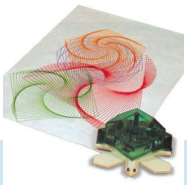
8 класс. Практическая работа №3 (Turtle)

Тема: Ввод данных. Подпрограмма (своя функция)



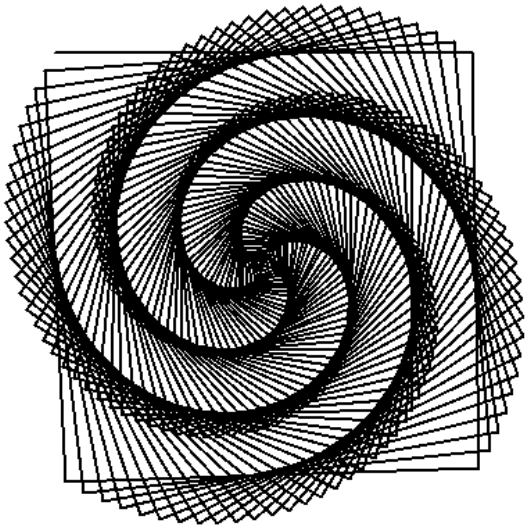
Ход работы

1. При реализации большой программы всегда полезно разбивать задачу на несколько подзадач и выносить эти подзадачи в отдельные алгоритмы. Такие части программы (подпрограммы), которые реализованы вне основной программы и которые вызываются, по мере необходимости, в основной программе называются **функциями**. Мы уже знаем множество функций модуля turtle. Эти функции созданы разработчиками модуля, мы же будем создавать свои функции. Синтаксис функций довольно простой:

```
def имя_функции(список параметров):
```

2. Все инструкции, находящиеся в теле функции, рассматриваются как блок, поэтому имеют отступ. Разработаем программу создающую спиральный узор «скручиванием» вовнутрь фигуры при рисовании многоугольника. Этот эффект достигается путем изменения угла поворота на угол чуть меньший (на 1 градус), чем в правильном многоугольнике и уменьшением длины стороны на каждом шаге цикла на единицу.

```
#Программа 3.1
from turtle import*
reset()
speed(10)
pensize(2)
def sp(k):
    for j in range(k):
        fd(k-j)
        rt(89)
sp(300)
```



3. Польза от этой функции невелика, так как не реализована работа с пользователем программы, которая бы позволила вводить в программу произвольные параметры функции, например, число сторон скручиваемого многоугольника, начальную длину стороны и т. п. Реализовать диалог с пользователем можно путем применением встроенной функции input(), например:

```
a = int(input("Введите длину стороны a = "))
```

Поскольку с клавиатуры вводятся текстовые данные, их необходимо преобразовать в целое число. Это можно сделать с помощью другой функции

— int(). В итоге программу можно преобразовать к следующему виду:

```
#Программа 3.2
from turtle import*
reset()
speed(10)
pensize(2)
a = int(input("Введите длину стороны a = "))
n = int(input("Введите число сторон n = "))
def sp(A, N):
    for j in range(A):
        fd(A - j)
        rt(360/N - 1)
sp(a, n)
```

Примечание. Обратите внимание, что в реализации функции используются другие имена переменных. Это нужно для того, чтобы различать переменные используемые в программе (глобальные по отношению к функции) и локальные переменные самой функции.

4. Создадим программу, которая будет выводить на холсте квадраты и запрашивать у пользователя их количество и цвет заливки.

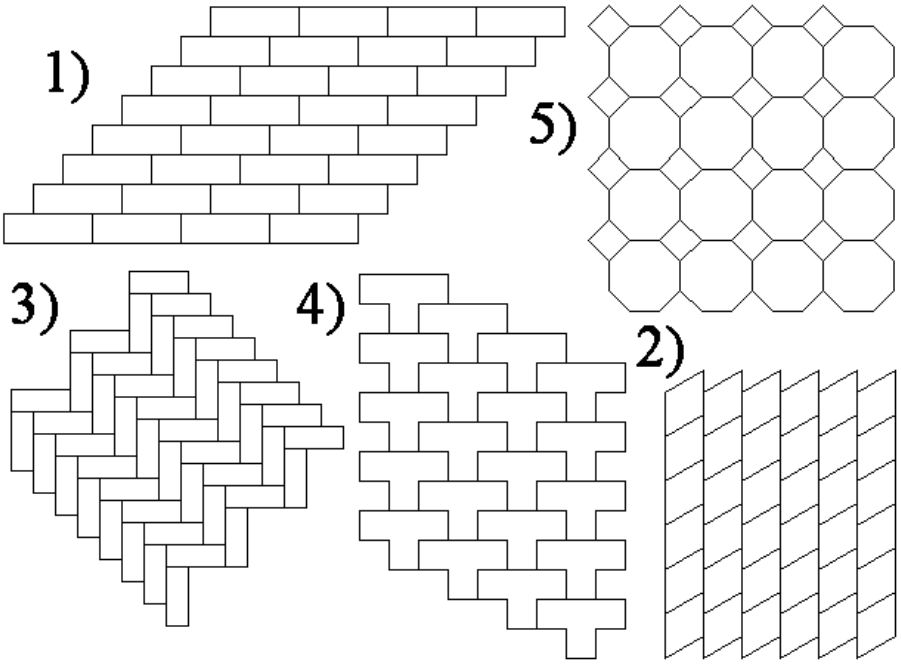
```
#Программа 3.3
from turtle import*
reset()
speed(10)
def Квадрат(c):
    color(c, c)
    begin_fill()
    for j in range(4):
        fd(50)
        rt(90)
    end_fill()
    up()
    fd(100)
    down()
n = int(input("Сколько нарисовать квадратов? n = "))
for j in range(n):
    Цвет = input("Введите название цвета => ")
    Квадрат(Цвет)
```

Примечание. Обратите внимание на то, что python хорошо справляется с кириллицей в именах переменных. Однако, лучше подбирать имена на латинице (осмысленные английские слова).

5. Создайте программу, которая бы имела пользовательскую функцию (можно несколько) и запрашивала ввод каких-то данных. Идеи можно почерпнуть на стр. 3.

Домашняя работа: Создайте программу рисования шахматной доски.
Примечание. *Шахматная доска* состоит из 64 одинаковых квадратов в восемь рядов и столбцов. Белые и чёрные квадраты чередуются.

1 вариант



2 вариант