Konstrukcje fraktalne – zadania maturalne

Konstrukcje fraktalne – zadania maturalne

Źródło: Nick Cooper, domena publiczna.

Wiemy już, czym charakteryzują się fraktale i potrafimy podać ich przykłady. W e-materiale Konstrukcje fraktalne poznaliśmy paproć Barnsleya oraz smoka Heighwaya, czyli obiekty samopodobne, które można generować za pomocą algorytmów IFS, tzn. sytemu funkcji iterowanych.

W tym e-materiale zapoznamy się z przykładowym zadaniem maturalnym dotyczącym tego zagadnienia.

Implementacje w poszczególnych językach programowania przedstawiamy w e-materiałach:

- Konstrukcje fraktalne w języku C++,
- Konstrukcje fraktalne w języku Java,
- Konstrukcje fraktalne w języku Python.

> **Twoje cele**
>
> - Prześledzisz sposób rozwiązywania zadań opartych na konstrukcjach fraktalnych.
> - Poznasz sposoby reprezentacji fraktali za pomocą tekstu lub tabeli.
> - Skonstruujesz algorytm do konstrukcji graficznej fraktali.

Przeczytaj

Zadanie 1

Smok Heighwaya to zbiór punktów w prostokątnym układzie współrzędnych, którego generowanie może być wynikiem serii następujących po sobie przekształceń geometrycznych, zachodzących na danym punkcie $p = (p_x,\ p_y)$. Można je opisać za pomocą dwóch par równań:

1. $$\begin{aligned} p'_x &= 0,5p_x - 0,5p_y \\ p'_y &= 0,5p_x + 0,5p_y \end{aligned}$$

2. $$\begin{aligned} p'_x &= -0,5p_x - 0,5p_y + 1 \\ p'_y &= 0,5p_x - 0,5p_y \end{aligned}$$

Wybór kolejnego przekształcenia jest losowy z jednakowym prawdopodobieństwem. Generowanie punktów zaczynamy od punktu początkowego $p_0 = (0,\ 0)$, wykonując określoną liczbę przekształceń kolejnych punktów.

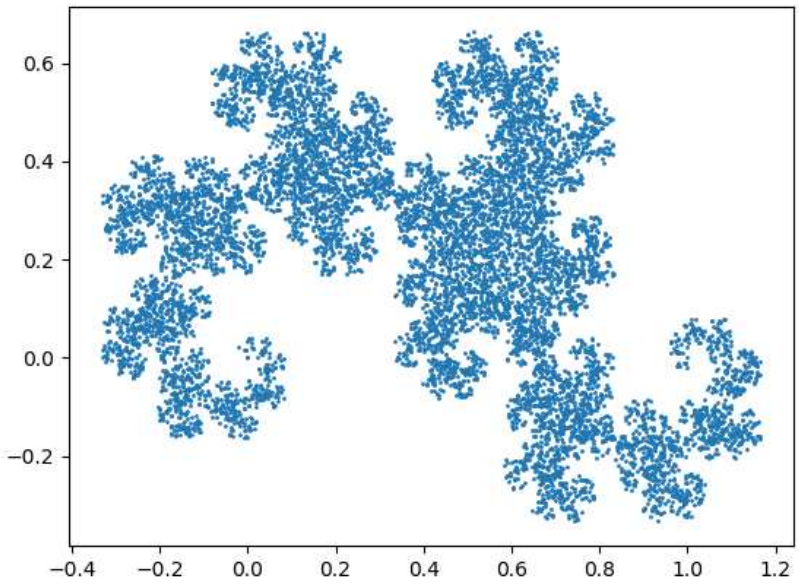
Korzystając z dostępnych narzędzi informatycznych, wykonaj poniższe polecenia.

Zadanie 1.1

W pliku `losowe_liczby.txt` znajduje się 10000 wierszy. Każdy z nich zawiera liczbę 1 lub 2. Dla każdej kolejnej liczby w pliku wykonaj przekształcenie poprzedniego punktu zgodnie z podanym sposobem generacji smoka Heighwaya (jeżeli w pliku znajduje się liczba 1, to zostaną wykonane przekształcenia oznaczone numerem 1, a w przypadku liczby 2 – przekształcenia oznaczone jako 2). Utwórz wykres, na którym zaprezentujesz wygenerowane punkty.

Plik `losowe_liczby.txt`
Plik o rozmiarze 29.30 KB w języku polskim

W wyniku przekształceń otrzymamy fraktal:



Zadanie 1.2

Wyznacz wysokość, a także szerokość utworzonego kształtu w układzie współrzędnych.

Zadanie 1.3

Obrót punktu $p = (p_x,\ p_y)$ wokół początku układu współrzędnych o kąt α można osiągnąć, stosując następujące przekształcenie:

$$p_x^{'} = p_x \cos(\alpha) - p_y \sin(\alpha)$$
$$p_y^{'} = p_x \sin(\alpha) + p_y \cos(\alpha)$$

Dokonaj obrotu utworzonych w podpunkcie 1. punktów o kąt $\alpha = 90^\circ$. Utwórz fraktal, na którym przedstawisz dwa bliźniacze smoki Heighwaya – każdy w innym kolorze.

Rozwiązanie:

Zadanie można rozwiązać w całości w arkuszu kalkulacyjnym, jednak prezentowane w tym materiale rozwiązanie będzie pokazane pod postacią pseudokodu, który należy zaimplementować w wybranym języku programowania.

Zadanie 1.1

Zacznijmy od napisania głównej pętli programu, w której wczytywane, a następnie przetwarzane będą kolejne losowe liczby z pliku `losowe_liczby.txt`.

```
dopóki plik losowe_liczby.txt się nie skończył, wykonuj:
    wczytaj liczbę a z pliku
```

W programie należy dodać także definicję aktualnie przetwarzanego punktu początkowego. W tym celu utwórzmy 2 zmienne `pX` oraz `pY` typu zmiennoprzecinkowego, które zainicjujemy wartościami 0 – zgodnie z warunkiem początkowym.

```
pX = 0.0
pY = 0.0

dopóki plik losowe_liczby.txt się nie skończył, wykonuj:
    wczytaj liczbę a z pliku
```

Wartość wczytanej zmiennej `a` zadecyduje, któremu z przekształceń 1 lub 2 należy poddać nasz punkt. W pętli głównej programu umieścimy instrukcję warunkową oraz zaimplementujmy przekształcenia podane w treści zadania. W tym celu dodamy w programie nowe zmienne `pXNowe`, `pYNowe`, które będą oznaczały współrzędne punktu po przekształceniu. Następnie obliczone wartości przypiszemy do aktualnego punktu – odpowiednio do zmiennych `pX` oraz `pY`.

```
pX = 0.0
pY = 0.0

dopóki plik losowe_liczby.txt się nie skończył, wykonuj:
    wczytaj liczbę a z pliku

    jeżeli a == 1 wykonaj:
        pXNowe = 0.5 * pX - 0.5 * pY
        pYNowe = 0.5 * pX + 0.5 * pY
    w przeciwnym razie wykonaj:
        pXNowe = - 0.5 * pX - 0.5 * pY + 1
        pYNowe = 0.5 * pX - 0.5 * pY

    pX = pXNowe
    pY = pYNowe
```

> **Ważne!**
>
> Warto zwrócić tutaj szczególną uwagę na potrzebę utworzenia nowych zmiennych pomocniczych: `pXNowe`, `pYNowe`. W przeciwnym wypadku moglibyśmy obliczyć nową współrzędną `pX`, a następnie wykorzystać ją do obliczenia `pX`.

Utworzony program jest w zasadzie gotowy. Pozostało tylko zamieścić w nim instrukcję, która odpowiadać będzie za rysowanie utworzonych punktów w układzie współrzędnych.

> Ze względu na ograniczony dostęp do bibliotek graficznych w niektórych środowiskach programistycznych w programie można zawrzeć kod przygotowujący dane dla arkusza kalkulacyjnego, w którym następnie utworzymy wykres.

```
pX = 0.0
pY = 0.0

dopóki plik losowe_liczby.txt się nie skończył, wykonuj:
    wczytaj liczbę a z pliku

    jeżeli a == 1 wykonaj:
        pXNowe = 0.5 * pX - 0.5 * pY
        pYNowe = 0.5 * pX + 0.5 * pY
    w przeciwnym razie wykonaj:
        pXNowe = - 0.5 * pX - 0.5 * pY + 1
        pYNowe = 0.5 * pX - 0.5 * pY

    pX = pXNowe
    pY = pYNowe

    rysuj punkt (pX, pY)
```

Zadanie 1.2

Aby wyznaczyć wysokość i szerokość utworzonego kształtu, będziemy potrzebowali dodatkowych zmiennych: `xMax`, `xMin`, `yMax`, `yMin`, które będą przechowywać maksymalne oraz minimalne wartości współrzędnych wśród wygenerowanych punktów. Za każdym razem po wyznaczeniu nowego punktu trzeba będzie porównywać jego współrzędne, aby wyznaczyć te, które są największe oraz najmniejsze odpowiednio na osiach OX oraz OY. Następnie pozostaje obliczyć różnicę pomiędzy `xMax` a `xMin` oraz między `yMax` a `yMax`.

```
pX = 0.0
pY = 0.0
```

```
3
4  xMax = 0.0
5  xMin = 0.0
6  yMax = 0.0
7  yMin = 0.0
8
9  dopóki plik losowe_liczby.txt się nie skończył, wykonuj:
10     wczytaj liczbę a z pliku
11
12     jeżeli a == 1 wykonaj:
13         pXNowe = 0.5 * pX - 0.5 * pY
14         pYNowe = 0.5 * pX + 0.5 * pY
15     w przeciwnym razie wykonaj:
16         pXNowe = - 0.5 * pX - 0.5 * pY + 1
17         pYNowe = 0.5 * pX - 0.5 * pY
18
19     pX = pXNowe
20     pY = pYNowe
21
22     jeżeli pX > xMax wykonaj:
23         xMax = pX
24     w przeciwnym razie wykonaj:
25         xMin = pX
26
27     jeżeli pY > yMax, wykonaj:
28         yMax = pY
29     w przeciwnym razie wykonaj:
30         yMin = pY
31
32 wysokość = yMax - yMin
33 szerokość = xMax - xMin
34 wypisz wysokość
35 wypisz szerokość
```

Dla danych dostarczonych w pliku otrzymamy w tym zadaniu odpowiedź zbliżoną do poniższej:

```
1  wysokość = 0.9961862761707053
2  szerokość = 1.495520476029701
```

Zadanie 1.3

W wykorzystywanych na egzaminie maturalnym językach programowania dostępne są funkcje oraz stałe matematyczne, dlatego w rozwiązaniu wykorzystamy funkcje trygonometryczne `sin()` oraz `cos()`, a także stałą `PI`. Ponieważ kąt, o który będziemy obracać punkty, jest stały, już na początku możemy obliczyć wartość używanych funkcji trygonometrycznych.

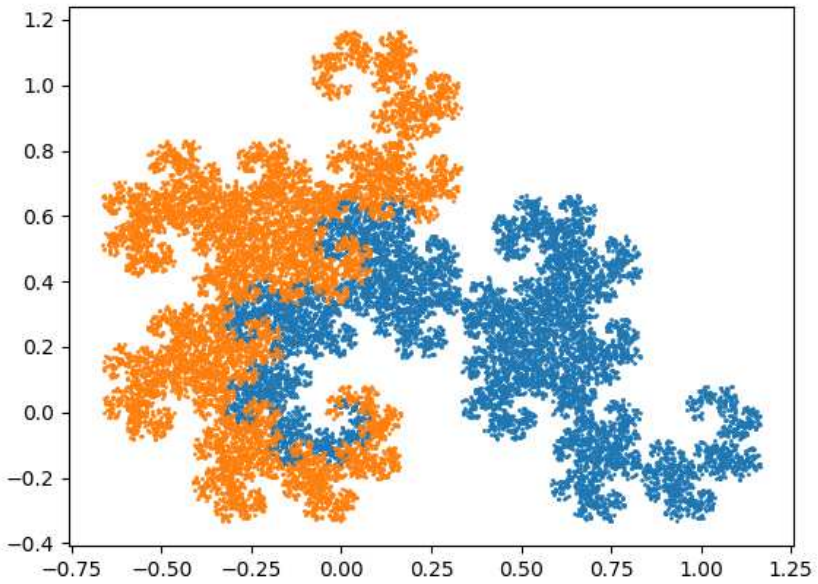
```
sin(90°) = sin(PI/2)
cos(90°) = cos(PI/2)
```

W celu realizacji przekształcenia utwórzmy dwie nowe zmienne `rX` oraz `rY`, które będą reprezentowały punkt obrócony o 90 stopni. Następnie w pseudokodzie wykorzystamy – podany w zadaniu – wzór wykorzystany do przekształcenia punktów. Kolejną czynnością będzie rysowanie punktów (oryginalnego oraz obróconego) w różnych kolorach – tak jak nakazuje rozwiązanie.

```
sin90° = sin(PI/2)
cos90° = cos(PI/2)

pX = 0.0
pY = 0.0

dopóki plik losowe_liczby.txt się nie skończył, wykonuj:
    wczytaj liczbę a z pliku

    jeżeli a == 1 wykonaj:
        pXNowe = 0.5 * pX - 0.5 * pY
        pYNowe = 0.5 * pX + 0.5 * pY
    w przeciwnym razie wykonaj:
        pXNowe = - 0.5 * pX - 0.5 * pY + 1
        pYNowe = 0.5 * pX - 0.5 * pY

    pX = pXNowe
    pY = pYNowe

    rX = pX * cos90° - pY * sin90
    rY = pX * sin90° + pY * cos90

    rysuj punkt (pX, pY) w kolorze niebieskim
```

```
24      rysuj punkt (rX, rY) w kolorze pomarańczowym
```

Po poprawnej implementacji powyższego pseudokodu, utworzony fraktal będzie prezentował się następująco:



Słownik

przekształcenie geometeryczne

funkcja przekształcająca dany zbiór punktów na inny zbiór punktów

układ współrzędnych

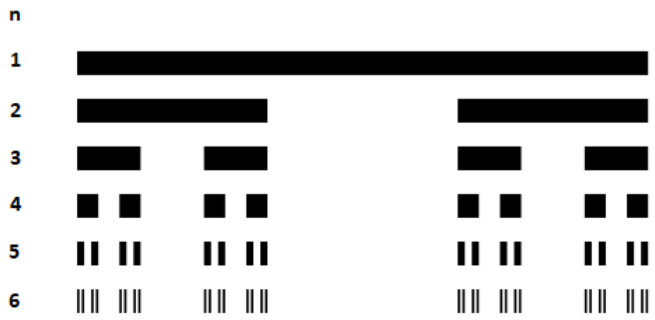
odwzorowanie, które każdemu punktowi w przestrzeni k-wymiarowej przypisuje ciąg k liczb rzeczywistych zwanych współrzędnymi

Prezentacja multimedialna

Polecenie 1

Przeanalizuj prezentację. Przedstawiono w niej przykładowe rozwiązywanie zadania typu maturalnego. Wykonaj polecenie 2.

1



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

Materiał audio dostępny pod adresem:

https://zpe.gov.pl/b/Pt7XUOITm

Zadanie 2

Zbiór Cantora to fraktal, który powstaje poprzez podzielenie odcinka (np. odcinka $\langle 0,\ 1\rangle$ na osi liczbowej) na trzy równe części, usunięcie środkowej oraz rekurencyjne powtórzenie tej procedury w stosunku do dwóch powstałych odcinków.

Inna metoda generowania tego kształtu polega na serii następujących po sobie przekształceń danej liczby p na osi liczbowej. Przekształcenia te to:

1. $p' = \frac{p}{3}$

2. $p' = \frac{p}{3} + \frac{2}{3}$

Wybór kolejnego przekształcenia jest losowy z jednakowym prawdopodobieństwem. Generowanie zaczynamy od punktu startowego

$p_0 = 0$, a następnie kontynuujemy aż do osiągnięcia określonej liczby punktów. Każdy wygenerowany punkt zaznaczamy na osi liczbowej.

W wybranej przez siebie notacji (lista kroków, wybrany język programowania, schemat blokowy) zapisz algorytm, który narysuje zbiór Cantora na odcinku $\langle 0,\ 1 \rangle$ na osi liczbowej wybraną przez siebie metodą.

Załóż, że masz dostęp do następujących funkcji:

- `losuj()` – funkcja zwracająca losową liczbę 0 lub 1 z jednakowym prawdopodobieństwem,
- `rysujPunkt(a)` – funkcja rysująca na osi liczbowej punkt a,
- `rysujOdcinek(a, b)` – funkcja rysująca dany odcinek $\langle \mathrm{a},\ \mathrm{b} \rangle$ na osi liczbowej.

Uwaga: w zapisie możesz wykorzystać dodatkowo tylko operacje arytmetyczne (dodawanie, odejmowanie, mnożenie, dzielenie, dzielenie całkowite, reszta z dzielenia), instrukcje sterujące i przypisania do zmiennych lub samodzielnie napisane funkcje zawierające wymienione operacje.

W przypadku wyboru pierwszej metody algorytm powinien tworzyć zbiór Cantora dla $n = 5$ (patrz rysunek). W przypadku wyboru drugiej metody – narysować 1000 punktów.

2

Materiał audio dostępny pod adresem:

https://zpe.gov.pl/b/Pt7XUOITm

Zaprezentujemy oba rozwiązania. Jako pierwsze omówimy rozwiązanie rekurencyjne, a następnie – losowe.

3

Materiał audio dostępny pod adresem:

https://zpe.gov.pl/b/Pt7XUOITm

Rozwiązanie rekurencyjne wymaga od nas utworzenia funkcji rekurencyjnej. Zastanówmy się, jakie parametry powinna przyjmować procedura rysująca zbiór Cantora.
W proponowanym przez nas rozwiązaniu przekazywane parametry to:

- a – początek przedziału, w którym aktualnie się znajdujemy,
- b – koniec przedziału, w którym aktualnie się znajdujemy,
- n – aktualnie rysowany stopień (patrz rysunek dołączony do treści zadania).

Funkcję nazwiemy `cantor()`.

4

Materiał audio dostępny pod adresem:

https://zpe.gov.pl/b/Pt7XUOITm

W pierwszej kolejności sprawdźmy warunek przerwania rekurencji – przypadek bazowy, czyli czy $n = 1$. W tym wypadku cały odcinek $\langle a, b \rangle$ przekazany w parametrach powinien zostać zaznaczony na osi liczbowej.

```
1 funkcja cantor(a, b, n)
2     jeżeli n == 1 wykonaj:
3         rysujOdcinek(a, b)
```

5

Materiał audio dostępny pod adresem:

https://zpe.gov.pl/b/Pt7XUOITm

Jeżeli warunek końcowy nie został spełniony, należy podzielić obecny odcinek na trzy części oraz dwie z nich poddać dalszej rekurencji ze stopniem pomniejszonym o 1. W tym celu tworzymy zmienną o nazwie `długość`, która posłuży do obliczenia długości odcinka w następnym wywołaniu rekurencji. Długość ta będzie długością obecnie przetwarzanego odcinka podzieloną przez 3.

```
długość = (b − a) / 3
```

Przedziały, które trzeba będzie poddać rekurencji to $\langle \mathrm{a},\ \mathrm{a} + \mathrm{długość} \rangle$ oraz $\langle \mathrm{b}$ - $\mathrm{długość},\ \mathrm{b} \rangle$. Przedział $\langle \mathrm{a} + \mathrm{długość},\ \mathrm{b}$ - $\mathrm{długość} \rangle$ powinien zostać pusty.

```
funkcja cantor(a, b, n)
    jeżeli n == 1 wykonaj:
        rysujOdcinek(a, b)
    w przeciwnym razie wykonaj:
        długość = (b − a) / 3
        cantor(a, a + długość, n − 1)
        cantor(b - długość, b, n − 1)
```

6

Materiał audio dostępny pod adresem:

https://zpe.gov.pl/b/Pt7XUOITm

Pozostało już tylko wywołać funkcję `cantor()` na przedziale $\langle 0,\ 1 \rangle$ z parametrem n

= 5 zgodnym z wymaganiem zadania. Całe rozwiązanie wygląda zatem następująco.

```
funkcja cantor(a, b, n)
    jeżeli n == 1 wykonaj:
        rysujOdcinek(a, b)
    w przeciwnym razie wykonaj:
        długość = (b − a) / 3
        cantor(a, a + długość, n − 1)
        cantor(b - długość, b, n − 1)

cantor(0, 1, 5)
```

7

Materiał audio dostępny pod adresem:

https://zpe.gov.pl/b/Pt7XUOITm

W drugiej metodzie, w pierwszej kolejności, implementujemy zmienną, w której będziemy przechowywać aktualne położenie punktu na osi liczbowej. Zmienną tę nazywamy p. Jako jej wartość początkową ustawiamy 0.

```
p = 0
```

Należy także wylosować liczbę determinującą, które przekształcenie wykonamy. Nazwijmy wylosowaną liczbę a. W celu losowania posługujemy się dostarczoną w zadaniu funkcją `losuj()`.

```
a = losuj()
```

8

Materiał audio dostępny pod adresem:

https://zpe.gov.pl/b/Pt7XUOITm

Ponieważ wartości zmiennej a mogą być tylko liczbą 0 lub 1, musimy stworzyć instrukcję warunkową, aby ustalić, które przekształcenie wykonamy. Zanim utworzymy tę instrukcję, zastanówmy się, jaki kod będzie odpowiadał za każde z przekształceń. Wartość liczby p po przekształceniu umieszczamy w zmiennej p1.

Kod realizujący pierwsze przekształcenie wygląda następująco:

```
p1 = p / 3
```

Kod realizujący drugie przekształcenie wygląda następująco:

```
p1 = p / 3 + 2 / 3
```

Ułamki mają taki sam mianownik, więc dodajemy je do siebie:

```
p1 = (p + 2) / 3
```

9

Materiał audio dostępny pod adresem:

https://zpe.gov.pl/b/Pt7XUOITm

W przypadku, gdy wartość zmiennej a wyniesie 0, realizujemy jedno z przekształceń, w przeciwnym razie – drugie.

```
a = losuj()

jeżeli a == 0 wykonaj:
    p1 = p / 3
w przeciwnym razie
wykonaj:
    p1 = (p + 2) / 3
```

10

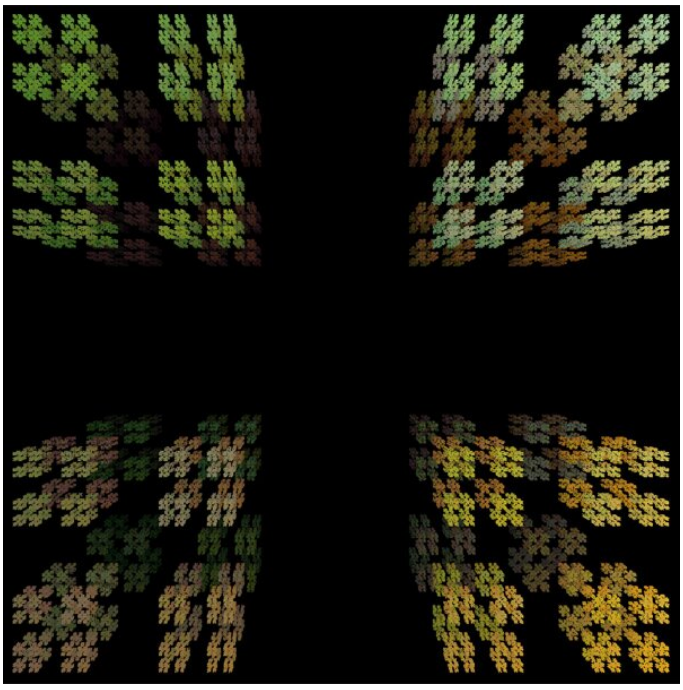
Materiał audio dostępny pod adresem:

https://zpe.gov.pl/b/Pt7XUOITm

Kolejną czynnością jest narysowanie nowego punktu na osi, a następnie ustawienie go jako nową wartość zmiennej p. W tym celu wykorzystujemy dostarczoną funkcję `rysujPunkt()`.

Rozwiązanie zapisane za pomocą pseudokodu prezentuje się następująco:

```
a = losuj()

jeżeli a == 0 wykonaj:
    p1 = p / 3
w przeciwnym razie
wykonaj:
    p1 = (p + 2) / 3

rysujPunkt(p1)
p = p1
```



Kostka Cantora – odpowiednik zbioru Cantora w trzech wymiarach.
Źródło: en.wikipedia.org, domena publiczna.

Materiał audio dostępny pod adresem:

https://zpe.gov.pl/b/Pt7XUOITm

Omówione do tej pory operacje powtarzamy 1000 razy. Końcowe rozwiązanie wygląda zatem następująco:

```
1  p = 0
2
3  dla i = 0, 1 ... 999
   wykonuj:
4      a = losuj()
5
6      jeżeli a == 0 wykonaj:
7          p1 = p / 3
8      w przeciwnym razie
   wykonaj:
9          p1 = (p + 2) / 3
10
11     rysujPunkt(p1)
12     p = p1
```

Polecenie 2

Zapisz algorytm, który narysuje zbiór Cantora na odcinku <0, 1> na osi liczbowej.

```
1
```

Sprawdź się

Zadanie 3

Słowa Fibonacciego to słowa S_n zdefiniowane przez następującą zależność.

$$S_n = \begin{cases} 0 & \text{dla n} = 0 \\ 01 & \text{dla n} = 1 \\ S_{n-1}S_{n-2} & \text{dla n} > 1 \end{cases}$$

Kilka pierwszych słów Fibonacciego wypisano poniżej:

$$S_0 = 0$$

$$S_1 = 01$$

$$S_2 = 010$$

$$S_3 = 01001$$

$$S_4 = 01001010$$

$$S_5 = 0100101001001$$

Fraktal Fibonacciego to kształt powstający przez odpowiednią graficzną interpretację słów Fibonacciego, według następujących zasad:

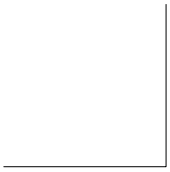
Dla każdego symbolu na pozycji k (numerujemy od zera):

1. Rysuj odcinek w obecnym kierunku.
2. Jeżeli znak na pozycji k to 0:
    - skręć w lewo, jeśli k jest parzyste,
    - skręć w prawo, jeśli k jest nieparzyste.

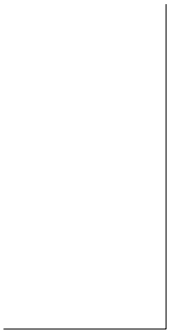
Rysowanie zaczynamy z dolnego lewego rogu, początkowy kierunek ustawiamy na prawo.

Fraktale Fibonacciego dla kilku pierwszych stopni pokazano poniżej:

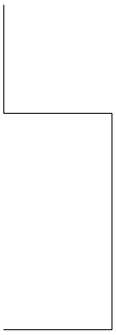
Fraktal Fibonacciego dla słowa "0"



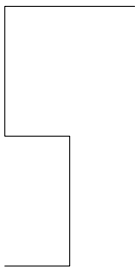
Fraktal Fibonacciego dla słowa "01"



Fraktal Fibonacciego dla słowa "010"



Fraktal Fibonacciego dla słowa "01001"



Fraktal Fibonacciego dla słowa "01001010"

Napisz program, który dla każdej liczby z tablicy `dane` wypisze słowo Fibonacciego stopnia `k`. Następnie dla najdłuższego z wypisanych słów utworzy fraktal Fibonacciego.

Przetestuj działanie programu dla tablicy `dane` o następującej zawartości:

```
4
3
12
8
```

Specyfikacja problemu:

Dane:

- `n` – liczba naturalna; liczba słów do wygenerowania
- `dane[0..n - 1]` – tablica zawierająca `n` liczb, dla których należy wypisać słowo Fibonacciego stopnia `k`

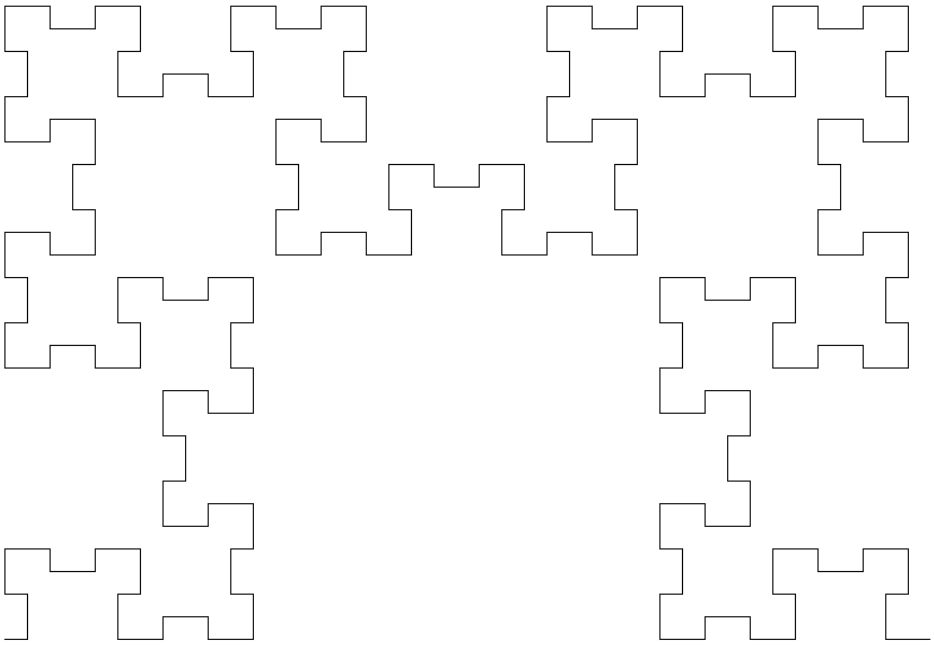
Wynik:

Słowa Fibonacciego oddzielone znakiem nowej linii oraz kształt (utworzony np. graficznie w tabeli lub w konsoli) fraktala Fibonacciego dla najdłuższego wypisanego słowa Fibonacciego.

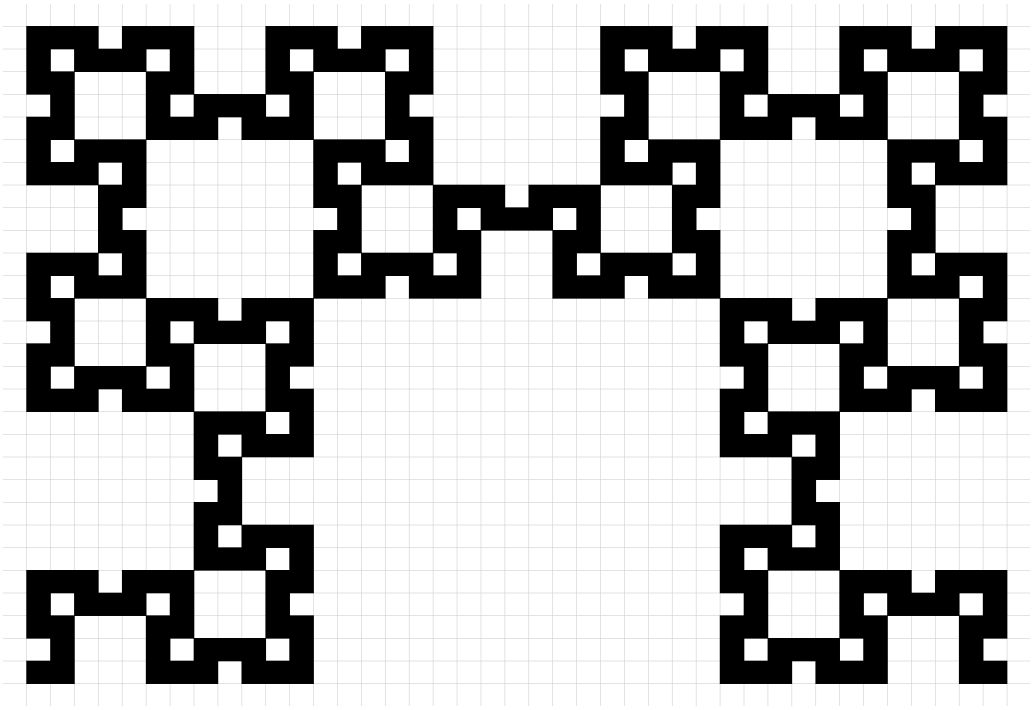
```
01001010
```

```
01001
0100101001001010010100100101001001010010100100101001010010010100101001001010
0100101001001010010100100101001001010010100100101001010010010100101001001010
```

Przykłady poprawnie utworzonych fraktali dla powyższego zadania:



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



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

```
### ###   ### ###       ### ###   ### ###
# ### #   # ### #       # ### #   # ### #
##   ##   ##   ##       ##   ##   ##   ##
 #   # ### #   #         #   # ### #   #
```

```
##   ### ###   ##        ##   ### ###   ##
# ###       ### #       # ###       ### #
### #       # ###       ### #       # ###
   ##       ##   ### ###   ##       ##
   #         #   # ### #   #         #
   ##       ##   ##   ##   ##       ##
### #       # ### #   # ### #       # ###
# ###       ### ###   ### ###       ### #
##   ### ###                 ### ###   ##
 #   # ### #                 # ### #   #
##   ##   ##                 ##   ##   ##
# ### #   #                   #   # ### #
### ###   ##                 ##   ### ###
       ### #                 # ###
       # ###                 ### #
       ##                       ##
        #                       #
       ##                       ##
       # ###                 ### #
       ### #                 # ###
### ###   ##                 ##   ### ###
# ### #   #                   #   # ### #
##   ##   ##                 ##   ##   ##
 #   # ### #                 # ### #   #
##   ### ###                 ### ###   ##
```

> **Ważne!**
>
> Chcąc wygenerować fraktal w tabeli lub tekstowo, możesz założyć, że jego rozmiar nie będzie większy niż 100×100.
> **Ważne!**
>
> W poniższych programach sprawdzających nie ma możliwości weryfikacji poprawności utworzonej grafiki. Fraktal należy zatem zrealizować w postaci tekstowej w taki sposób, jak powyżej.

Pokaż ćwiczenia:

> JĘZYK C++

Ćwiczenie 1

Twoje zadania

1. Program dla każdej liczby k z tablicy dane wypisuje k-te słowo Fibonacciego. Dla najdłuższego słowa tworzy tekstowy fraktal Fibonacciego.

```cpp
#include <iostream>

int main() {
    int n = 4;

    int dane[] = { 4, 3, 12, 8 };
}
```

```
1
```

Pobierz poniższy plik. W pierwszej linijce zawiera on liczbę testów n. Każda kolejna linijka zawiera stopień słowa Fibonacciego, które należy wypisać. Napisz program w środowisku programistycznym na swoim komputerze, w którym załadujesz dane z pobranego pliku, wypiszesz odpowiednie słowa Fibonacciego oraz utworzysz fraktal Fibonacciego dla najdłuższego wypisanego słowa Fibonacciego.

Plik o rozmiarze 48.00 B w języku polskim

JĘZYK JAVA

Ćwiczenie 2

Twoje zadania

1. Program dla każdej liczby k z tablicy dane wypisuje k-te słowo Fibonacciego. Dla najdłuższego słowa tworzy tekstowy fraktal Fibonacciego.

```java
public class Main {
    public static void main(String[] args) {
        int n = 4;

        int[] dane = { 4, 3, 12, 8 };
    }
}
```

```
1
```

Pobierz poniższy plik. W pierwszej linijce zawiera on liczbę testów n. Każda kolejna linijka zawiera stopień słowa Fibonacciego, które należy wypisać. Napisz program w środowisku programistycznym na swoim komputerze, w którym załadujesz dane z pobranego pliku, wypiszesz odpowiednie słowa Fibonacciego oraz utworzysz fraktal Fibonacciego dla najdłuższego wypisanego słowa Fibonacciego.

Plik o rozmiarze 48.00 B w języku polskim

JĘZYK PYTHON

Ćwiczenie 3

Twoje zadania

1. Program dla każdej liczby k z tablicy `dane` wypisuje k-te słowo Fibonacciego. Dla najdłuższego słowa tworzy tekstowy fraktal Fibonacciego.

```
n = 4
dane = [4, 3, 12, 8]
```

```
```

Pobierz poniższy plik. W pierwszej linijce zawiera on liczbę testów n. Każda kolejna linijka zawiera stopień słowa Fibonacciego, które należy wypisać. Napisz program w środowisku programistycznym na swoim komputerze, w którym załadujesz dane z pobranego pliku, wypiszesz odpowiednie słowa Fibonacciego oraz utworzysz fraktal Fibonacciego dla najdłuższego wypisanego słowa Fibonacciego.

Plik o rozmiarze 48.00 B w języku polskim

Dla nauczyciela

Autor: Maurycy Gast
Przedmiot: Informatyka

Temat: Konstrukcje fraktalne – zadania maturalne

Grupa docelowa:

Szkoła ponadpodstawowa, liceum ogólnokształcące, technikum, zakres podstawowy

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):

- Prześledzisz sposób rozwiązywania zadań opartych na konstrukcjach fraktalnych.
- Poznasz sposoby reprezentacji fraktali za pomocą tekstu lub tabeli.
- Skonstruujesz algorytm do konstrukcji graficznej fraktali.

Strategie nauczania:

- konstruktywizm;
- konektywizm.

Metody i techniki nauczania:

- dyskusja;
- rozmowa nauczająca z wykorzystaniem multimedium i ćwiczeń interaktywnych;
- ćwiczenia praktyczne.

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 C++, w tym kompilator GCC/G++ 4.5 (lub nowszej wersji) i Code::Blocks 16.01 (lub nowszej wersji), Orwell Dev-C++ 5.11 (lub nowszej wersji) lub Microsoft Visual Studio;

- oprogramowanie dla języka Java SE 8 (lub nowszej wersji), w tym Eclipse 4.4 (lub nowszej wersji);
- 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 – zadania maturalne". Nauczyciel prosi uczniów o zapoznanie się z treściami w sekcji „Przeczytaj".

Faza wstępna:

1. Nauczyciel wprowadza uczniów szczegółowo w temat lekcji i jej cele. Może posłużyć się wyświetloną na tablicy zawartością sekcji „Wprowadzenie".
2. **Rozpoznanie wiedzy uczniów.** Nauczyciel zadaje uczniom pytanie dotyczące ich aktualnego stanu wiedzy w obszarze poruszanego tematu, opartego o programowanie.

Faza realizacyjna:

1. **Praca z tekstem.** Nauczyciel ocenia, na podstawie informacji na platformie, stan przygotowania uczniów do zajęć. Jeżeli jest ono niewystarczające, prosi wybraną osobę o przedstawienie najważniejszych informacji z sekcji „Przeczytaj".
   Uczniowie pracując w parach, implementują w wybranym języku programowania rozwiązanie zadań 1.1, 1.2 i 1.3. Nauczyciel weryfikuje poprawność rozwiązań.
2. **Praca z multimedium.** Nauczyciel wyświetla zawartość sekcji „Prezentacja multimedialna". Uczniowie wspólnie analizują prezentację, w której przedstawiono przykładowe rozwiązywanie zadania typu maturalnego. Z prezentacji dowiedzą się, jak w wybranej przez siebie notacji (lista kroków, wybrany język programowania, schemat blokowy) zapisać algorytm, który narysuje zbiór Cantora na odcinku $\langle 0,\ 1\rangle$ na osi liczbowej wybraną przez uczniów metodą.
3. **Ćwiczenia umiejętności.** Uczniowie, pracując w parach rozwiązują w wybranym języku programowania zadanie 3 z sekcji „Sprawdź się". Nauczyciel sprawdza poprawność rozwiązań.

Faza podsumowująca:

1. Na koniec zajęć nauczyciel raz jeszcze wyświetla na tablicy temat lekcji i cele zawarte w sekcji „Wprowadzenie". W odniesieniu do ich realizacji dokonuje szczegółowej oceny rozwiązania zastosowanego przez wybranego ucznia.
2. Na koniec zajęć z programowania w Javie nauczyciel prosi uczniów o rozwinięcie zdania: „Na dzisiejszych zajęciach nauczyłam/łem się jak…".

Praca domowa:

1. Uczniowie wykonują polecenie 2 z sekcji „Prezentacja multimedialna”. Ćwiczenie realizują w jednym z dostępnych języków programowania.

Materiały pomocnicze:

- Oficjalna dokumentacja techniczna dla języka C++.
- Oficjalna dokumentacja techniczna dla kompilatora GCC/G++ 4.5 (lub nowszej wersji).
- Oficjalna dokumentacja techniczna dla oprogramowania Code::Blocks 16.01 (lub nowszej wersji), Orwell Dev-C++ 5.11 (lub nowszej wersji) lub Microsoft Visual Studio.
- Oficjalna dokumentacja techniczna dla oprogramowania Eclipse 4.4 (lub nowszej wersji).
- Oficjalna dokumentacja techniczna dla oprogramowania PyCharm lub IDLE.

Wskazówki metodyczne:

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