

7 класс. Практическая работа №3

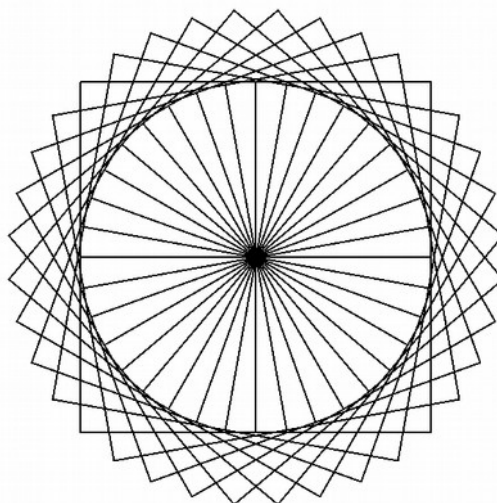
Тема: Вложенные циклы



Ход работы

1. В следующей задаче мы будем вращать квадрат вокруг вершины, находящейся в центре холста. Для этого в цикле мы будем рисовать квадрат и производить поворот Черепахи на некоторый угол (угол поворота * число повторов = 360°).

```
# Программа 3.1
from turtle import*
reset()
speed(10)
for j in range(36):
    fd(200)
    rt(90)
    fd(200)
    rt(90)
    fd(200)
    rt(90)
    fd(200)
    rt(90)
    lt(10)
mainloop()
```



2. В этой программе вновь появились повторения команд. Можно ли эту программу как-то сократить? Да, можно! Для этого нужно применить цикл внутри другого цикла! Перепишем программу 3.1 следующим образом:

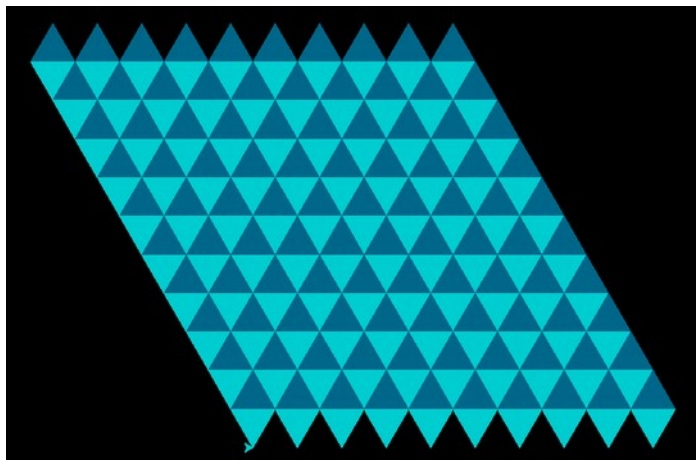
```
# Программа 3.2
from turtle import*
reset()
speed(10)
for j in range(36):
    for k in range(4):
        fd(200)
        rt(90)
    lt(10)
mainloop()
```

Примечание. Обратите внимание на следующие факты: 1. инструкции вложенного цикла имеют дополнительный отступ относительно внешнего цикла, 2. переменная вложенного цикла должна иметь другое имя, отличное от имени переменной внешнего цикла, 3. в данной программе можно обойтись и одним циклом.

3. Как работает вложенный цикл? Все очень просто! На один шаг внешнего цикла, вложенный делает все определенные в нем шаги.

Хорошим примером использования вложенных циклов являются рисунки мозаики (паркета). В программе ниже используются два основных цикла: внешний с переменной *k* для определения рядов плиток, а вложенный для укладки текущего ряда, с переменной *j*.

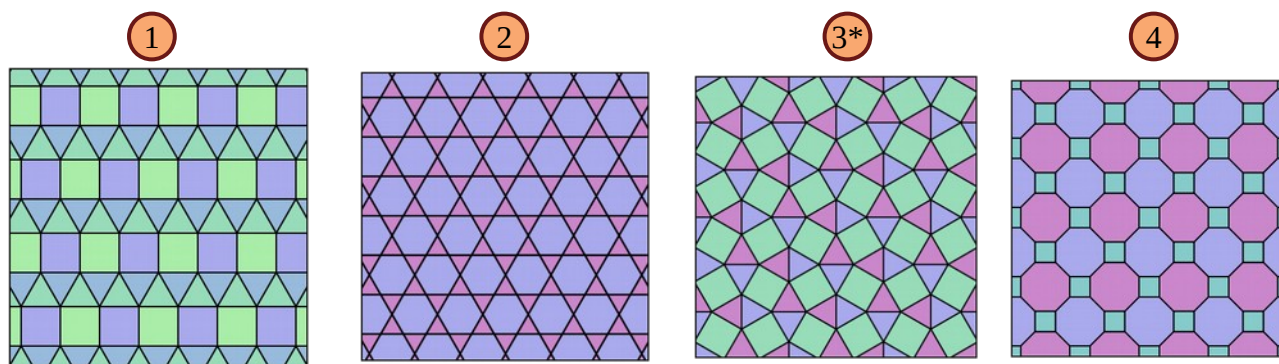
```
# Программа 3.3
from turtle import*
reset()
speed(10)
bgcolor("black")
goto(-300, 300)
M = int(input("Количество рядов -> "))
N = int(input("Количество плит -> "))
D = int(input("Длина стороны -> "))
for k in range(M):
    for j in range(N):
        color("DeepSkyBlue4")
        begin_fill()
        for z in range(3):
            fd(D)
            lt(120)
        end_fill()
        color("dark turquoise")
        begin_fill()
        for z in range(3):
            fd(D)
            rt(120)
        end_fill()
        fd(D)
    up()
    bk(N * D)
    rt(60)
    fd(D)
    lt(60)
    down()
mainloop()
```



4. В программе 3.3 мы ввели использование переменных. Это довольно удобно, если мы хотим каждый раз запускать программу с разными значениями для количества рядов, плиток в ряду и длины стороны. Чтобы получить с клавиатуры значение переменной используется функция `input()` аргументом которой выступает строка подсказки и функция `int()` для преобразования полученной строки в целое число. Казалось бы такой простой орнамент, а сколько кода! Несложно заметить, что во вложенном цикле дважды используется один и тот же большой фрагмент кода (заливки и рисования треугольника) за той лишь разницей, что повороты осуществляются в разные стороны. Вскоре мы научимся не повторяться в программе дважды и большее количество раз.

Примечание. Если вы работаете в IDLE, то значение нужно вводить в Python Shell

5. Используя приобретенные ранее знания замостите холст следующими плиточными рисунками:



Автор: R. A. Nonenmacher - Created by me, CC BY-SA 4.0, <https://commons.wikimedia.org/w/index.php?curid=4461523>

Домашняя работа

