

# НЕРАВНОВЕСНАЯ АГРЕГАЦИЯ, ФРАКТАЛЫ

- Виноградова Варвара НФИбд-01-18
- Жижченко Глеб НФИбд-01-18
- Жижченко (Ветошкина) Валерия НФИбд-03-18
- Греков Максим НФИбд-01-18
- Кондратьева Анастасия НФИбд-01-18
- Иванова Ольга НФИбд-01-18



# ПОДКЛЮЧЕНИЕ БИБЛИОТЕК

```
from tkinter import *
```

```
from numpy import pi, cos, sin
```

# ФУНКЦИЯ TREE И ЕЕ ВХОДНЫЕ ДАННЫЕ

## TREE(C, N, X0, Y0, LENGTH, BASE\_ANGLE, ANGLE)

C – полотно для рисования

N – количество итераций

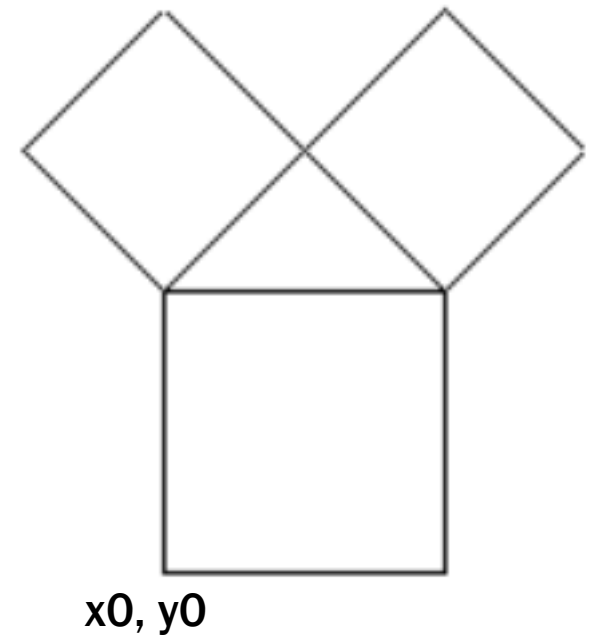
X0 – координата нижней левой точки по оси X для первого квадрата

Y0 – координата нижней левой точки по оси Y для первого квадрата

Length – длина стороны первого квадрата

base\_angle – базовый угол наклона

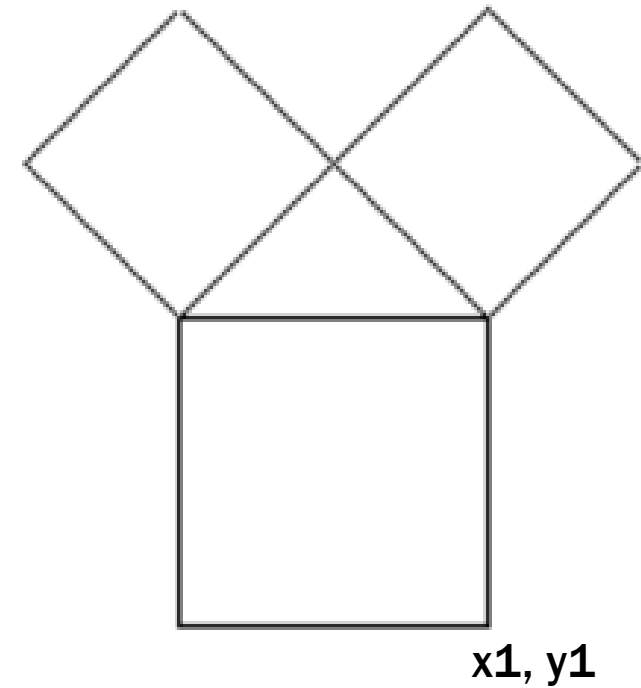
Angle – угол наклона левой ветви



# **(X1, Y1)**

**$x1 = x0 + (\text{int})(\text{length} * \cos(\text{base\_angle}))$**

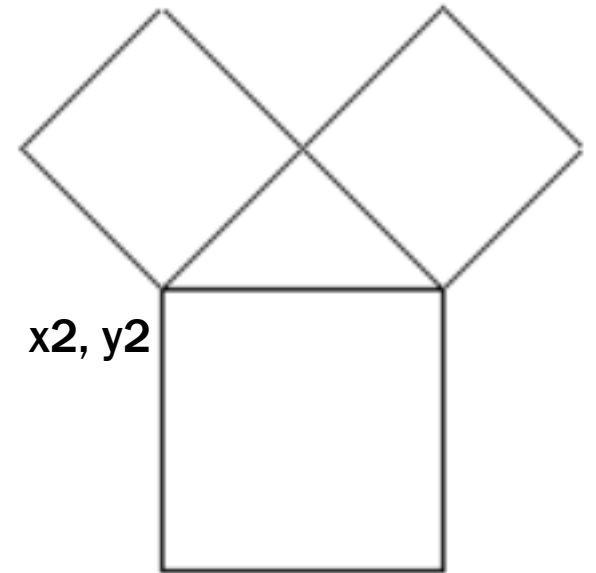
**$y1 = y0 - (\text{int})(\text{length} * \sin(\text{base\_angle}))$**



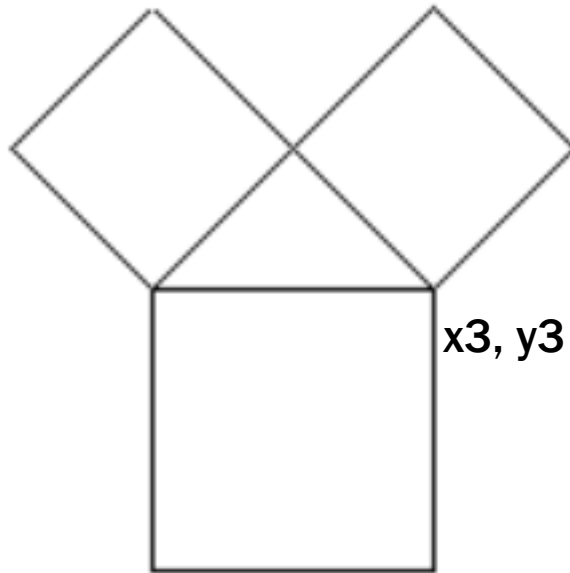
# **(X2, Y2)**

```
x2 = x0 + (int)(length * cos(base_angle  
+ pi / 2))
```

```
y2 = y0 - (int)(length * sin(base_angle +  
pi / 2))
```



# **(X3, Y3)**



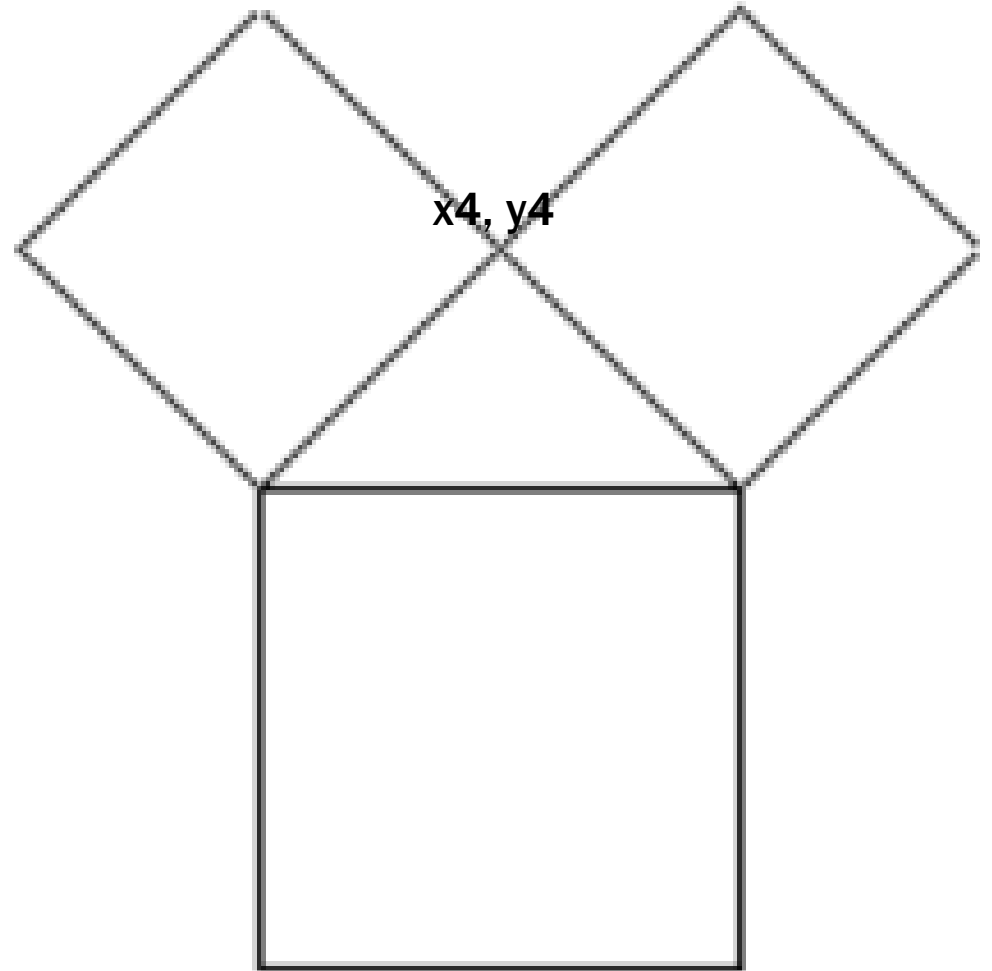
$$x3 = x1 + (\text{int})(\text{length} * \cos(\text{base\_angle} + \pi / 2))$$

$$y3 = y1 - (\text{int})(\text{length} * \sin(\text{base\_angle} + \pi / 2))$$

# $(X4, Y4)$

$x4 = x2 + (\text{int})(\text{length} * \cos(\text{angle}) * \cos(\text{base\_angle} + \text{angle}))$

$y4 = y2 - (\text{int})(\text{length} * \cos(\text{angle}) * \sin(\text{base\_angle} + \text{angle}))$



# СОЕДИНЯЕМ ТОЧКИ ЛИНИЯМИ

```
c.create_line(x0, y0, x1, y1)
```

```
c.create_line(x0, y0, x2, y2)
```

```
c.create_line(x2, y2, x3, y3)
```

```
c.create_line(x1, y1, x3, y3)
```



**IF  $N > 0$ :**

# ОТРИСОВКА ЛЕВОЙ ВЕТВИ

```
left_base_angle = base_angle + angle
```

```
left_length = length * cos(angle)
```

```
tree(c, n - 1, x2, y2, left_length, left_base_angle, angle)
```

# ОТРИСОВКА ПРАВОЙ ВЕТВИ

```
right_base_angle = base_angle - pi / 2 + angle
```

```
right_length = length * sin(angle)
```

```
tree(c, n - 1, x4, y4, right_length, right_base_angle, angle)
```

# ИНИЦИАЛИЗАЦИЯ ПОЛОТНА ДЛЯ РИСОВАНИЯ

```
root = Tk()
```

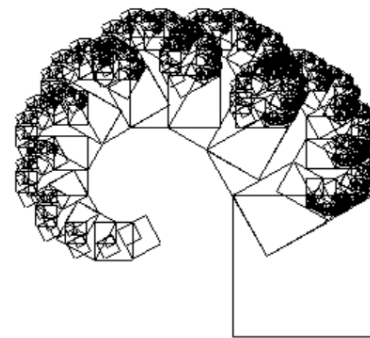
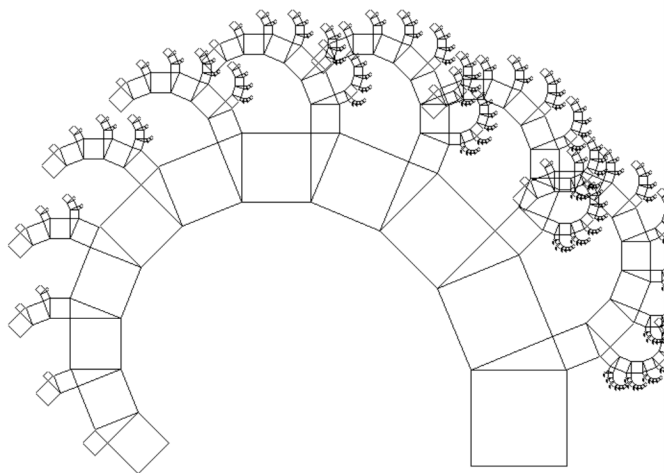
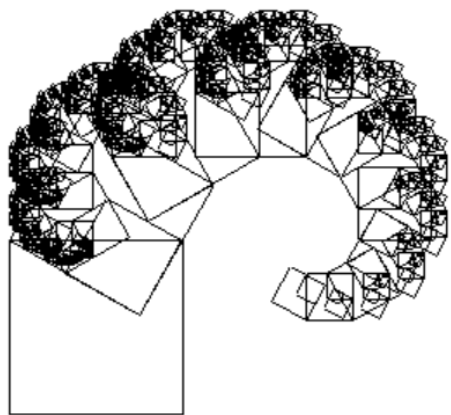
```
c = Canvas(root, width=700,  
height=700, bg='white')
```

```
c.pack()
```

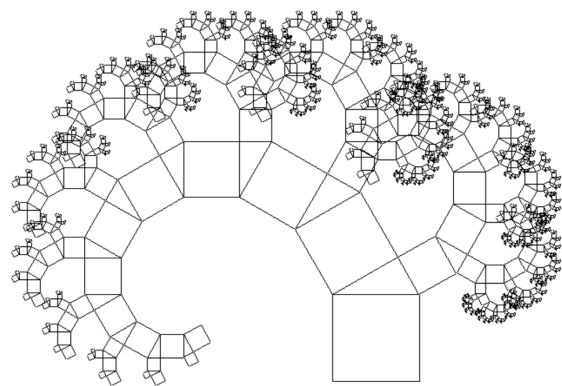
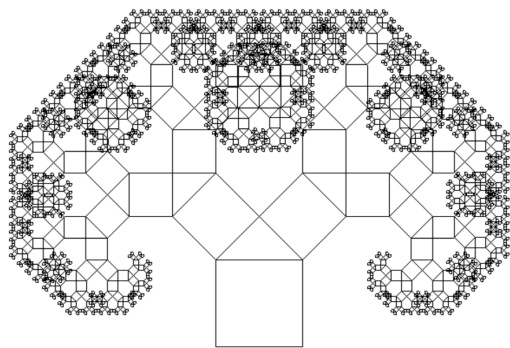
# ВЫЗОВ ФУНКЦИИ ДЛЯ ОТРИСОВКИ ДЕРЕВА ПИФАГОРА

```
tree(c, 10, 300, 500, 100, 0, pi / 4)
```

```
root.mainloop()
```



**УНИВЕРСАЛЬНОСТЬ ПРОГРАММЫ**



**УНИВЕРСАЛЬНОСТЬ  
ПРОГРАММЫ**

```
In [1]: from tkinter import *  
from numpy import pi, cos, sin
```

```
In [34]: def tree(c, n, x0, y0, length, base_angle, angle):  
    x1 = x0 + (int)(length * cos(base_angle))  
    y1 = y0 - (int)(length * sin(base_angle))  
    x2 = x0 + (int)(length * cos(base_angle + pi / 2))  
    y2 = y0 - (int)(length * sin(base_angle + pi / 2))  
    x3 = x1 + (int)(length * cos(base_angle + pi / 2))  
    y3 = y1 - (int)(length * sin(base_angle + pi / 2))  
    x4 = x2 + (int)(length * cos(angle) * cos(base_angle + angle))  
    y4 = y2 - (int)(length * cos(angle) * sin(base_angle + angle))  
  
    c.create_line(x0, y0, x1, y1)  
    c.create_line(x0, y0, x2, y2)  
    c.create_line(x2, y2, x3, y3)  
    c.create_line(x1, y1, x3, y3)  
  
    if n > 0:  
        left_base_angle = base_angle + angle  
        left_length = length * cos(angle)  
  
        tree(c, n - 1, x2, y2, left_length, left_base_angle, angle)  
  
        right_base_angle = base_angle - pi / 2 + angle  
        right_length = length * sin(angle)  
  
        tree(c, n - 1, x4, y4, right_length, right_base_angle, angle)
```

```
In [35]: root = Tk()  
c = Canvas(root, width=700, height=700, bg='white')  
c.pack()
```

```
In [36]: tree(c, 12, 300, 500, 100, 0, pi / 4)  
root.mainloop()  
#Kernel->Restart & RunAll
```



# ВЫВОДЫ



Создали универсальную  
программу для  
отрисовки дерева  
Пифагора



Закончили смысловую  
часть проекта