обозначен цифрой 1.**



Чтобы точно сравнивать углы и определять их величину, надо научиться их измерять.

Посмотрим на часы:

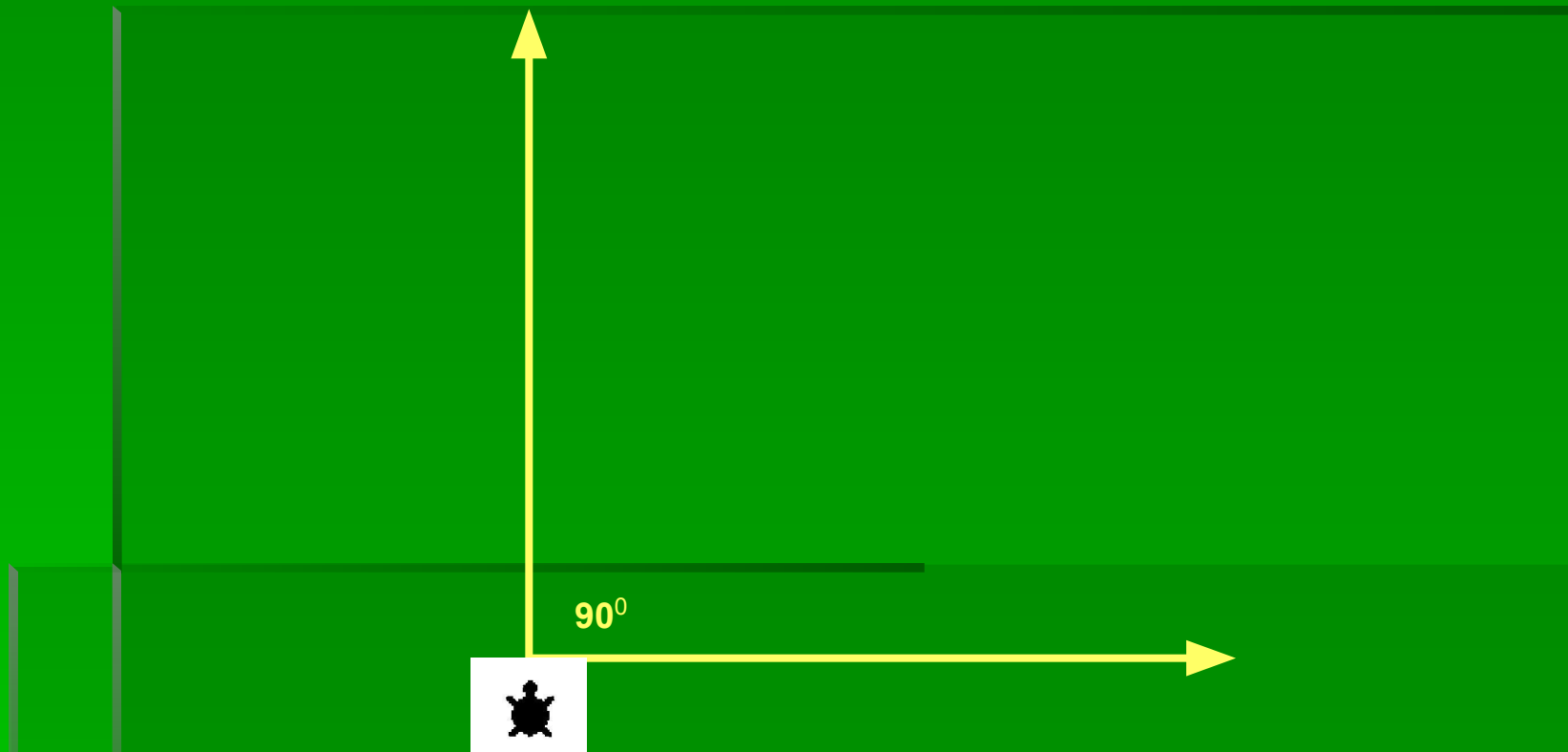
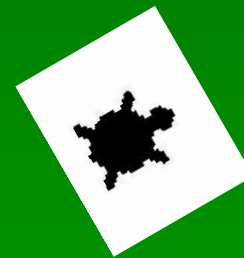


По циферблату бежит секундная стрелка. Условимся считать начальным положением стрелки самую верхнюю точку циферблата, отмеченную числом **12**.

Стрелка пробежала от начального положения полный круг и оказалась опять в точке циферблата, отмеченной числом **12**.

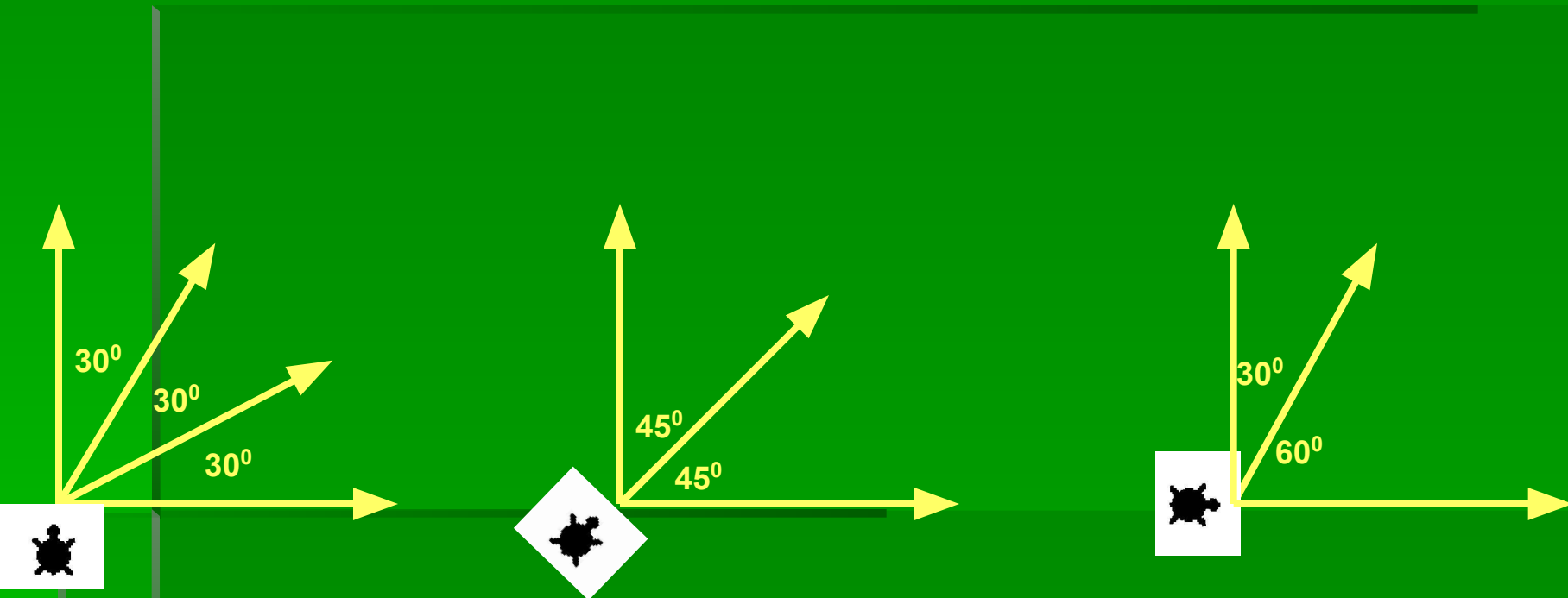
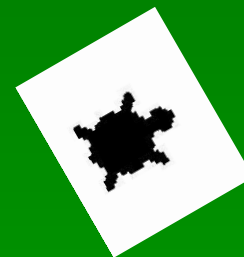
Такой угол, который соответствует полному обороту стрелки, считают равным **360 градусам**.

# УГЛЫ БЫВАЮТ:



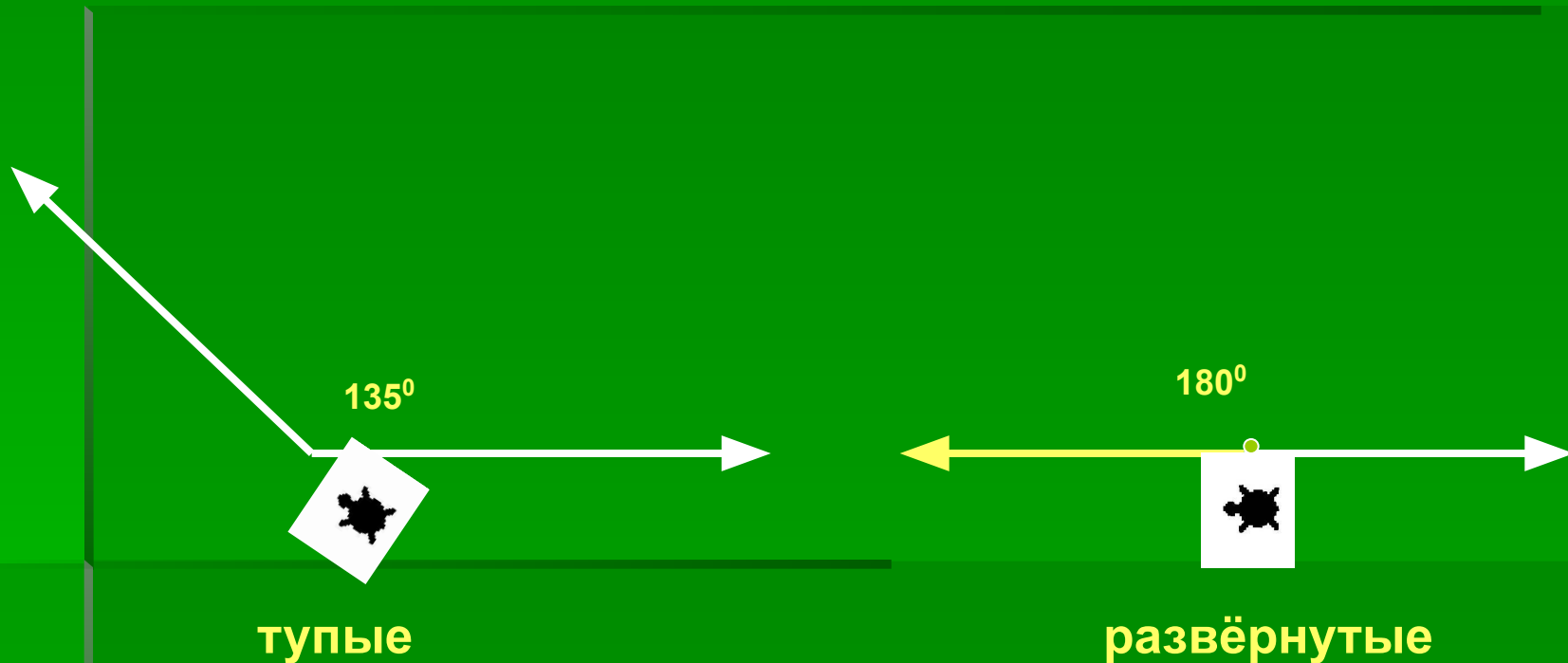
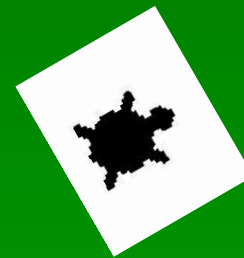
прямые

# УГЛЫ БЫВАЮТ:



острые

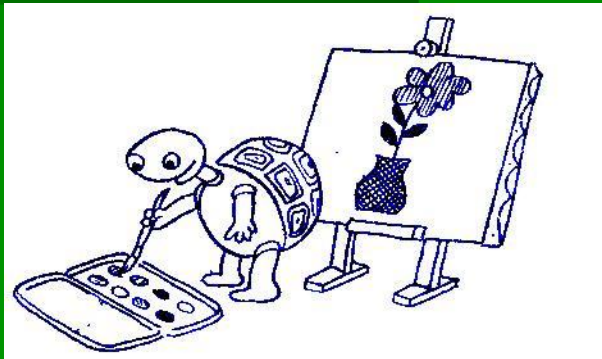
# УГЛЫ БЫВАЮТ:





# Урок 3.

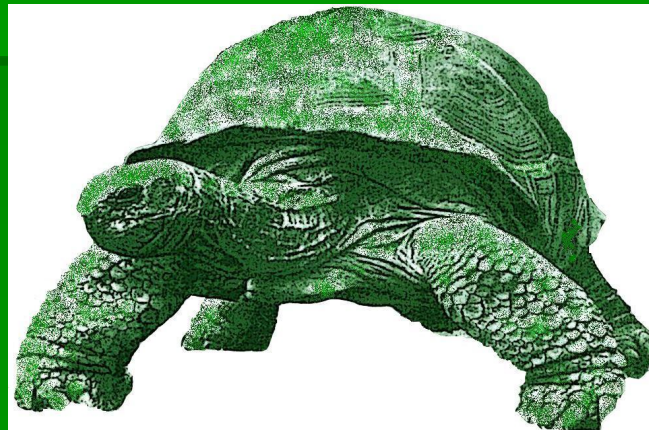
## «Черепашья» графика.



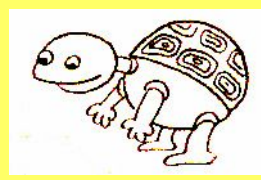
Язык **ЛОГО** имеет большие графические  
ВОЗМОЖНОСТИ.

Перемещаясь по экрану компьютера черепашка  
может оставлять след, подобно живой черепахе,  
которая волочит свой хвост по песку.

Эту её способность мы будем использовать, чтобы  
чертить фигуры на экране.



# Запомни:



□ Черепашка может двигаться по экрану оставляя **след**, а может и не оставлять **след**.

□ Чтобы черепашка оставляла след нужно дать команду

**Перо Опустити – ПО**

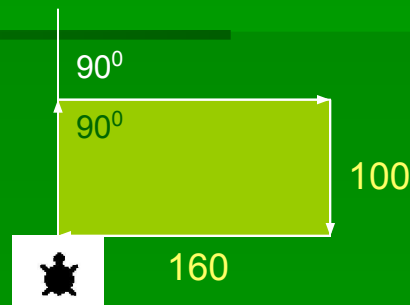
□ Чтобы черепашка не оставляла след нужно дать команду

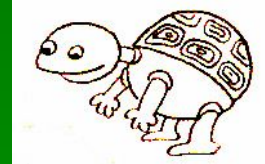
**Перо Подними – ПП**

## Задание 1.

### Научим черепаху рисовать прямоугольник.

Одна сторона этого прямоугольника будет **100** шагов, другая - **160** шагов. Все углы у прямоугольника - **прямые**, значит повороты черепашки при переходе со стороны на сторону будут **90** градусов.





Составим алгоритм в виде нумерованного списка  
(словесная форма):

1. Начало.
2. Вперёд на величину ширины прямоугольника.
3. Поворот на  $90^0$  вправо.
4. Вперёд на величину длины прямоугольника.
5. Поворот на  $90^0$  вправо.
6. Вперёд на величину ширины прямоугольника.
7. Поворот на  $90^0$  вправо.
8. Вперёд на величину длины прямоугольника.
9. Поворот на  $90^0$  вправо.
10. Конец.

# Черепаха, шагом марш!

Напишем программу на языке ЛОГО:

```
по  
вперед 100  
направо 90
```

```
вперед 160  
направо 90
```

```
вперед 100  
направо 90
```

```
вперед 160  
направо 90
```

## Задание 2.

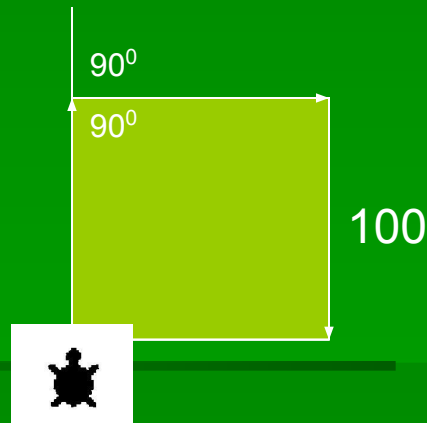
А теперь нарисуем квадрат. У квадрата все стороны равны. Сторона квадрата будет 100 шагов.

по  
вперед 100  
направо 90

вперед 100  
направо 90

вперед 100  
направо 90

вперед 100  
направо 90



## Программировать – значит и редактировать.

При построении прямоугольника и квадрата черепашка  
двигалась,  
поворачиваясь только  
**Направо.**

Теперь попросим её нарисовать такие же фигуры,  
но поворачиваясь только  
**Налево.**

Для этого отредактируем наши программы,  
т. е. заменим команду **Направо** на команду **Налево.**





### Задание 3.

Нарисуем лесенку с тремя ступенями.

Размер ступеньки по ширине и высоте равен 20.

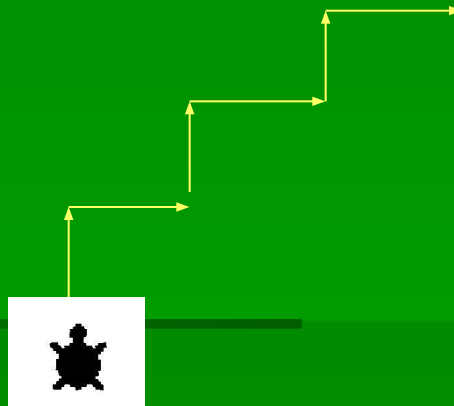
*В этом задании черепашка будет двигаться поворачиваясь **налево** и **направо**.*

по  
вперед 20  
направо 90  
вперед 20

налево 90  
вперед 20  
направо 90

вперед 20  
налево 90  
вперед 20

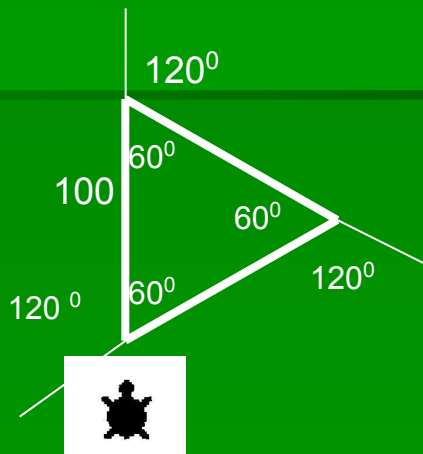
направо 90  
вперед 20  
налево 90



## Задание 4.

Построим равносторонний треугольник.  
Длина стороны 100 шагов.  
Все углы в равностороннем  
треугольнике равны  $60^{\circ}$ .

Составим алгоритм в виде блок-схемы.



## Блок-схема



# Напишем программу на языке ЛОГО:

**по**

**вперед 100  
направо 120**

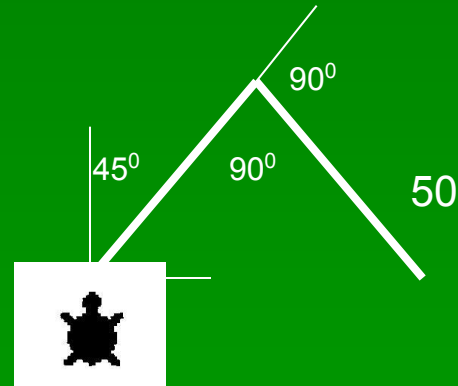
**вперед 100  
направо 120**

**вперед 100  
направо 120**

## Задание 5.

Построим уголок. Длина линии 50 шагов.

по  
направо 45  
вперед 50  
направо 90  
вперед 50  
налево 90



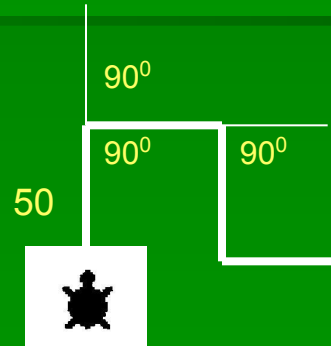
## Задание 6.

Построим фигуру. Длина линии 50 шагов. Угол поворота  $90^\circ$ .

по  
вперед 50  
направо 90

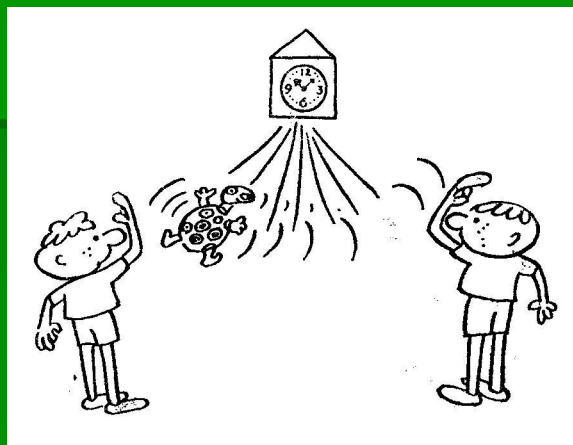
вперед 50  
направо 90

вперед 50  
налево 90  
вперед 50  
налево 90



# Урок 4.

## ЦИКЛЫ.



**Цикл – это многократное повторение  
чего – либо.**

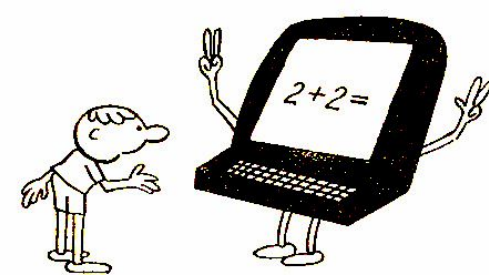
Например, повторение дней недели, повторение времён года  
и т. д.

В программировании повторяться могут только команды.

В ЛОГО – повторение некоторое количество раз одной или  
нескольких команд обеспечивает команда

**Повтори**

# Запомни:



Закл $\ddot{u}$ ч $\ddot{e}$ нная в квадратные скобки последовательность команд называется ***телом цикла.***

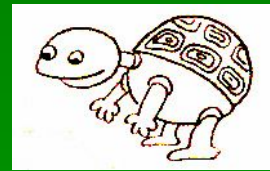
При выполнении команды цикла, тело повторяется столько раз, сколько указывает число, стоящее после слова Повтори.

Например:

Повтори 4[Вперед 100 Направо 90]



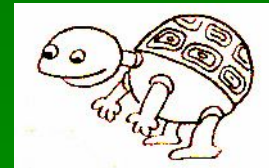
# Используем команду Повтори для построения фигур.



## Пример 1.

Для построения квадрата команда черепашке будет выглядеть так:

**повтори 4 [вперед 100 направо 90]**

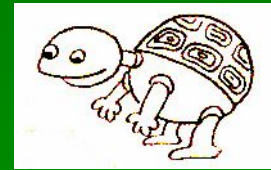


## Пример 2.

Для построения **треугольника**  
команда черепашке будет  
выглядеть так:

**повтори 3 [вперед 100 направо 120]**

### Пример 3.



Для построения **пяти уголков** команда черепашке будет выглядеть так:

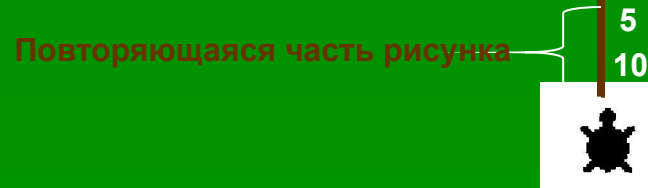
**направо 45**

**повтори 5 [вперед 50 направо 90  
вперед 50 налево 90]**

## Пример 4.

Начертим вертикальную пунктирную линию.  
Линия содержит 10 пунктиров,  
длина каждого пунктира 10,  
расстояние между пунктирами 5.

повтори 10 [по вперед 10 пп вперед 5 ]



## Пример 5.

Начертим горизонтальную пунктирную линию.

Линия содержит 10 пунктиров,

длина каждого пунктира 10, расстояние между пунктирами 5.

направо 90

повтори 10 [по вперед 10 пп вперед 5 ]



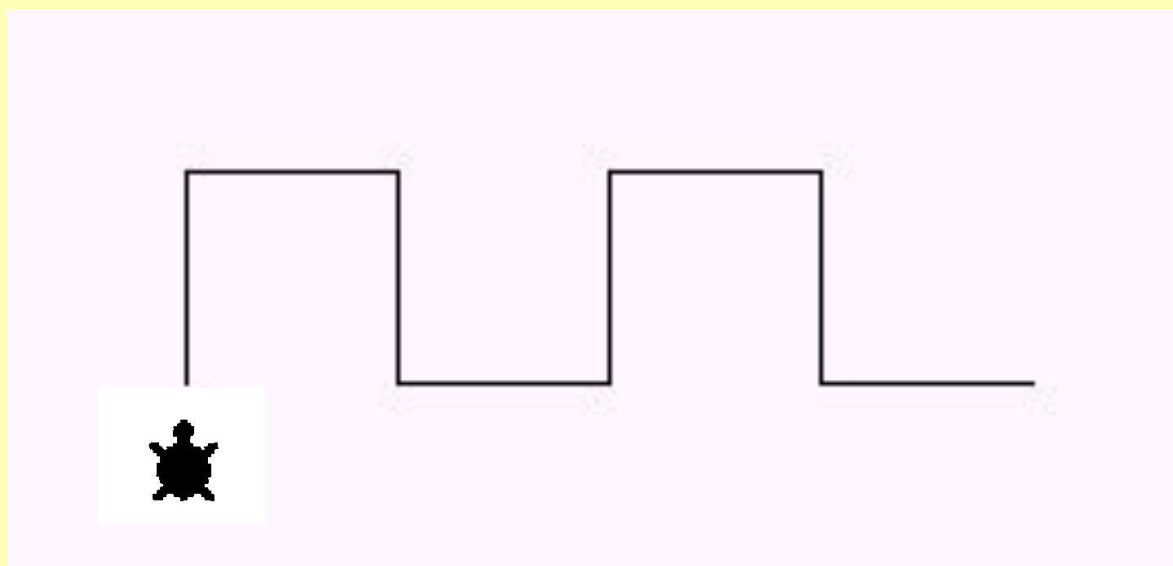
## Пример 6.

Какую фигуру начертит черепаха после выполнения команд:

ПО

повтори 2 [вперед 50 направо 90  
вперед 50 направо 90 вперед 50  
налево 90 вперед 50 налево 90]

# Черепаша начертит:



## Пример 7.

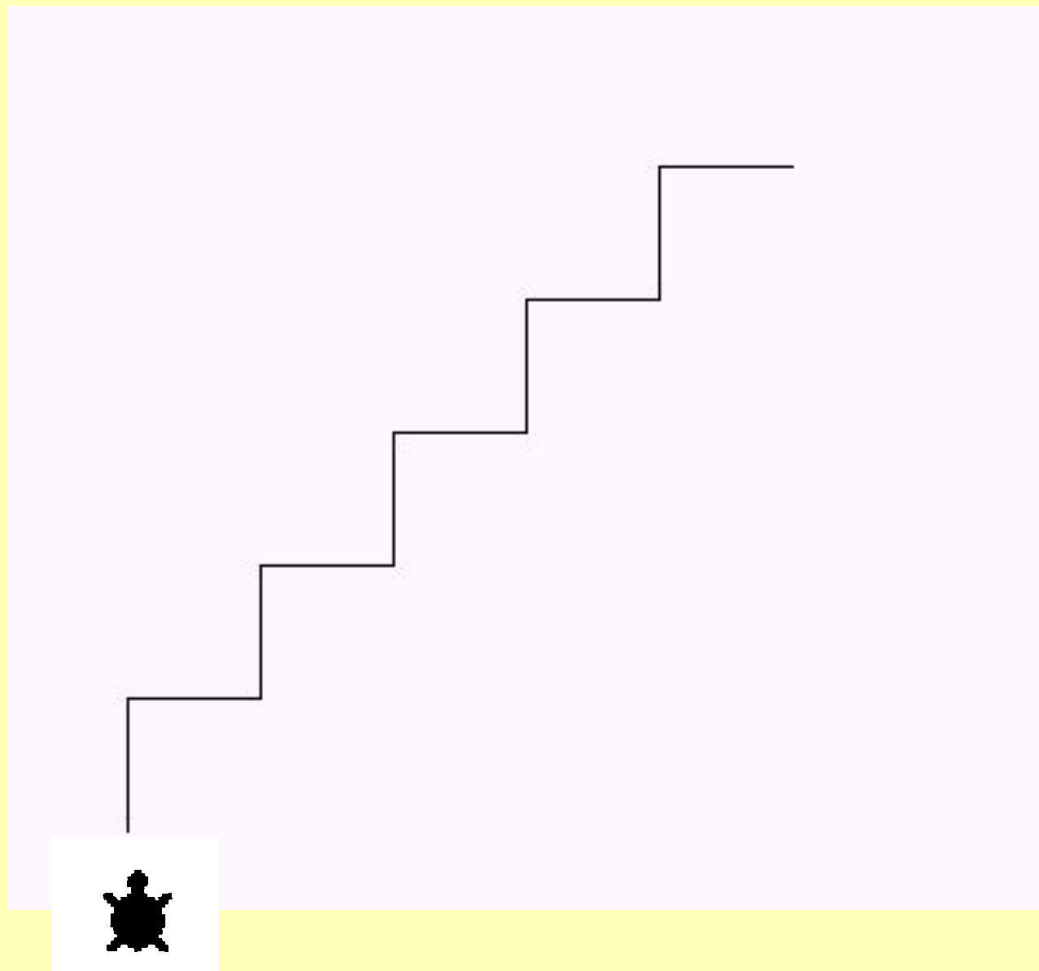
Какую фигуру начертит черепаха после выполнения команд:

ПО

повтори 5 [вперед 50 направо 90  
вперед 50 налево 90]

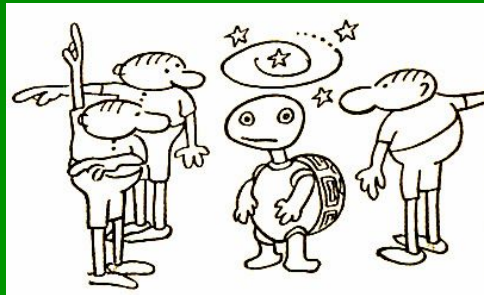


# Черепаха начертит:



# Урок 5.

## Построение правильных многоугольников.



Правильными называются  
многоугольники,  
у которых все стороны и углы равны.

Чтобы построить любой правильный  
многоугольник,  
воспользуемся командой  
**повтори.**

Получим значение угла поворота.

Для этого

$360^0$  (число градусов в полном круге)

разделим на

$N$  (число углов многоугольника)

$360/N$

# Получим значение угла поворота:

1.  $360/3=120$  – для треугольника
2.  $360/4=90$  – для прямоугольника
3.  $360/5=72$  – для пятиугольника
4.  $360/6=60$  – для шестиугольника
5.  $360/8=45$  – для восьмиугольника

# Построим фигуры.

1. Правильный треугольник:

**повтори 3 [вперед 100 направо 120]**

2. Правильный прямоугольник:

**повтори 4 [вперед 100 направо 90]**

# Построим фигуры.

3. Правильный пятиугольник:  
повтори 5[вперед 100 направо 72]

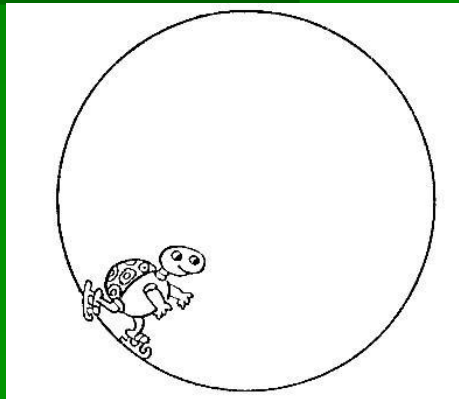
4. Правильный шестиугольник:  
повтори 6[вперед 100 направо 60]

5. Правильный восьмиугольник:  
повтори 8[вперед 100 направо 45]

⋮

# Урок 6.

## Построение окружности.





Любую окружность можно считать правильным многоугольником с очень малой стороной и соответственно очень малым углом поворота.

Для построения окружности будем использовать команду повтори.

Для значения угла поворота, равного **N**, получим формулу окружности:

**повтори 360/N [вперед 1 направо N]**

- ❖ Построим окружность в виде 360-угольника.
- ❖ Найдём угол поворота:  $360^0/360 = 1^0$
- ❖ Длину стороны возьмём за 1 шаг.
- ❖ Используем команду:

**повтори 360[вперед 1 направо 1]**

## Пользуясь формулой окружности заполним таблицы:

Уменьшение  
окружности:

Значение угла поворота N	Количество повторений $360/N$
$2^{\circ}$	$180^{\circ}$
$3^{\circ}$	$120^{\circ}$
$4^{\circ}$	$90^{\circ}$
$5^{\circ}$	$72^{\circ}$
$6^{\circ}$	$60^{\circ}$
$7^{\circ}$	$51^{\circ}$
$8^{\circ}$	$45^{\circ}$
$9^{\circ}$	$40^{\circ}$
$10^{\circ}$	$36^{\circ}$

Увеличение  
окружности:

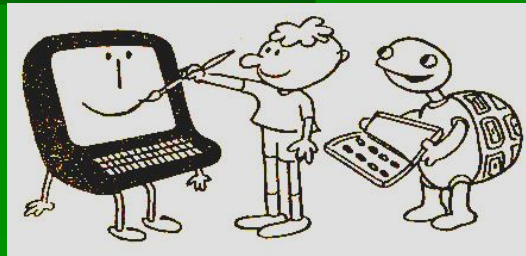
Значение угла поворота N	Количество повторений $360/N$
$0.9^{\circ}$	400
$0.8^{\circ}$	450
$0.7^{\circ}$	514
$0.6^{\circ}$	600
$0.5^{\circ}$	720
$0.4^{\circ}$	900
$0.3^{\circ}$	1200

**Построим окружности разного размера пользуясь данными таблиц:**

- Повтори 180 [вперед 1 направо 2]
- Повтори 400 [ вперед 1 направо 0.9]
- Повтори 36 [ вперед 1 направо 10]
- Повтори 514 [ вперед 1 направо 0.7]
- Повтори 120 [ вперед 1 направо 3]

## Построение дуги.

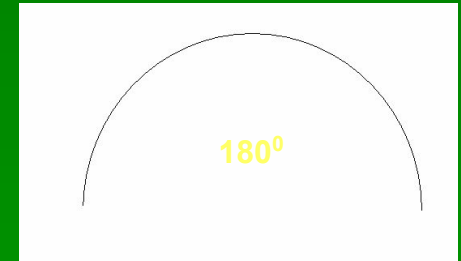
*Дуга – это часть окружности.*



## Например:

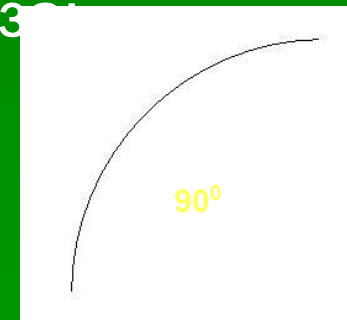
1. Эта дуга – половина окружности, поэтому количество повторений надо уменьшить в два раза:

повтори 180 [вперед 1 направо 1]



2. Эта дуга – четверть окружности, поэтому количество повторений надо уменьшить в четыре раза:

повтори 90 [вперед 1 направо 1]



## Задание.

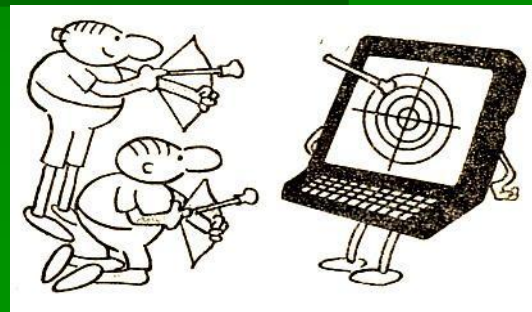
Построить дугу, которая составляет восьмую часть окружности.

**повтори 45 [вперед 1 направо 1]**



# Урок 7.

## Процедуры.





**Процедура** – это последовательность  
известных черепашке  
(базовых) команд,  
имеющая  
**ИМЯ.**

По этому имени процедуру можно  
*вызывать,*  
т. е. выполнять все входящие в неё  
команды.

# Запомни!

Процедура состоит из трёх частей:

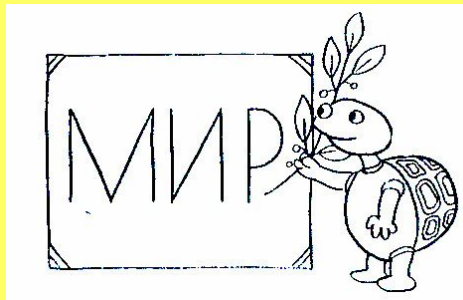
1. Заголовок процедуры.
2. Тело процедуры (описание).
3. Окончание процедуры.

# Запомни!

**Заголовок** процедуры пишется на отдельной строке и состоит из обязательного слова **ЭТО**, за которым указывается имя процедуры.

**Тело процедуры** – все команды, расположенные между заголовком и окончанием.

**Окончание процедуры** – слово **КОНЕЦ**.



## Вызов процедуры.

Имя процедуры, использованное в качестве команды, называется вызовом процедуры.

## Задание 1.

Составим процедуру *квадрат*:

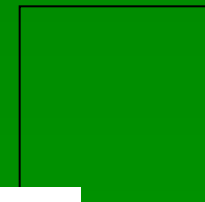
**это квадрат**

**по**

**повтори 4 [вперед 100 направо 90]**

**конец**

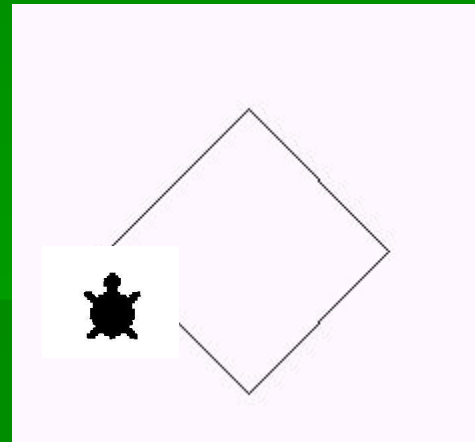
**Вызов процедуры - квадрат**



## Задание 2.

С помощью процедуры *квадрат* изобразить квадрат, повернутый на 45 градусов:

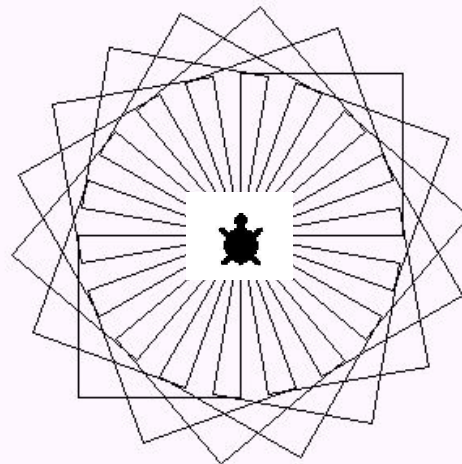
**направо 45  
квадрат**



### Задание 3.

Используя процедуру *квадрат* заставить квадрат покружиться на месте:


**повтори 18 [квадрат направо 20]**



## Задание 4.

Составить процедуру **дуга** и используя эту процедуру нарисовать лепесток .

---

 **это дуга**  
**по**  
**повтори 45 [вперед 1 направо 2]**  
**конец**

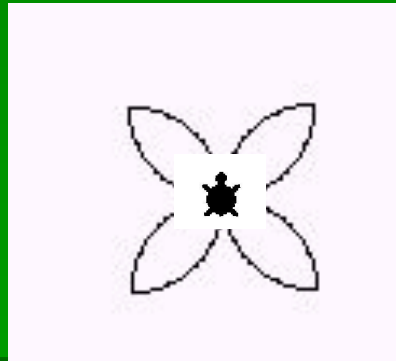
 **дуга направо 90 дуга**

---



## Задание 5.

Имея описание процедуры дуга напишите программу, которая рисует цветочек, состоящий из четырёх лепестков:

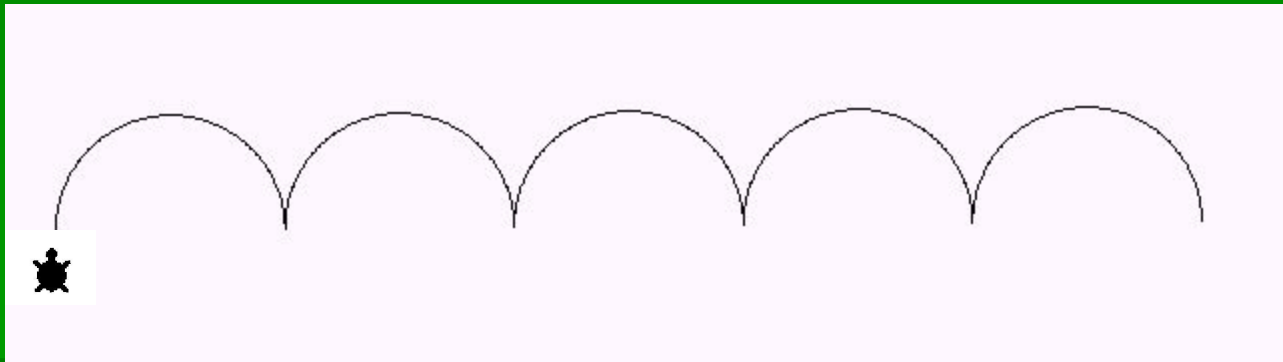


## Программа на ЛОГО:

- ☞ это дуга  
по  
повтори 45[ вперед 1 направо 2]  
конец
- ☞ повтори 4 [дуга направо 90 дуга]

## Задание 6.

Имея описание процедуры дуга, нарисовать море: \_\_\_\_\_



## Программа на ЛОГО:



это дуга

по

повтори 180 [вперед 1 направо 1]

конец

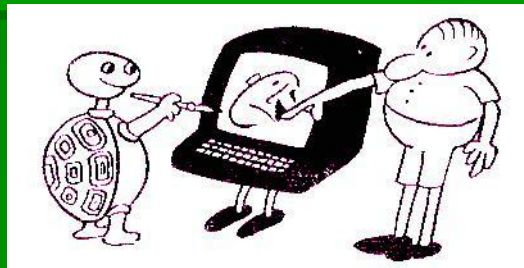


повтори 5 [дуга налево 180]



# Урок 8.

## Процедуры и параметры.



Составим *процедуру*, которая будет рисовать *квадрат*, но вместо конкретного количества шагов, определяющего размер квадрата, поставим *параметр - переменную*, значение которой можно изменять.

## Процедура с параметром:

```
это квадрат :сторона  
повтори 4 [вперед :сторона направо 90]  
конец
```

Параметр **:сторона** указан в заголовке процедуры вслед за её именем.

Этот же параметр встречается (обязательно) и в теле процедуры, в том месте, где предусмотрено использование переменной величины.

# Запомни:

Переменную можно представить себе как ящик, на котором написано имя, а внутри находится значение.

Это значение можно изменять.

Перед именем переменной ставится двоеточие, пробел между двоеточием и именем переменной не допускается.



# Переменные в процедурах называются параметрами.

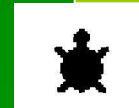
- Значения параметров внутри процедуры не меняются.
- Параметры, указываемые в описании процедуры ( в её заголовке и в теле), называют *формальными параметрами* её описания.
  - *Формальный параметр* всегда представляет собой *имя* некоторой величины.

# Запомни:

В момент вызова процедуры с параметрами надо указать значение этого параметра:

**квадрат 100**

Это значит, что значение переменной для команды вперед следует взять из переменной *сторона*, которая стоит в заголовке процедуры.



Значение величины, указываемое при вызове процедуры называется *фактическим параметром* или параметром вызова процедуры.

*Фактический параметр* всегда является значением, заменяющим собой формальный параметр в описании процедуры;  
с этим значением параметра и выполняется вызванная процедура.

# Процедура с несколькими параметрами.

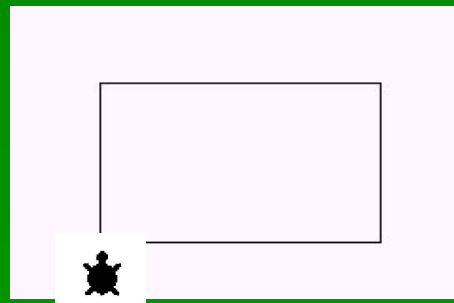
Пример 1.

Составим описание *процедуры*, рисующей прямоугольник, длина и высота которого произвольны:

```
это прямоугольник :высота :длина  
повтори 2 [вперед :высота направо 90  
           вперед :длина направо 90]  
конец
```

Вызов процедуры: **прямоугольник 50 100**

**прямоугольник**



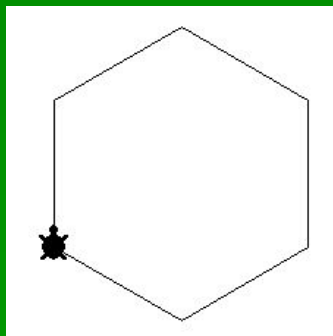
## Пример 2.

Составим *процедуру*, которая будет рисовать любой правильный **многоугольник**, но вместо конкретного количества повторений, количества шагов и угла поворота поставим переменные, значение которых можно изменять:

**это многоугольник** :число :сторона :угол  
**повтори** :число [**вперед** :сторона **направо** :угол]  
**конец**

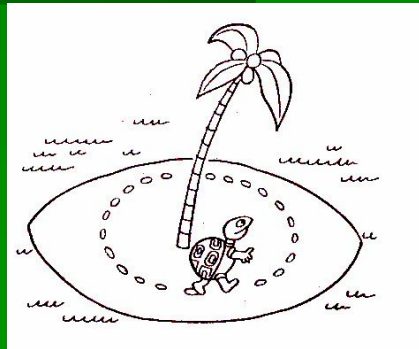
Вызов процедуры: **многоугольник** 6 100 60

**шестиугольник**



# Урок 9.

## Рекурсивные процедуры



**Мы уже знаем, как заставить черепаху начертить  
круговую орбиту:**

это орбита

повтори 360 [ вперед 1 направо 1 ]

конец

**Но как заставить её кружить по этой орбите  
подобно спутнику?**

**Для этого после того, как черепаха начертит  
окружность, она снова**

**должна выполнить процедуру **орбита**:**

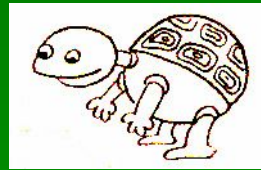
это орбита

повтори 360 [ вперед 1 направо 1 ]

орбита

конец

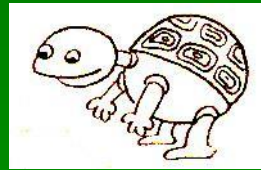
# Запомни:



Процедуры которые вызывают сами себя  
называются рекурсивными.



Рекурсивная процедура  
выполняется бесконечно...



Прервать программу можно  
одновременным нажатием клавиш  
**<Ctrl >** и **<Break>**

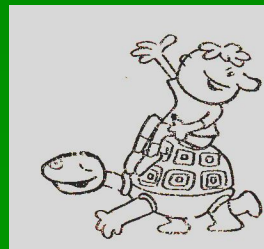
# Бесконечное движение черепашки по треугольной орбите:

**это треугольник**

**повтори 3 [вперед 100 направо 120]**

**треугольник**

**конец**



# Бесконечное движение черепашки по квадратной орбите:

**это квадрат**

**повтори 4 [вперед 100 направо 90]**

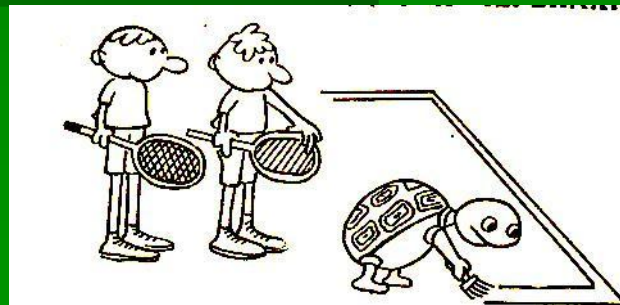
**квадрат**

**конец**



# Урок 10.

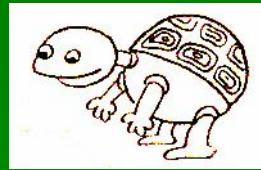
## Координаты.



## Запомни:

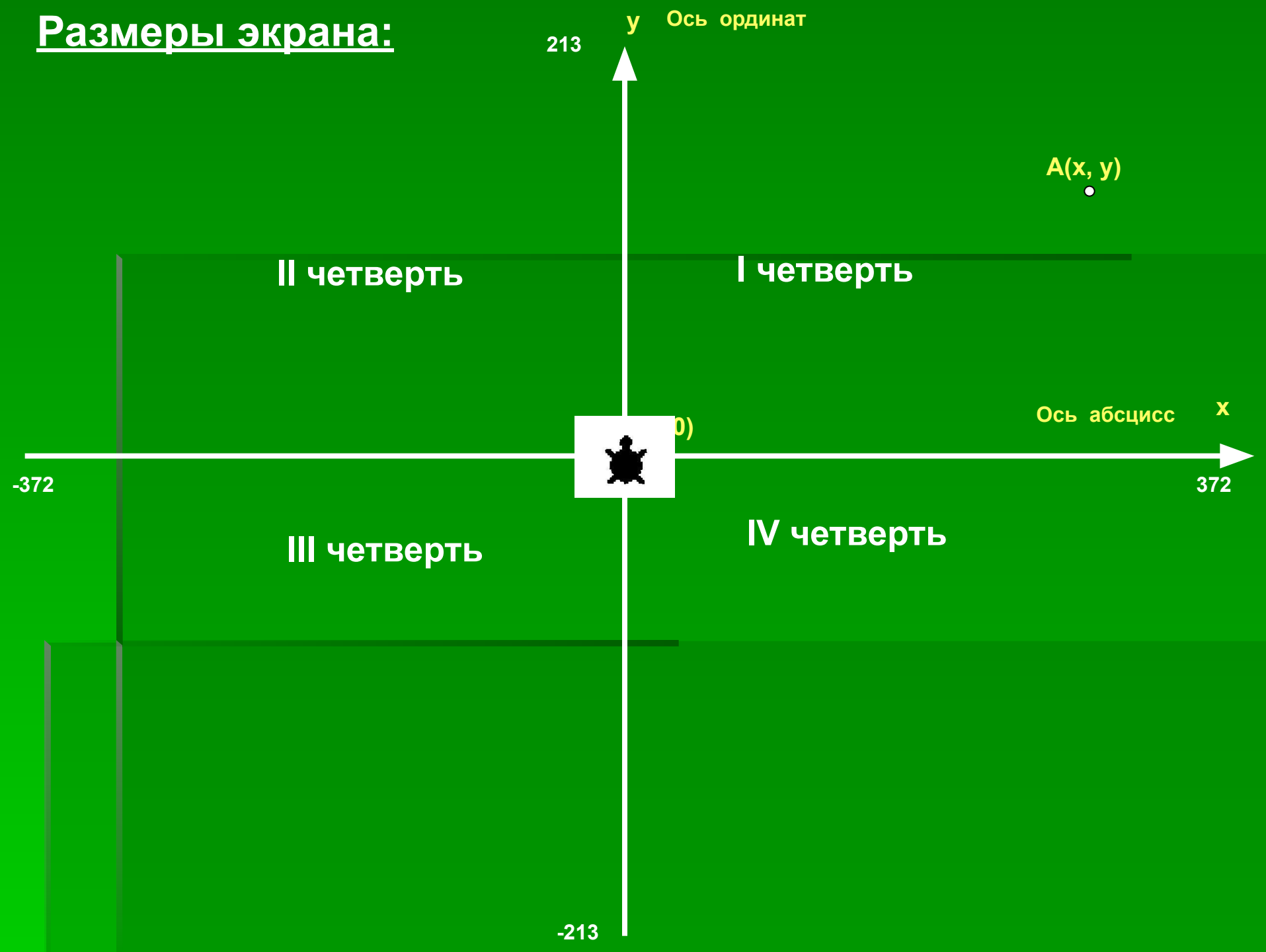
- Два числа, определяющие положение точки на плоскости, называются *координатами*.
- Плоскость с отмеченной на ней точкой начала отсчёта и проходящими через эту точку двумя линиями, называется *системой координат*.
- Точку начала отсчёта называют *началом координат*.
- Направленные линии – *осями координат*.

# Запомни:



Для того, чтобы точка могла быть точно определена парой координат, первым элементом пары всегда является величина  $X$  (абсцисса точки), а вторым элементом – всегда величина  $Y$  (ордината точки).

# Размеры экрана:



Оси координат разбивают плоскость на четыре части, которые называются *координатными четвертями*.

Мы будем работать только в первой координатной четверти

y

I четверть



Координаты центра экрана [0 0]

x



# Внимание!

Решая задачи с использованием координат, мы не будем пользоваться командами вперед, назад, направо, налево.

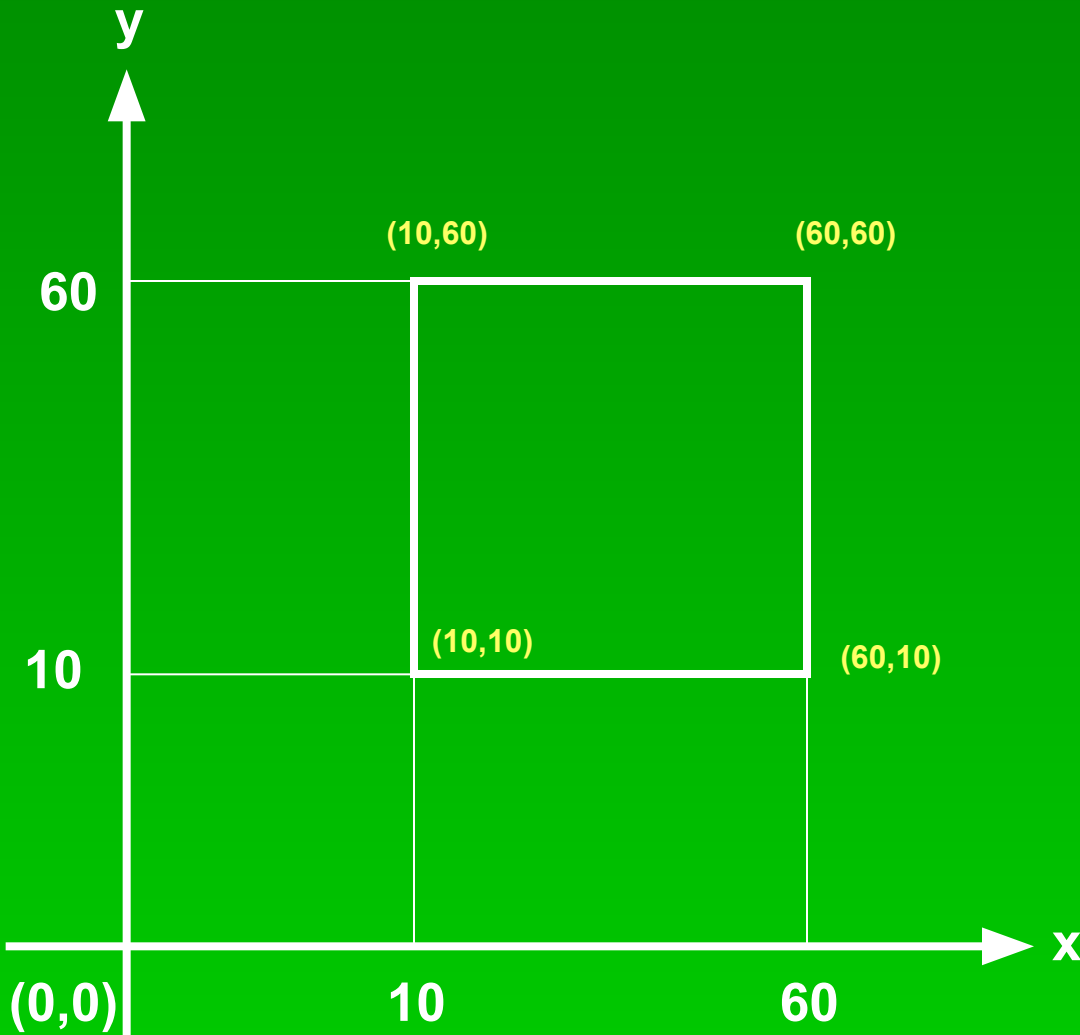
## Новые команды:

- **покажи место** – сообщает координаты точки, в которой находится черепашка;
- **нов\_место [x y]** – перемещает черепашку в точку с координатами [x y];
- **нов\_y число** – изменяет y-координату черепашки;
- **нов\_x число** – изменяет x-координату черепашки;

*Передвигаясь от точки к точке черепашка может оставлять след, а может не оставлять след.*

## Пример 1.

Построить квадрат, пользуясь координатами его вершин.



Программа на ЛОГО:

пп

нов\_место [10 10]

по

нов\_место [10 60]

нов\_место [60 60]

нов\_место [60 10]

нов\_место [10 10]

## Пример 2.

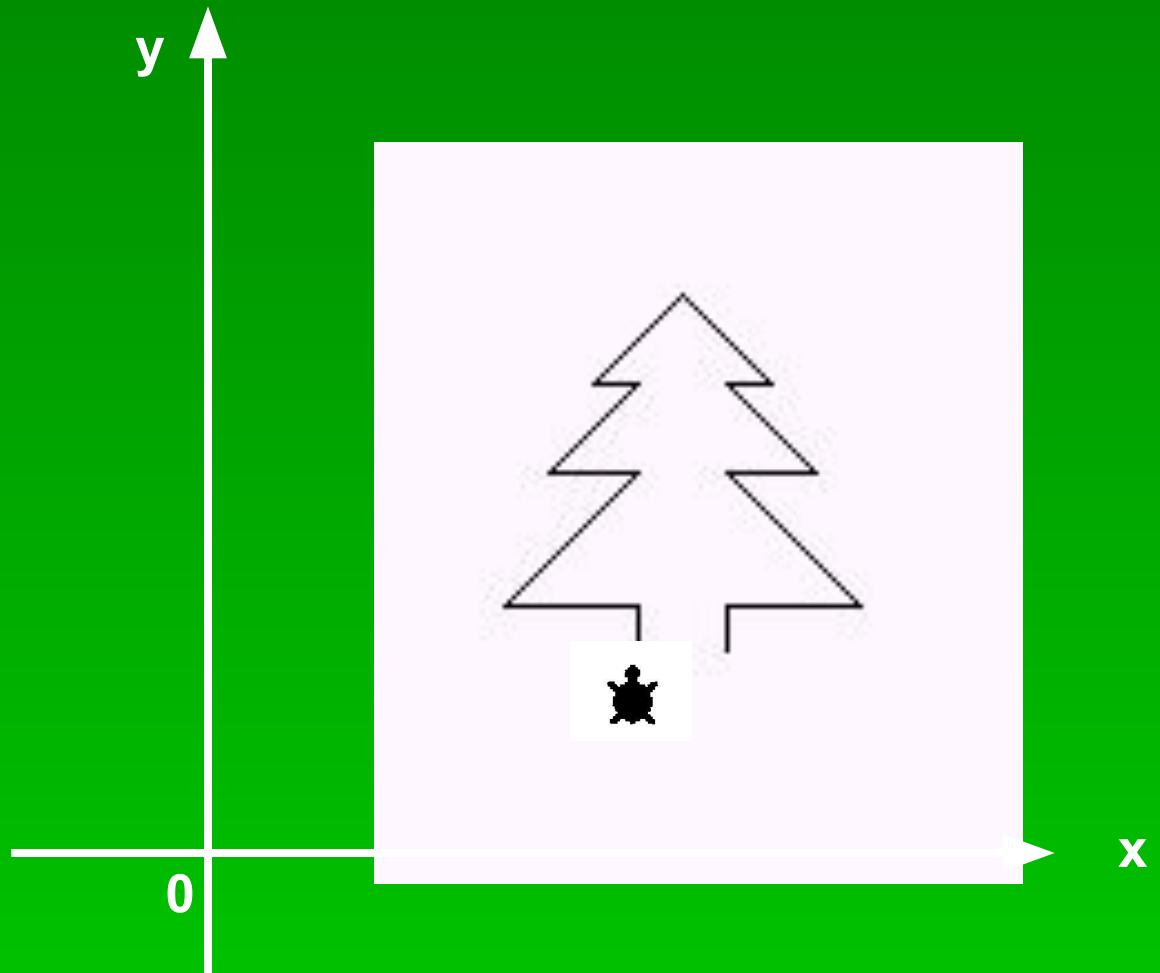
Даны координаты пятнадцати точек.  
Построить фигуру по этим координатам.

1. (40,10)
2. (40,20)
3. (10,20)
4. (40,50)
5. (20,50)
6. (40,70)
7. (30,70)
8. (50,90)
9. (70,70)
0. (60,70)
1. (80,50)
2. (60,50)
3. (90,20)
4. (60,20)
5. (60,10)

## Программа на ЛОГО:

```
пп
нов_место [40 10]
по
нов_место [ 40 20 ]
нов_место [ 10 2 0]
нов_место [ 40 5 0]
нов_место [ 20 50 ]
нов_место [ 40 70 ]
нов_место [ 30 70 ]
нов_место [ 50 9 0]
нов_место [ 70 70 ]
нов_место [ 60 7 0]
нов_место [ 80 5 0]
нов_место [ 60 5 0]
нов_место [ 90 20 ]
нов_место [ 60 20 ]
нов_место [ 60 10 ]
```

Черепашка построит:



**Пример 3.**

Даны координаты двенадцати точек.

Построить фигуру по этим координатам.

1. (30, 60)
2. (60, 30)
3. (150, 30)
4. (180, 60)
5. (100, 60)
6. (100, 160)
7. (130, 160)
8. (120, 150)
9. (130, 140)
0. (100, 140)
1. (160, 60)
2. (30, 60)

пп

нов\_место [30 60]

По

нов\_место [ 60 30]

нов\_место [ 150 30]

нов\_место [ 180 60]

нов\_место [ 100 60]

нов\_место [ 100 160]

нов\_место [ 130 160]

нов\_место [ 120 150]

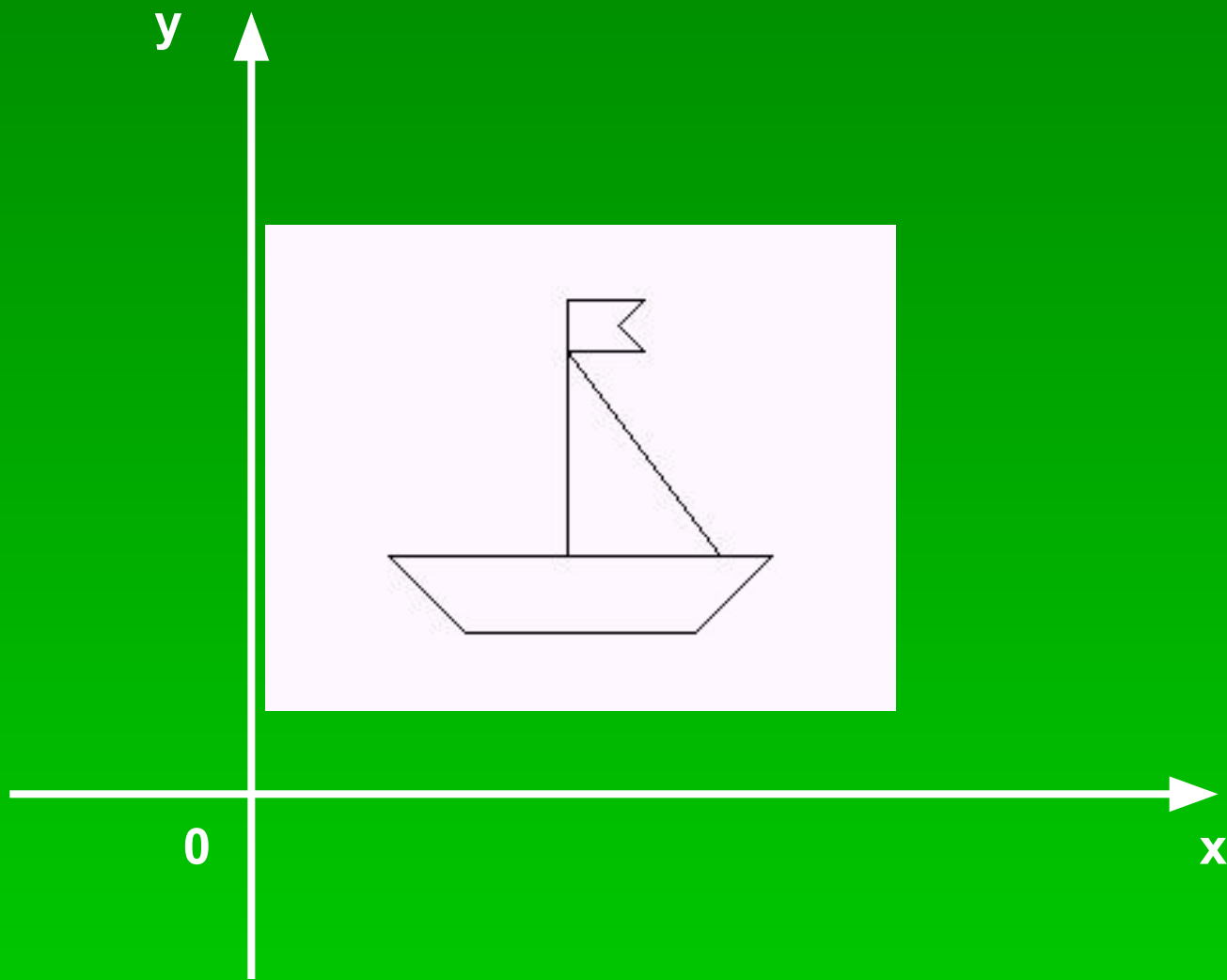
нов\_место [ 130 140]

нов\_место [ 100 140]

нов\_место [ 160 60]

нов\_место [ 30 60]

**Черепашка построит:**



**Слышу – и забываю,  
вижу – и запоминаю,  
делаю – и понимаю.**

(Восточная мудрость)

