

Тема урока: «Понятие рекурсии. Построение рекурсивных алгоритмов в среде исполнителя»

Этап 1. Повторение пройденного материала.

Учитель: Повторить, что такое процедура, как записываются её аргументы.

Процедура — это именованная последовательность команд.

(франц. procedure — от лат. procedo — продвигаюсь)

Использование процедур необходимо тогда, когда мы можем вызвать уже разработанные ранее алгоритмы или нам надо несколько раз выполнить одни и те же действия.

Вопрос ученикам: Как записывается процедура в среде «Kturtle»?

Ответ учеников:

выучи имя_процедуры имя_параметра [тело процедуры]

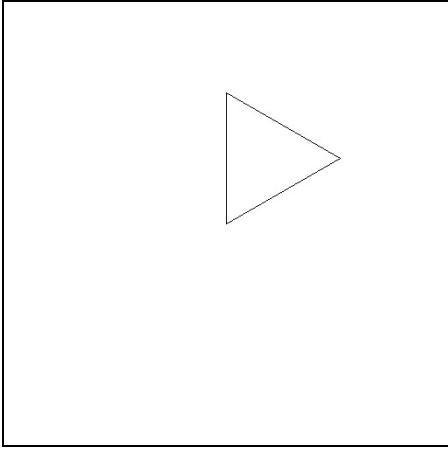
Вопрос ученикам: А как она потом вызывается?

Ответ учеников: имя_процедуры значение_параметра

Значит, в процедуру мы можем передавать параметры. Вспомним, что при описании процедуры параметры называются формальными, а при вызове — фактическими. При вызове процедуры фактические параметры становятся значениями соответствующих формальных параметров, чем и обеспечивается передача данных в процедуру.

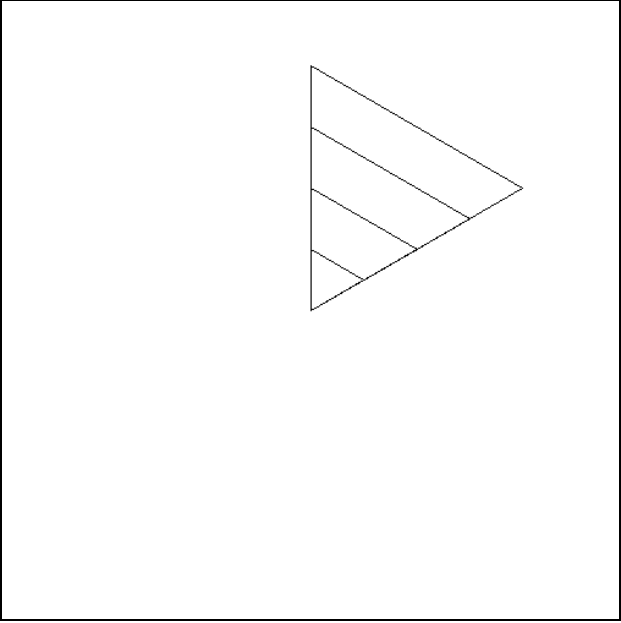
Вопрос ученикам: Написать алгоритм рисования треугольника в среде «Kturtle», используя процедуру с одним параметром — длиной стороны.

Ответ учеников:

<pre>сброс НРХ 500, 500 центр выучи треуг R [повтори 3 [ВП R ПР 120]] треуг 150 спрячь</pre>	
--	--

Вопрос ученикам: Как в алгоритме использовать цикл для вызова нескольких раз процедуры рисования треугольника с разными входными значениями?

Ответ учеников:

сброс HPX 500, 500 центр D=50 выучи треугол R [повтори 3 [ВП R ПР 120]] пока D<=200 [треуг D D=D+50] спрячь	
---	--

Или «повтори 4 [...] вместо пока»

Этап 2. Объяснение нового материала

Объяснение понятия «рекурсия». Исправление алгоритма рисования нескольких треугольников в цикле с разными входными значениями на рекурсивный. Демонстрация различных рекурсий в графическом виде. Показ результатов работы различных рекурсивных алгоритмов в программе «KTurtle». Подробный разбор рекурсивного алгоритма построения треугольника Серпинского.

Учитель:

Рекурсия – это такая организация алгоритма, при которой процедура обращается к самой себе. Сама процедура называется рекурсивной.

В жизни вам не раз приходилось сталкиваться с рекурсией. Вспомните хотя бы стихотворение "У попа была собака" или то, как, сидя в поезде, вы ловили свое отражение в зеркале, которое отражалось в зеркале напротив, которое отражалось в зеркале напротив...

Любую программу с итерациями можно представить в виде рекурсии. Исправим программу рисования нескольких треугольников:

Сначала сотрём цикл и вызовем процедуру рисования треугольника с конкретным параметром:

```

выучи треугол R [  

  повтори 3 [  

    ВП R  

    ПР 120  

  ]  

]
  

треуг 50

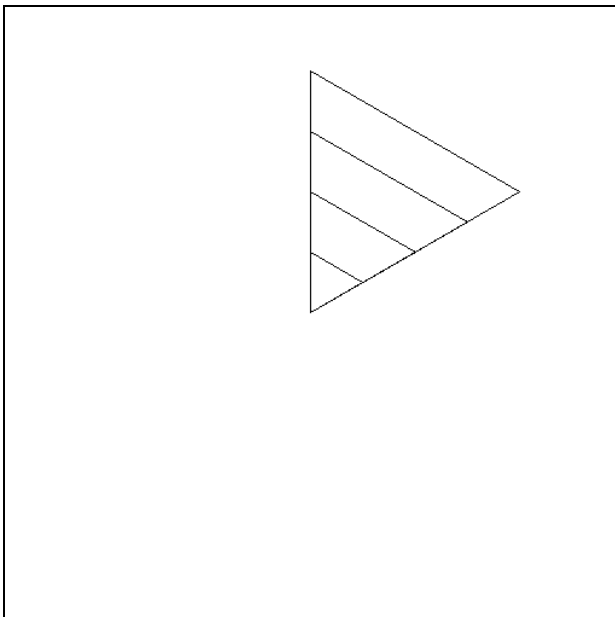
```

Затем попробуем нарисовать несколько треугольников, всё с БОльшей длиной, для этого внутри процедуры построения треугольника вызовём её же, но с параметром, больше на 50 точек:

```
выучи треуг R [  
  треуг R+50  
  повтори 3 [  
    ВП R  
    ПР 120  
  ]  
]  
  
треуг 50
```

Если мы запустим в таком виде программу, то она зависнет. Будет выполняться бесконечный цикл. Ведь там, где мы пишем цикл, должно быть обязательно условие выхода из него!

Допустим, мы будем рисовать треугольники, пока длина не достигнет значения 200. Тогда перед рекурсивным вызовом процедуры мы напишем если $R \leq 200$ [...]

<pre>сброс НРХ 500, 500 центр выучи треуг R [если $R \leq 200$ [треуг R+50 повтори 3 [ВП R ПР 120]]] треуг 50 спрячь</pre>	
--	---

Показ презентации с картинками – результатами рекурсий.

Показ результатов работы различных рекурсивных алгоритмов в программе «Kturtle».

Подробный разбор построения треугольника Серпинского

```

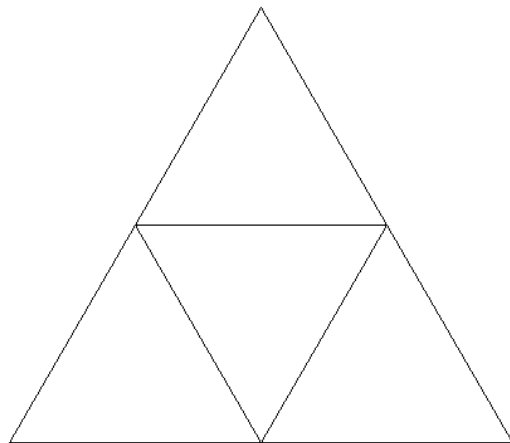
сброс
НРХ 500, 450
ИДИ 50, 400
пр 30

выучи треугол дл
[
если дл>100 [
    повтори 3 [
        треугол дл/2
        ВП дл
        ПР 120
    ]
]
]

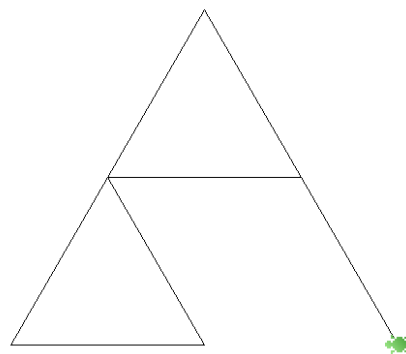
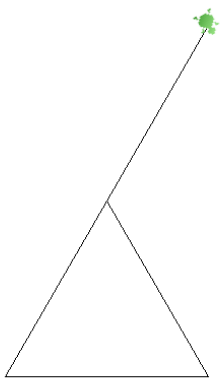
треугол 400

спрячь

```



Рисование по итерациям:



Изменим условие выхода из цикла в два раза:

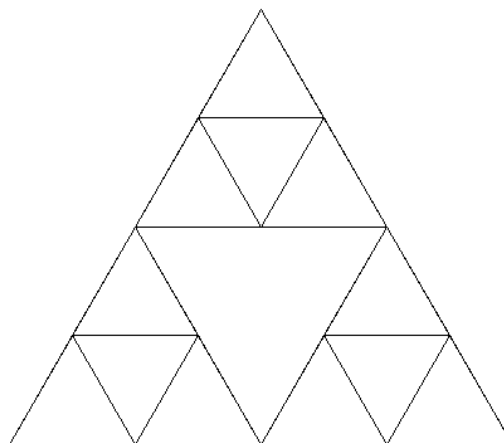
```

выучи треугол дл
[
если дл>50 [
  повтори 3 [
    треугол дл/2
    ВП дл
    ПР 120
  ]
]
]

треугол 400

спрячь

```



Изменим ещё в два раза:

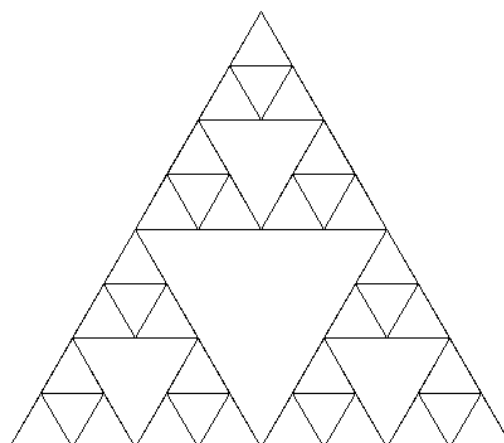
```

выучи треугол дл
[
если дл>25 [
  повтори 3 [
    треугол дл/2
    ВП дл
    ПР 120
  ]
]
]

треугол 400

спрячь

```



И так далее, попробуем задать условие – 10, 5, и наконец – 2.

Как можно модифицировать программу построения треугольника Серпинского, чтобы получился квадрат Серпинского?

Вопрос ученикам: как вы думаете, можно ли изменить алгоритм построения треугольника Серпинского, на построение его квадрата? Или надо будет заново составлять алгоритм?

Ответ учеников: да, надо только 4 раза повторять рисование линии, сменить угол с 120 градусов до 90 и изначально не поворачивать черепашку на 30 градусов.

Давайте попробуем и посмотрим, что получается. (размер холста увеличивается)

```

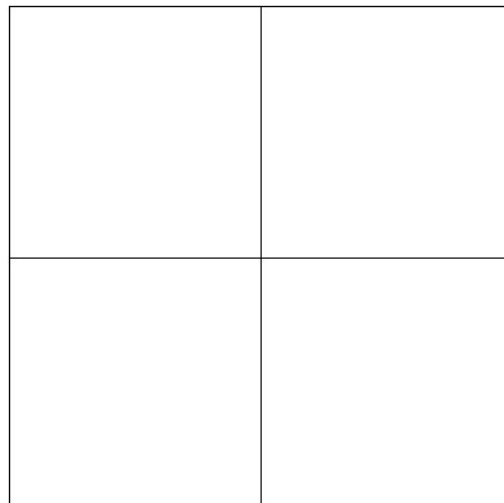
сброс
НРХ 500, 500
ИДИ 50, 450

выучи треуг дл
[
если дл>100 [
    повтори 4 [
        треуг дл/2
        ВП дл
        ПР 90
    ]
]
]

треуг 400

спрячь

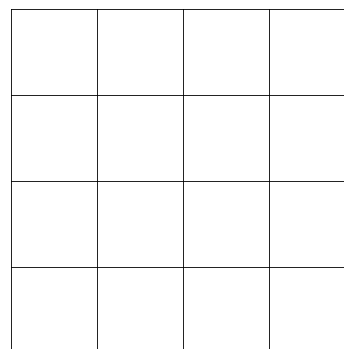
```



```

выучи треуг дл
[
если дл>50 [
    повтори 4 [
        треуг дл/2
        ВП дл
        ПР 90
    ]
]
]

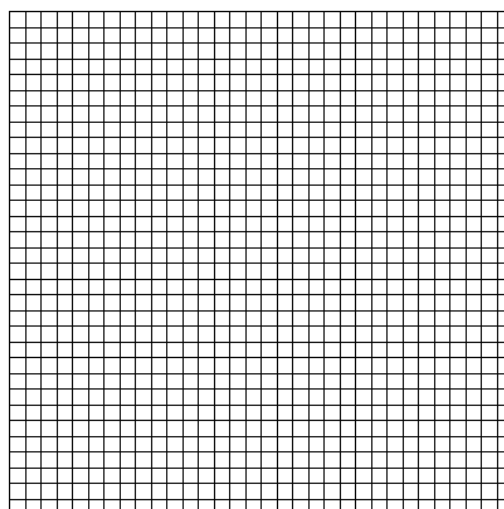
```



```

выучи треуг дл
[
если дл>10 [
    повтори 4 [
        треуг дл/2
        ВП дл
        ПР 90
    ]
]
]

```

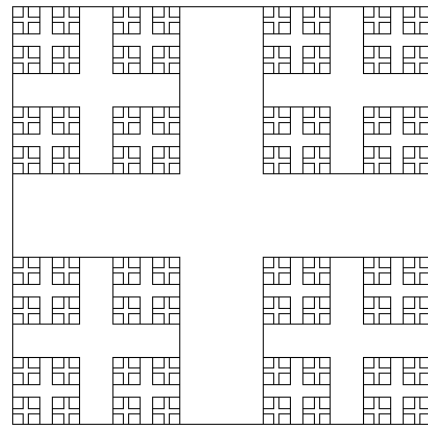


Получается некрасиво, как же исправить?

Давайте вместо уменьшения стороны квадрата в два раза, уменьшим на БОльшее значение. Деление на 2 — это умножение на 0.5, значим мы умножим, например, на 0.4.
Посмотрим на результат:

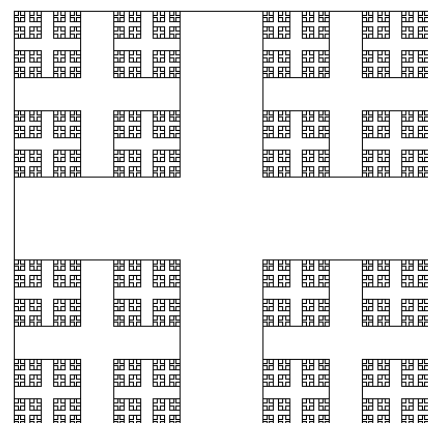
выучи треугол дл

```
[
если дл>10 [
  повтори 4 [
    треугол дл*0.4
    ВП дл
    ПР 90
  ]
]
```



выучи треугол дл

```
[
если дл>2 [
  повтори 4 [
    треугол дл*0.4
    ВП дл
    ПР 90
  ]
]
```



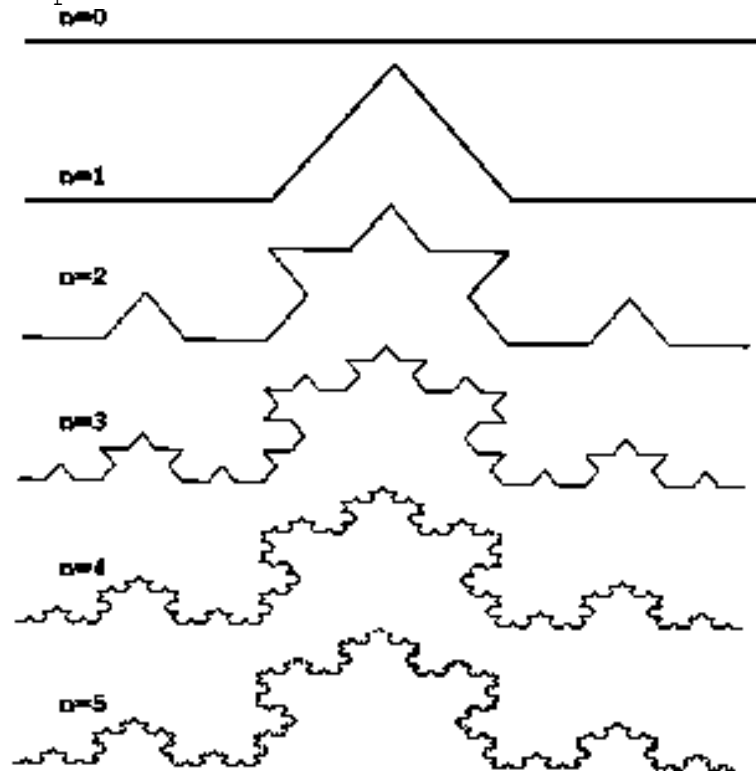
Этап 3. Закрепление нового материала

Задание ученикам: Набрать рекурсивные алгоритмы для построения треугольника и квадрата Серпинского с различными параметрами в среде «Kturtle»

Этап 4. Домашнее задание.

Задание ученикам: Составить алгоритм построения кривой Коха.

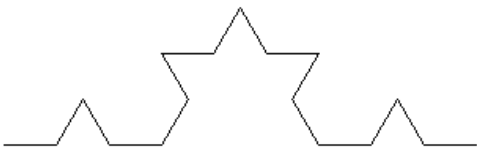
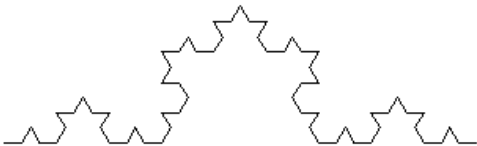
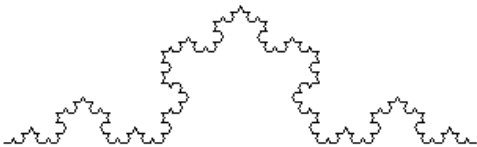

Как строится кривая Коха?



```
сброс  
НРХ 400, 200  
ИДИ 50, 150  
пр 90
```

```
выучи кох дл  
[  
если дл>50 [  
    кох дл/3  
    лв 60  
    кох дл/3  
    ПР 120  
    кох дл/3  
    лв 60  
    кох дл/3  
]  
иначе вп 3*дл  
]
```



кох 100 спрячь	
выучи кох дл [если дл>25 [кох дл/3 лв 60 кох дл/3 ПР 120 кох дл/3 лв 60 кох дл/3] иначе вп 3*дл]	
выучи кох дл [если дл>10 [кох дл/3 лв 60 кох дл/3 ПР 120 кох дл/3 лв 60 кох дл/3] иначе вп 3*дл] выучи кох дл [если дл>3 [кох дл/3 лв 60 кох дл/3 ПР 120 кох дл/3 лв 60 кох дл/3] иначе вп 3*дл]	 
выучи кох дл [если дл>1 [кох дл/3 лв 60 кох дл/3 ПР 120 кох дл/3]	

лв 60 кох дл/3] иначе вп 3*дл]	
---	--

Посмотреть с параметром 100 (должна получиться прямая)

Вывод:

Ученики вспомнили понятие процедуры, как изменить её параметры.
 Узнали, что такое рекурсивные алгоритмы, посмотрели примеры
 рекурсивных алгоритмов и научились их строить.