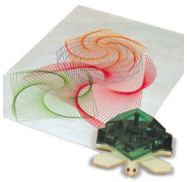
8 класс. Практическая работа №2 (Turtle)

Тема: Выполнение программы в режиме запуска модуля.
Использование цикла for и списка



Ход работы

1. Запустите Python Shell. В меню выполните команды: File → New File (Ctrl+N). Откроется окно редактора скриптов на языке python. В этом окне мы будем набирать и сохранять исходный текст программы. Это нельзя было сделать в интерактивном режиме. Для запуска программы необходимо выполнить команду в меню: Run → Run Module (функциональная клавиша F5). Файл не будет выполнен, пока не будет сохранен. Сохраните файл в папке проектов с осмысленным именем, содержащим латинские символы.
2. Выполняя практическую работу №1 вы заметили, что в программе приходилось многократно писать одни и те же инструкции в результате выполнения которых исполнитель повторял одни и те же действия (например, при рисовании квадрата необходимо было четырежды выполнить команды fd() и rt()). Но что, если потребуется выполнять повторяющиеся действия, скажем, 100 раз? В «ручную» это сделать не представляется возможным. На помощь приходит инструкция цикла for. Её синтаксис таков:

```
for j in range(n):
```

Здесь range() - функция, которая создает арифметическую прогрессию (в данном случае с шагом 1). Результатом работы функции будет числовая последовательность от 0 до n-1. Числа этой последовательности будут передаваться в переменную цикла j, пока последовательность не будет завершена. Таким образом, если range() имеет аргумент 4, то цикл выполнит четыре шага, при этом переменная j будет изменять свои значения на каждом шаге цикла от 0 до 3. В результате, код программы (для рисования квадрата) будет выглядеть следующим образом:

```
for i in range(4):
    fd(150)
    rt(90)
```

Обратите внимание на отступ для группы инструкций, которые повторяются в цикле! Это не случайно. Для того, чтобы некоторые инструкции выполнялись на каждом шаге цикла как единое целое, их нужно объединить в блок (составную инструкцию). Синтаксически блок оформляется в виде отступов одинаковой величины (например, однократное нажатие на клавишу Tab). При этом, используются одинаковые непечатаемые символы.
3. Усложним программу. Из центра холста нарисуем несколько окружностей так, чтобы их центры были равноудалены друг от друга, как показано на рис. 1.

```
# Программа 2.1
from turtle import*
reset()
speed(10)
pensize(2)
for j in range(10):
    circle(100)
    rt(36)
```

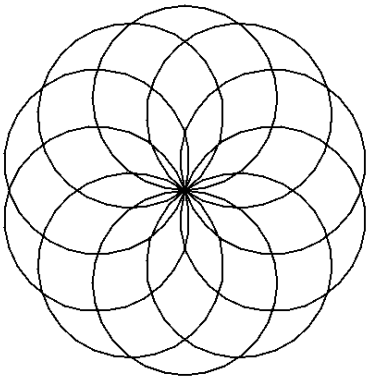


Рис. 1: Задание 2

4. Добавим цвета. В цикле также можно использовать функцию заливки. Но возникает вопрос: «Как изменять цвета в цикле автоматически?». Возможно несколько вариантов. Мы рассмотрим один из них. Создадим список (массив) цветов с именем colors, к элементам которого мы будем обращаться, используя небольшую хитрость:
colors = ["red", "blue", "salmon", "magenta", "cyan", "orange", "lime", "khaki"]
Примечание. Перечень цветов python можно получить по этой ссылке:
https://matplotlib.org/1.5.1/mpl_examples/color/named_colors.hires.png
Элементы списка нумеруются (индексируются) от 0 до 7 (всего 8 цветов). Обратиться к элементу списка можно используя следующий синтаксис colors[n], где n — индекс элемента. Хитрость заключается в том, что мы используем операцию % - взятия остатка от деления переменной цикла j на количество элементов в списке для получения текущего индекса элемента списка. Таким образом, при изменении значения переменной цикла (j), мы поочередно получаем значения индексов элементов от 0 до 7

```
# Программа 2.2
from turtle import*
reset()
speed(10)
colors = ["red", "blue", "salmon", "magenta", "cyan",
"orange", "lime", "khaki"]
for j in range(10):
    fillcolor(colors[j % 8])
    pencolor(colors[j % 8])
    begin_fill()
    circle(100)
    end_fill()
    rt(36)
```

Рис. 2: Задание 3

Примечание. Обратите внимание на новые функции fillcolor() - цвет заливки и pencolor() - цвет пера (как альтернатива color()).

5. Разработайте программу, которая выводит вращение вокруг центра холста некоторого правильного многоугольника.

Домашняя работа