



Konstrukcje fraktalne w języku Python

- [Wprowadzenie](#)
- [Film samouczek](#)
- [Przeczytaj](#)
- [Sprawdź się](#)
- [Dla nauczyciela](#)



Konstrukcje fraktalne w języku Python

Źródło: Nick Cooper, domena publiczna.

W e-materiale [Konstrukcje fraktalne](#), dowiedzieliśmy się, czym są paproć Barnsleya oraz smok Heighwaya. Te obiekty samopodobne można generować za pomocą algorytmów IFS, tzn. systemu funkcji iterowanych.

W tym e-materiale zajmiemy się stworzeniem tych fraktali w języku Python.

Ciekawi cię, jak wyglądają implementacje w innych językach programowania? Możesz się z nimi zapoznać w dwóch pozostałych e-materiałach z tej serii:

- [Konstrukcje fraktalne w języku C++](#),
- [Konstrukcje fraktalne w języku Java](#).

Więcej zadań? Sięgnij do [Konstrukcje fraktalne – zadania maturalne](#).

Twoje cele

- Użyjesz modułu `matplotlib` w programach w języku Python.
- Zdefiniujesz funkcje służące do rysowania fraktali.
- Wygenerujesz atrakcyjne grafiki.

Film samouczek

Konstrukcje fraktalne są to samopodobne obiekty, które w każdej skali mają nietrywialną strukturę. W tej sekcji zaprezentujemy program, którego zadaniem będzie wygenerowanie smoka Heighwaya w trybie tekstowym.

Zadanie to podzielimy na dwie części. Funkcja `generuj_smoka` będzie odpowiadać za generowanie ciągu liczb 0 i 1, które będą reprezentowały obroty w lewo lub w prawo naszego wirtualnego ołówka. Funkcja `drukuj_smoka` zinterpretuje ciąg dostarczony przez funkcję `generuj_smoka`, prezentując go w formie graficznej.

Polecenie 1

Napisz program, którego zadaniem będzie wygenerowanie smoka Heighwaya w trybie tekstowym.

```
1 # Tutaj dodaj własny kod. Użyj funkcji  
2 # print()
```

```
1
```

Polecenie 2

Porównaj swoje rozwiązanie z przedstawionym w filmie.



Konstrukcje fraktalne jako zastosowanie rekurencji II

Implementacja w języku Python



Film dostępny pod adresem </preview/resource/R1FCJGjEksSwn>

Źródło: Contentplus.pl Sp. z o. o., licencja: CC BY-SA 3.0.

Materiał filmowy dotyczący konstrukcji fraktali jako zastosowanie rekurencji.

Przeczytaj

Geometria fraktalna, czyli zagadnienia związane z obliczeniami wyników funkcji iterowanych IFS, znajduje zastosowanie w kompresji danych.

Przyjrzyjmy się dwóm przykładowym fraktalom zbudowanym z wykorzystaniem IFS. Będą to znane ci już paproć Barnsleya i smok Heighwaya.

Ważne!



W celu graficznego przedstawienia obrazu złożonego z wielu par punktów wygodnie jest użyć modułu `matplotlib`.

Paproć Barnsleya

Paproć Barnsleya jest fraktalem, który wyróżnia się ze względu na niezwykle podobieństwo do liści paproci występujących w naturze. Aby utworzyć taki fraktal, należy wielokrotnie (najlepiej ok. 100 000 razy) przekształcać punkt początkowy z wykorzystaniem jednej z czterech funkcji. Funkcji tych trzeba użyć w sposób losowy w proporcjach A : B : C : D wynoszących:

$$85 : 7 : 7 : 1$$

Wspomnianymi wcześniej funkcjami są:

$$f_1(x) = (0.85x + 0.04y), f_1(y) = (-0.04x + 0.85y + 1.6) - \text{element A proporcji (85)}$$

$$f_2(x) = (-0.15x + 0.28y), f_2(y) = (0.26x + 0.24y + 0.44) - \text{element B proporcji (7)}$$

$$f_3(x) = (0.20x - 0.26y), f_3(y) = (0.23x + 0.22y + 1.6) - \text{element C proporcji (7)}$$

$$f_4(x) = (0), f_4(y) = (0.16y) - \text{element D proporcji (1)}$$

Przykład 1

Oto zapisany w postaci pseudokodu algorytm służący do skonstruowania paproci Barnsleya:

```
1 paprotka_barnsleya
2     utwórz puste tablice X, Y na kolejne wartości x, y
3     x, y przypisz współrzędne początkowe (0, 0)
4
5     powtórz n razy:
6         q przypisz wartość proporcji od 0 do 100
7         w zależności od wartości q wykonaj:
8             dla q < 1 oblicz x i y bazując na funkcji (proporcj
9             dla q < 86 oblicz x i y bazując na funkcji (proporc
10            dla q < 93 oblicz x i y bazując na funkcji (proporc
11            dla pozostałych q oblicz x i y bazując na funkcji (
12
13         dodaj nowe x, y do tablic X oraz Y
14
15     narysuj obraz zawierający punkty X i Y
```

Polecenie 1

Przeanalizujemy działanie funkcji generującej rysunek paproci Barnsleya, napisanej na podstawie przedstawionego powyżej pseudokodu.

Będziemy analizować następującą funkcję:

1

Funkcję wywołamy z argumentem o wartości 5000:

2

Importujemy niezbędne funkcje:

Tworzymy puste obiekty typu `list` o nazwach `X` i `Y`. Zapiszemy w nich współrzędne punktów składających się na rysunek paprotki Barnsleya:

3

Definiujemy zmienne `x` i `y` i przypisujemy im wartości początkowe:

Zwróć uwagę na wielkość liter. W języku Python `x` jest inną zmienną (obiektem) niż `X`.

4

Inicjujemy pętlę, która zostanie wykonana 5000 razy:

Losujemy liczbę z zakresu od 0 do 100:

5

Wylosowaną liczbę wykorzystamy do określenia, której funkcji użyjemy podczas obliczania kolejnych par współrzędnych `x` oraz `y`. Zastosujemy proporcje:

$85 : 7 : 7 : 1$

6

Jeśli wylosowana liczba będzie mniejsza niż 1, wybierzemy równanie:

(ten warunek odpowiada czwartemu członowi proporcji).

Jeśli wylosowana liczba będzie mniejsza niż 86,
wyberzemy równanie:

7

(warunek ten odpowiada pierwszemu członowi
proporcji).

8

Jeśli wylosowana liczba będzie mniejsza niż 93,
wyberzemy równanie:

(warunek taki odpowiada drugiemu członowi
proporcji).

W pozostałych przypadkach wybierzemy
równanie:

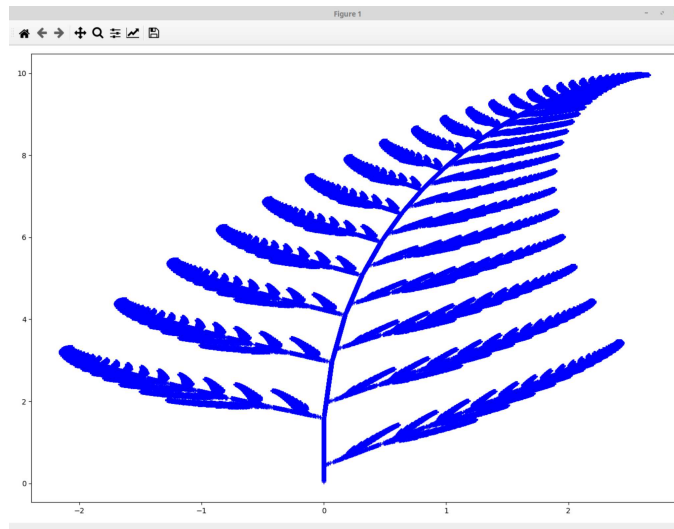
9

(ten warunek odpowiada trzeciemu członowi
proporcji).

10

Obliczone w ten sposób pary współrzędnych
zapiszemy w tablicach X i Y. Posłużą nam one
później do wygenerowania wykresu:

Ponieważ w tablicach znajduje się 5000 par
liczb x oraz y, z tylu właśnie punktów będzie
składał się rysunek paprotki Barnsleya.



Źródło: Contentplus.pl Sp. z o.o., licencja: CC BY-SA 3.0.

Wykorzystując listy punktów, rysujemy wykres:

Źródło: Contentplus.pl Sp. z o.o., licencja: CC BY-SA 3.0.

Smok Heighwaya

Smok Heighwaya jest obrazem powstającym w wyniku wielokrotnego przekształcenia punktu o współrzędnych (x, y) za pomocą stosowanych naprzemiennie funkcji:

$$f_1(x) = (-0.4x - 1), f_1(y) = (-0.4y + 0.1)$$

$$f_2(x) = (0.76x - 0.4y), f_2(y) = (0.4x + 0.76y)$$

Przykład 2

Przeanalizujemy algorytm opisujący tworzenie takiego fraktala:

```
1 smok_heighwaya
2     utwórz puste tablice X, Y na kolejne wartości x, y
3     x, y przypisz współrzędne początkowe (0, 0)
4
5     powtórz n razy:
```

```

6      wybierz losowo z jednakowym prawdopodobieństwem n 0 lub
7      jeśli n wynosi 1 wykonaj:
8          x = -0.4 * x - 1
9          y = -0.4 * y + 0.1
10     w przeciwnym wypadku wykonaj:
11         x = 0.76 * x - 0.4 * y
12         y = 0.4 * x + 0.76 * y
13
14     dodaj nowe x,y do tablic X oraz Y
15
16     narysuj obraz zawierający punkty X i Y

```

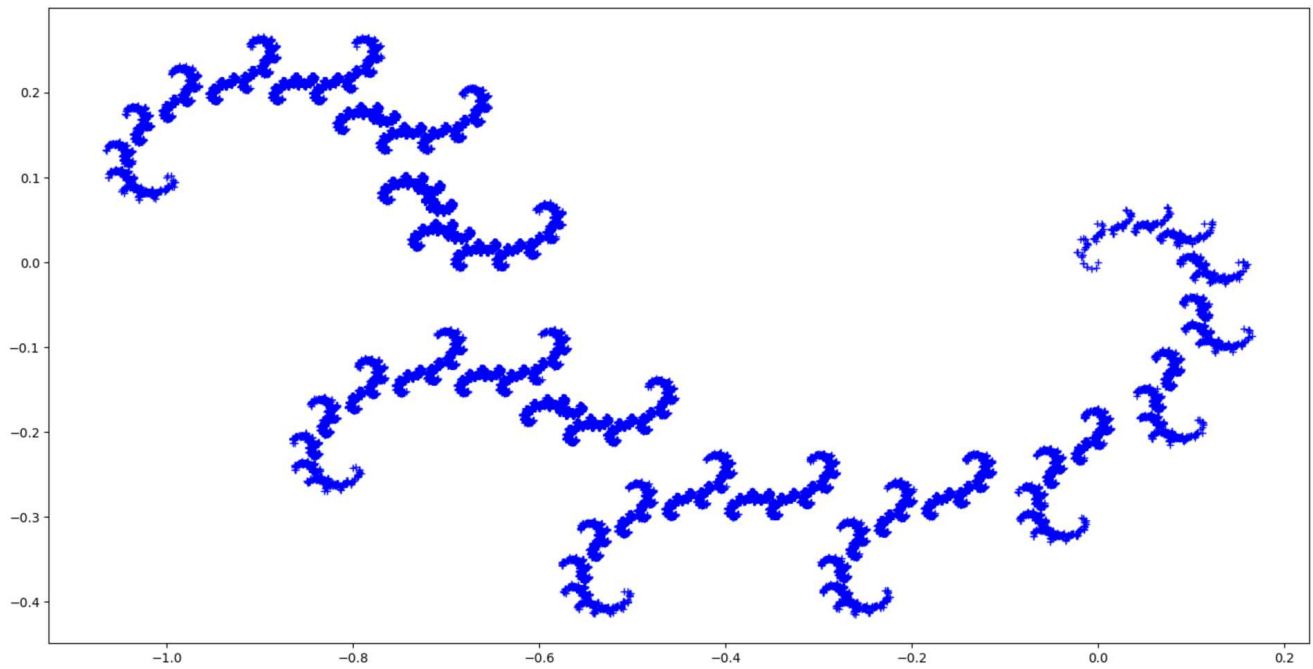
Zdefiniujmy funkcję, która zrealizuje przedstawiony pseudokod w języku Python.

```

1 def smok_heighwaya(powtorzen):
2     from random import randint
3     import matplotlib.pyplot as plt
4
5     X = []
6     Y = []
7     x, y = 0, 0
8
9     for kolejny in range(powtorzen):
10         q = randint(0, 1)
11
12         if q == 1:
13             x = -0.4 * x - 1
14             y = -0.4 * y + 0.1
15         else:
16             x = 0.76 * x - 0.4 * y
17             y = 0.4 * x + 0.76 * y
18
19         X.append(x)
20         Y.append(y)
21
22     # tworzenie wykresu - niebieskie znaki '+'
23     plt.plot(X, Y, 'b+')
24     plt.show()

```

Wynikiem działania funkcji będzie następujący obraz:



Źródło: Contentplus.pl Sp. z o.o., licencja: CC BY-SA 3.0.

Już wiesz

Podsumujmy najważniejsze elementy bieżącego e-materiału:

- funkcje IFS są używane do generowania fraktali;
- moduł `matplotlib` jest wykorzystywany do generowania rysunków w Pythonie.

Słownik

IFS

(ang. *iterated function system*) system funkcji iterowanych, wykorzystywany do konstruowania fraktali

fraktal

obiekt samopodobny (taki, którego poszczególne części są podobne do całości)

matplotlib

biblioteka służąca do przedstawienia obrazów złożonych z punktów o współrzędnych x oraz y (wykresów, histogramów, rozkładów itp.); moduł `matplotlib` nie jest dostępny w standardowej instalacji Pythona; należy go zainstalować, korzystając z mechanizmu `pip`

Sprawdź się

Pokaż ćwiczenia:   

Ćwiczenie 1



Wskaż właściwą odpowiedź.

Przykładowe fraktale zbudowane z wykorzystaniem funkcji IFS to...

☐ smok Heighwaya.

☐ róża Barnsleya

☐ dywan Sierpińskiego.

☐ płatek Kocha.

☐ paproć Barnsleya.

Ćwiczenie 2



Wskaż poprawny zapis przygotowania list do użycia ich z modułem matplotlib:

```
1 # tworzymy dwie listy dla kodu:
2 import matplotlib.pyplot as plt
3
4 lista_x = []
5 lista_y = []
6
7 plt.plot(lista_x, lista_y)
8 plt.show()
```



```
1 lista_x = [s for s in range(100)]
2 lista_y = [s*s for s in range(100)]
```



```
1 for x in range(100):
2     lista_x.append(x)
3     for y in range(100):
4         lista_y.append(y)
```



```
1 lista_x = [x in range(100)]
2 lista_y = [y in range(100)]
```

Ćwiczenie 3



Wskaż, jaką wartość zwróci polecenie:

```
s1 = {1:2, 2:3, 3:4, 4:5, 5:6, 7:8}
print(s1[s1[4]]).
```



5



IndexError: list index out of range



6



Zdefiniuj funkcję testowa (parametr), która opierając się na funkcji obliczającej punkty dla smoka Heighwaya pozwoli wyznaczyć środek smoka, czyli średnie wartości x i y (z zaokrągleniem do pięciu cyfr po przecinku). Niech opisana funkcja pomija 150 pierwszych iteracji oraz sprawdza typ obiektu parametr (powinien być nim int) oraz zwraca wynik dla wartości parametr > 200 , a dla mniejszych wartości zwróci false.

Twoje zadania

1. Zdefiniowanie funkcji testowa (parametr), działającej zgodnie z opisem.
2. Sprawdzenie, czy funkcja dla parametru 200 zwraca typ bool.
3. Sprawdzenie, czy funkcja dla parametru 20.1 zwraca typ None.
4. Sprawdzenie, czy funkcja dla parametru 260 zwraca wartość $(-0.62403, -0.04723)$.

```
1 def testowa(parametr):
2     # kolejne dwie linie muszą pozostać w programie
3     from random import seed
4     seed(4095037966)
5
6     # tutaj dodaj własny kod
7
8     return None
9
10 # niżej podaj przykład wywołania funkcji
```



Dla nauczyciela

Autor: Adam Jurkiewicz

Przedmiot: Informatyka

Temat: Konstrukcje fraktalne w języku Python

Grupa docelowa:

Szkoła ponadpodstawowa, liceum ogólnokształcące, technikum, zakres rozszerzony

Podstawa programowa:

Cele kształcenia – wymagania ogólne

I. Rozumienie, analizowanie i rozwiązywanie problemów na bazie logicznego i abstrakcyjnego myślenia, myślenia algorytmicznego i sposobów reprezentowania informacji.

II. Programowanie i rozwiązywanie problemów z wykorzystaniem komputera oraz innych urządzeń cyfrowych: układanie i programowanie algorytmów, organizowanie, wyszukiwanie i udostępnianie informacji, posługiwanie się aplikacjami komputerowymi.

Treści nauczania – wymagania szczegółowe

II. Programowanie i rozwiązywanie problemów z wykorzystaniem komputera i innych urządzeń cyfrowych.

Zakres rozszerzony. Uczeń spełnia wymagania określone dla zakresu podstawowego, a ponadto:

3) sprawnie posługuje się zintegrowanym środowiskiem programistycznym przy pisaniu, uruchamianiu i testowaniu programów;

I + II. Zakres rozszerzony. Uczeń spełnia wymagania określone dla zakresu podstawowego, a ponadto:

1) zapisuje za pomocą listy kroków, schematu blokowego lub pseudokodu, i implementuje w wybranym języku programowania, algorytmy poznane na wcześniejszych etapach oraz algorytmy:

1a) rekurencyjnego tworzenia fraktali: zbiór Cantora, drzewo binarne, dywan Sierpińskiego, płatek Kocha;

3) objaśnia, a także porównuje podstawowe metody i techniki algorytmiczne oraz struktury danych, wykorzystując przy tym przykłady problemów i algorytmów, w szczególności:

b) rekurencję (do generowania ciągów liczb, potęgowania, sortowania liczb, generowania fraktali),

Kształtowane kompetencje kluczowe:

- kompetencje cyfrowe;
- kompetencje osobiste, społeczne i w zakresie umiejętności uczenia się;
- kompetencje matematyczne oraz kompetencje w zakresie nauk przyrodniczych, technologii i inżynierii.

Cele operacyjne (językiem ucznia):

- Użyjesz modułu `matplotlib` w programach w języku Python.
- Zdefiniujesz funkcje służące do rysowania fraktali.
- Wygenerujesz atrakcyjne grafiki.

Strategie nauczania:

- konstruktywizm;
- konektywizm.

Metody i techniki nauczania:

- dyskusja;
- rozmowa nauczająca z wykorzystaniem multimediu i ćwiczeń interaktywnych.

Formy pracy:

- praca indywidualna;
- praca w parach;
- praca w grupach;
- praca całego zespołu klasowego.

Środki dydaktyczne:

- komputery z głośnikami, słuchawkami i dostępem do internetu;
- zasoby multimedialne zawarte w e-materiale;
- tablica interaktywna/tablica, pisak/kreda;
- oprogramowanie dla języka Python 3 (lub nowszej wersji), w tym PyCharm lub IDLE.

Przebieg lekcji

Przed lekcją:

1. **Przygotowanie do zajęć.** Nauczyciel loguje się na platformie i udostępnia e-materiał: „Konstrukcje fraktalne w języku Python”. Uczniowie mają zapoznać się z treściami w sekcji „Film samouczek”.

Faza wstępna:

1. Ustalenie celu lekcji i kryteriów sukcesu.
2. **Rozpoznanie wiedzy uczniów.** Uczniowie tworzą pytania dotyczące tematu zajęć, na które odpowiedzą w trakcie lekcji.

Faza realizacyjna:

1. Uczniowie w parach wykonują Polecenie 1.
2. Chętne lub wybrane osoby prezentują swoje rozwiązanie. Klasa dyskutuje na jego temat, nauczyciel wskazuje ew. problemy.
3. Uczniowie w grupach zapoznają się z filmem i porównują swoje rozwiązania.

Faza podsumowująca:

1. Wybrany uczeń podsumowuje zajęcia, zwracając uwagę na nabyte umiejętności, omawia ewentualne problemy podczas rozwiązywania ćwiczeń z programowania w języku Python.

Praca domowa:

1. Uczniowie wykonują ćwiczenia interaktywne.

Materiały pomocnicze:

- Oficjalna dokumentacja techniczna dla języka Python 3 (lub nowszej wersji).
- Oficjalna dokumentacja techniczna dla oprogramowania PyCharm lub IDLE.

Wskazówki metodyczne:

- Treści w sekcji „Przeczytaj” można wykorzystać na lekcji jako podsumowanie i utrwalenie wiedzy uczniów.