Laboratorium Podstaw Programowania Komputerów

Laboratorium Python 8

Moduł turtle - rysowanie fraktali

Moduł **turtle**, będący częścią biblioteki standardowej języka Python, wykorzystywany jest najczęściej w celach edukacyjnych. Pozwala on na tworzenie prostych, dwuwymiarowych grafik poprzez sterowanie obiektem rysującym na ekranie, określanym żółwiem lub robotem. Obiekt ten najczęściej porusza się do przodu i do tyłu (rysując za sobą linie) oraz wykonuje obroty (zmienia kierunek, w którym się przemieszcza). Taką formę tworzenia grafiki przyjęło nazywać się grafiką żółwia a sam pomysł modułu **turtle** języka Python wywodzi się z języka Logo, stworzonego w 1966 przez Wally'ego Feurziga i Seymoura Paperta.

Moduł **turtle** do programu w języku Python dołączamy poprzez poniższą instrukcję:

```
import turtle
```

Z kolei obiekt żółwia w programie tworzymy wywołując inicjalizator (inaczej konstruktor) klasy `Turtle`:

```
t = turtle.Turtle()
```

Sterowanie odbywa się poprzez wywoływanie odpowiednich metod na stworzonym w ten sposób obiekcie żółwia. Podstawowe operacje to:

`t.forward(20)` – przesuń żółwia 20 pikseli do przodu,
`t.backward(40)` – przesuń żółwia o 40 pikseli do tyłu,
`t.left(45)` – obróć żółwia o 45 stopni w lewo,
`t.right(90)` – obróć żółwia o 90 stopni w prawo.

W ten sposób narysowanie kwadratu o boku 20 pikseli wymaga następujących instrukcji:

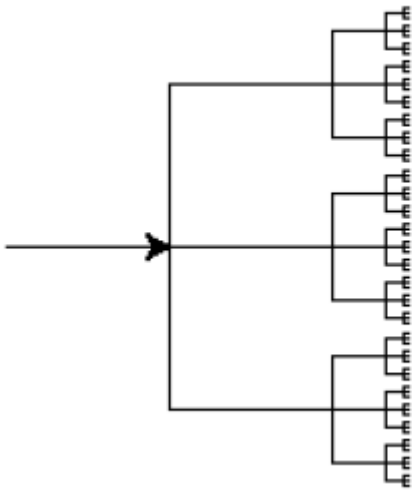
```
t.forward(20)
t.left(90)
t.forward(20)
t.left(90)
t.forward(20)
t.left(90)
t.forward(20)
t.left(90)
```

lub zapisując to samo krócej, w postaci pętli:

```
for i in range(4):
     t.left(90)
     t.forward(20)
```

Jeśli do tych prostych instrukcji rysujących doda się możliwość rekurencyjnego wywoływania funkcji, to można z wykorzystaniem tego modułu tworzyć fraktale (najczęściej obiekty samo-podobne). Jak więc przykładowo narysować prosty fraktal przedstawiony na rys. 1? Można to

zrobić na wiele sposobów. Poniżej przedstawiono ogólny schemat postępowanie w tego typu przypadkach oraz przykładowe rozwiązanie.



Rysunek 1: Prosty fraktal

Kroki postępowania w celu narysowania figury bazującej na rekurencji:

1. Zidentyfikuj element (lub pewien schemat) powtarzający się w ramach całej figury na kolejnych jej poziomach.
2. Znajdź jak najprostszy ciąg instrukcji potrzebny do narysowania takiego elementu i umieść go w funkcji.
3. Dodaj w tak napisanej funkcji jej rekurencyjne wywołania w miejscach, gdzie element się „powtarza". Zwróć uwagę na kierunek i zwrot obiektu rysującego przed i po wywołaniu funkcji. Dodaj warunek ograniczający dalsze rekurencyjne wywołania funkcji.

Przykładowo dla prostego fraktala przedstawionego na rys. 1 może to przebiegać następująco:

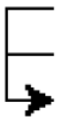
1. Jako powtarzający się element fraktala przyjmujemy fragment przedstawiony na rys. 4 i 5.
2. Aby narysować taki fragment wystarczy wykonać ciąg instrukcji, które przedstawiono poniżej w funkcji `frac_widly()`:

```
import turtle
t = turtle.Turtle()
t.speed("fastest")

def frac_widly(x):
    # odkomentuj w punkcie 4. if x<=1:
    # odkomentuj w punkcie 4.     return
    t.left(90)
    t.forward(x)
    t.right(90)
    t.forward(x)
    # odkomentuj w punkcie 3. frac_widly(x/3)
```

```
    # cofnij się
    t.backward(x)
    t.right(90)
    t.forward(x)
    # rysuj środkową część wideł (rys. 3)
    t.left(90)
    t.forward(x)
    # odkomentuj w punkcie 3. frac_widly(x/3)
    # cofnij się
    t.backward(x)
    # rysuj prawą część wideł (rys. 4)
    t.right(90)
    t.forward(x)
    t.left(90)
    t.forward(x)
    # odkomentuj w punkcie 3. frac_widly(x/3)
    # cofnij się do punktu początkowego i obróć się
    # w kierunku początkowym (rys. 5)
    t.backward(x)
    t.left(90)
    t.forward(x)
    t.right(90)

    # odkomentuj w punkcie 5. x = 64
    # odkomentuj w punkcie 5. t.forward(x)
    # odkomentuj w punkcie 5. frac_widly(x)
```

Rysunek 2:

Rysunek 3:



Rysunek 4:

Rysunek 5:

3. Dodajemy rekurencyjne wywołania `frac_widly(x/3)` w odpowiednich miejscach w funkcji (gdzie element zaczyna się powtarzać z innym wymiarem).
4. Ponadto na początku definicji funkcji `frac_widly()` należy dodać jeszcze warunek, który ograniczy liczbę jej rekurencyjnych wywołań:

```
if x <= 1:
    return
```

5. Chcąc narysować fraktal z rys. 1 wystarczy już tylko przypisanie zmiennej x wartości, narysowanie pierwszego odcinka o długości x i wywołanie funkcji `frac_widly` z tym samym parametrem x:

```
x = 64
t.forward(x)
frac_widly(x)
```

Inne istotne uwagi

Pisząc funkcje rekurencyjne warto zwrócić uwagę na pewne właściwości zaprezentowanych wcześniej metod dla obiektu typu `Turtle`, które mogą być przydatne podczas rysowania niektórych figur:

`backward(x)` można uzyskać także jako `forward(-x)`
`left(x)` można uzyskać także jako `right(-x)`
`left(360 - x)` można uzyskać także jako `right(x)`

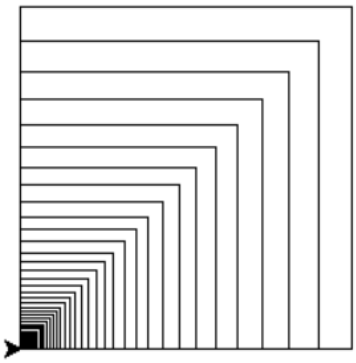
Ćwiczenia do samodzielnej realizacji:

Zadanie 1.

Napisz program zawierający instrukcję rysowania trójkąta równobocznego .

Zadanie 2.

Napisz program pozwalający na narysowanie poniżej przedstawionego fraktala:



Zadanie 3*.

Napisz program pozwalający na narysowanie fraktala przedstawionego na następnej stronie.