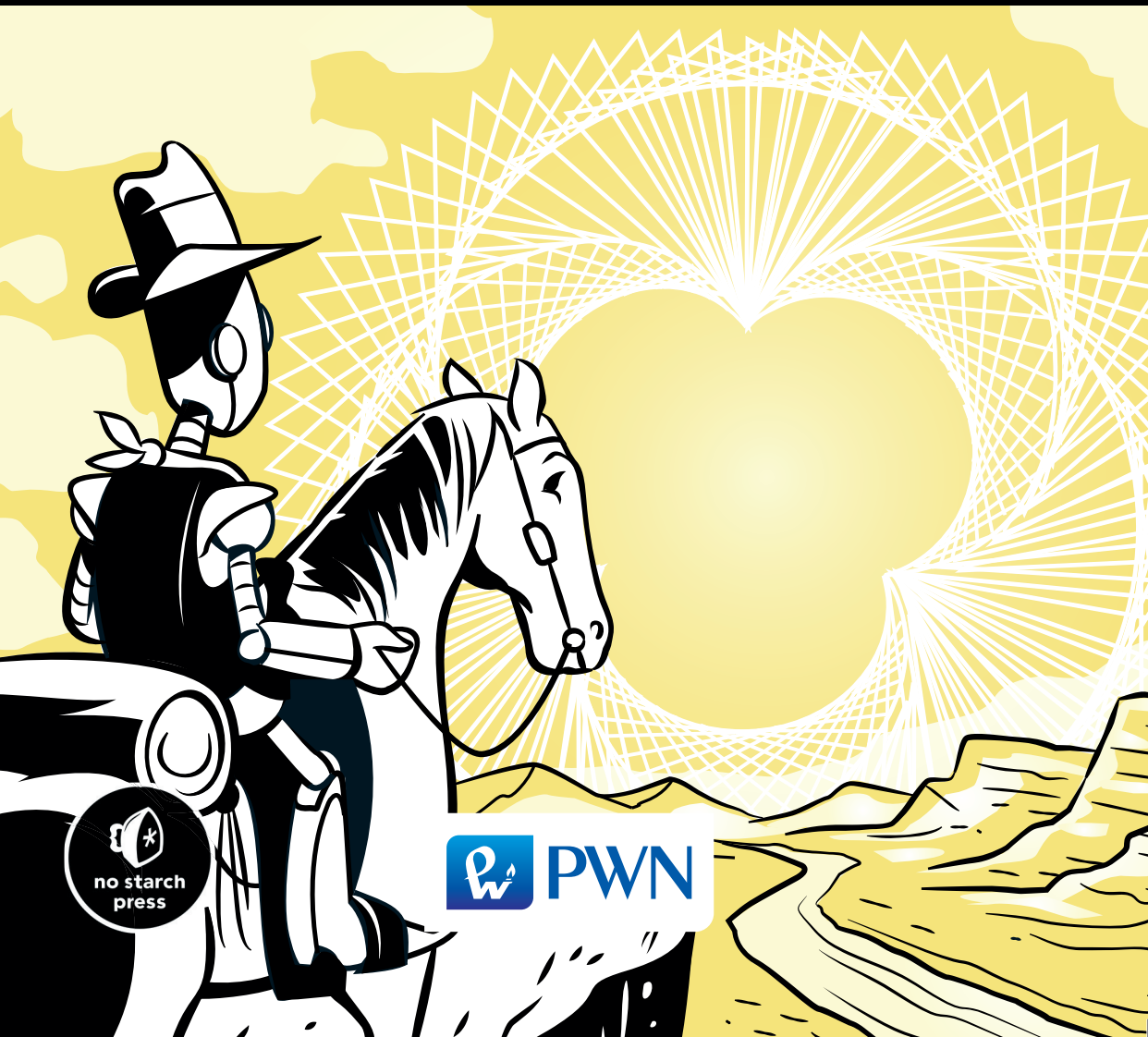


# MATEMATYCZNE PRZYGODY Z PYTHONEM

ILUSTROWANY PODRĘCZNIK DO NAUKI  
MATEMATYKI PRZEZ PROGRAMOWANIE

PETER FARRELL



# **MATEMATYCZNE PRZYGODY Z PYTHONEM**



# MATEMATYCZNE PRZYGODY Z PYTHONEM

ILUSTROWANY PODRĘCZNIK DO NAUKI  
MATEMATYKI PRZEZ PROGRAMOWANIE

PETER FARRELL



Dane oryginału

Copyright © 2019 by Peter Farrell. Title of English-language original: *Math Adventures with Python: An Illustrated Guide to Exploring Math with Code*, ISBN 978-1-59327-867-0, published by No Starch Press. Polish-language edition copyright © 2019 by Polish Scientific Publishers PWN Wydawnictwo Naukowe PWN Spółka Akcyjna. All rights reserved.

Przekład **Natalia Chounlamany-Turalska** na zlecenie **WITKOM Witold Sikorski**

Projekt okładki polskiego wydania **Joanna Andryjowicz** na podstawie oryginału

Wydawca **Edyta Kawala**

Redaktor prowadzący **Jolanta Kowalczuk**

Redaktor **Małgorzata Dąbkowska-Kowalik**

Koordynator produkcji **Anna Bączkowska**

Skład i łamanie **Tomasz Wojtanowicz**

Rysunki: 10.2 opublikowano za zgodą Acadac (rysunek na podstawie Avsa);  
11.19 – za zgodą Fabienne Serriere

Konsultacja merytoryczna

**dr inż. Maciej Bartoszuk**, Politechnika Warszawska

Zastrzeżonych nazw firm i produktów użyto w książce wyłącznie w celu identyfikacji.

Copyright © for the Polish edition by Wydawnictwo Naukowe PWN SA  
Warszawa 2019

ISBN 978-83-01-20823-3

Wydanie I

Warszawa 2019

Wydawnictwo Naukowe PWN SA  
02-460 Warszawa, ul. Gottlieba Daimlera 2  
tel. 22 69 54 321, faks 22 69 54 288  
infolinia 801 33 33 88  
e-mail: [pwn@pwn.com.pl](mailto:pwn@pwn.com.pl), [reklama@pwn.pl](mailto:reklama@pwn.pl)  
[www.pwn.pl](http://www.pwn.pl)

Druk i oprawa Totem.com.pl

Książkę dedykuję wszystkim moim uczniom,  
od których tak wiele się nauczyłem.

## O autorze

Peter Farrell przez osiem lat był nauczycielem matematyki, a pracę tę rozpoczął jako wolontariusz Korpusu Pokoju w Kenii. Później przez trzy lata pracował jako nauczyciel informatyki. Lektura książki Seymoura Paperta, *Burze mózgow*, oraz poznanie Pythona, które zawdzięcza jednemu z uczniów, zainspirowały go do wzbogacenia lekcji matematyki o elementy programowania. Autor książki jest ogromnym zwolennikiem wykorzystywania komputerów, które sprawiają, że nauka matematyki jest bardziej praktyczna, ciekawa i rozwijająca.

## O recenzencie technicznym

Paddy Gaunt ukończył studia inżynierskie kilka tygodni po narodzinach pierwszego komputera IBM PC i powiązanego z nim systemu MS DOS. Dużą część reszty kariery poświęcił zastosowaniu koncepcji matematycznych i technicznych w praktycznych programach. Ostatnio ponowił współpracę z Uniwersytetem w Cambridge i objął stanowisko głównego programisty pi3d, modułu Pythona dla grafiki 3D, zaprojektowanego początkowo z myślą o komputerach Raspberry Pi.

# SKRÓCONY SPIS TREŚCI

Podziękowania .....	xv
Wprowadzenie .....	xvii
Część I: Zaprzęgamy Pythona do pracy	
Rozdział 1: Rysowanie wielokątów przy użyciu modułu turtle .....	3
Rozdział 2: Zamienianie żmudnej arytmetyki w zabawę przy użyciu list i pętli .....	19
Rozdział 3: Zgadywanie i sprawdzanie przy użyciu instrukcji warunkowych .....	37
Część II: Wjeżdżamy na terytorium matematyki	
Rozdział 4: Przekształcanie i przechowywanie liczb zgodnie z zasadami algebry .....	53
Rozdział 5: Przekształcanie figur przy użyciu geometrii .....	77
Rozdział 6: Tworzenie oscylacji przy użyciu trygonometrii .....	103
Rozdział 7: Liczby zespolone .....	127
Rozdział 8: Wykorzystywanie macierzy do grafiki komputerowej i układów równań. . . .	145
Część III: Przecieramy szlaki	
Rozdział 9: Budowanie obiektów przy użyciu klas .....	175
Rozdział 10: Tworzenie fraktali przy użyciu rekurencji .....	201
Rozdział 11: Automaty komórkowe .....	225
Rozdział 12: Rozwiązywanie problemów przy użyciu algorytmów genetycznych .....	247
Indeks .....	273





# SZCZEGÓŁOWY SPIS TREŚCI

<b>Podziękowania</b>	<b>xv</b>
<b>Wprowadzenie</b>	<b>xvii</b>
Problemy ze szkolną matematyką	xviii
○ tej książce	xx
Kto powinien skorzystać z tej książki	xxi
Co można znaleźć w tej książce?	xxi
Pobieranie i instalowanie Pythona	xxii
Uruchamianie IDLE	xxiii
Instalowanie oprogramowania Processing	xxiii

## CZĘŚĆ I: ZAPRZĘGAMY PYTHONA DO PRACY

<b>1</b>	
<b>Rysowanie wielokątów przy użyciu modułu turtle</b>	<b>3</b>
Moduł turtle Pythona	4
Importowanie modułu turtle	4
Ruch żółwiem	5
Zmiana kierunku	6
Powtarzanie kodu za pomocą pętli	7
Wykorzystywanie pętli for	7
Wykorzystywanie pętli for do rysowania kwadratu	9
Tworzenie skrótów za pomocą funkcji	9
Wykorzystywanie zmiennych do rysowania kształtów	10
Wykorzystywanie zmiennych w funkcjach	11
Błędy zmiennych	12
Trójkąt równoboczny	13
Pisanie funkcji triangle()	13
Zmienianie zmiennych	14
Podsumowanie	16

## 2

### Zamienianie żmudnej arytmetyki w zabawę przy użyciu list i pętli

19

Podstawowe operatory	19
Działania na zmiennych	20
Wykorzystywanie operatorów do pisania funkcji average()	21
Pamiętaj o kolejności wykonywania działań!	21
Wykorzystywanie nawiasów w połączeniu z operatorami	22
Typy danych w Pythonie	22
Liczby całkowite i zmiennoprzecinkowe	22
Łańcuchy	23
Wartości logiczne	24
Sprawdzanie typów danych	24
Wykorzystywanie list do przechowywania wartości	25
Dodawanie elementów do list	26
Działania na listach	26
Usuwanie elementów z listy	27
Stosowanie list w pętlach	27
Uzyskiwanie dostępu do poszczególnych elementów przy użyciu indeksów	28
Uzyskiwanie dostępu do indeksu i wartości przy użyciu funkcji enumerate()	29
Indeksy rozpoczynają się od zera	29
Uzyskiwanie dostępu do zakresu elementów listy	29
Znajdowanie indeksu elementu	30
Łańcuchy również mają indeksy	31
Sumowanie	31
Tworzenie zmiennej sumy bieżącej	32
Pisanie funkcji mysum()	33
Znajdowanie średniej dla listy liczb	33
Podsumowanie	34

## 3

### Zgadywanie i sprawdzanie przy użyciu instrukcji warunkowych

37

Operatory porównań	38
Podejmowanie decyzji przy użyciu instrukcji if oraz else	38
Wykorzystywanie instrukcji warunkowych do znajdowania dzielników	40
Pisanie programu factors.py	40
Wędrujący żółw	41
Tworzenie gry w zgadywanie liczb	43
Tworzenie generatora liczb losowych	44
Zbieranie danych	44
Zamiana danych wejściowych na liczby całkowite	45
Wykorzystywanie instrukcji warunkowych do sprawdzania prób odgadnięcia	45
Wykorzystywanie pętli do ponawiania prób odgadnięcia	46
Wskazówki dotyczące zgadywania	47
Znajdowanie pierwiastka kwadratowego	48
Stosowanie logiki gry w zgadywanie liczb	48
Pisanie funkcji squareroot()	49
Podsumowanie	50

## CZĘŚĆ II: WJEŹDŹAMY NA TERYTORIUM MATEMATYKI

### 4

#### Przekształcanie i przechowywanie liczb zgodnie z zasadami algebry

**53**

Rozwiązywanie równań pierwszego stopnia	54
Znajdowanie wzoru dla równań pierwszego stopnia	55
Pisanie funkcji <code>equation()</code>	56
Zastosowanie <code>print()</code> zamiast <code>return</code>	57
Rozwiązywanie równań wyższego stopnia	58
Rozwiązywanie równań kwadratowych przy użyciu funkcji <code>quad()</code>	59
Rozwiązywanie równań sześciennych przy użyciu funkcji <code>plug()</code>	60
Graficzne rozwiązywanie równań	61
Rozpoczęcie pracy z Processingiem	61
Tworzenie własnego narzędzia graficznego	63
Rysowanie wykresu równania	69
Znajdowanie pierwiastków przy użyciu metody zgadywania i sprawdzania	73
Pisanie funkcji <code>guess()</code>	73
Podsumowanie	75

### 5

#### Przekształcanie figur przy użyciu geometrii

**77**

Rysowanie okręgu	78
Określanie położenia przy użyciu współrzędnych	79
Funkcje przekształceń	80
Translacje obiektów z wykorzystaniem funkcji <code>translate()</code>	80
Obracanie obiektów z wykorzystaniem funkcji <code>rotate()</code>	83
Rysowanie okręgu okręgów	84
Rysowanie okręgu kwadratów	85
Animowanie obiektów	86
Tworzenie zmiennej <code>t</code>	86
Obracanie poszczególnych kwadratów	87
Zapisywanie orientacji przy użyciu funkcji <code>pushmatrix()</code> i <code>popmatrix()</code>	88
Obrót wokół środka	89
Tworzenie interaktywnej siatki tęczy	89
Rysowanie siatki obiektów	90
Dodawanie kolorów tęczy do obiektów	91
Rysowanie złożonych wzorów przy użyciu trójkątów	93
Trójkąt 30-60-90	94
Rysowanie trójkąta równobocznego	96
Rysowanie wielu obracających się trójkątów	98
Przesunięcia fazowe obrotu	99
Kończenie projektu	100
Podsumowanie	102

## 6

### **Tworzenie oscylacji przy użyciu trygonometrii** **103**

Wykorzystywanie trygonometrii do obrotów i oscylacji	105
Pisanie funkcji do rysowania wielokątów	106
Rysowanie sześciokąta za pomocą pętli	107
Rysowanie trójkąta równobocznego	109
Tworzenie fal sinusoidalnych	110
Pozostawianie śladu	113
Korzystanie z wbudowanej funkcji Pythona enumerate()	114
Tworzenie programu spirografu	116
Rysowanie mniejszego okręgu	116
Wprawianie mniejszego okręgu w ruch	117
Tworzenie harmonografów	120
Pisanie programu harmonografu	121
Natychmiastowe wypełnianie listy	123
Co dwa wahadła to nie jedno	124
Podsumowanie	126

## 7

### **Liczby zespolone** **127**

Układ współrzędnych płaszczyzny zespolonej	128
Dodawanie liczb zespolonych	129
Mnożenie liczby zespolonej przez $i$	130
Mnożenie dwóch liczb zespolonych	131
Pisanie funkcji magnitude()	132
Tworzenie zbioru Mandelbrota	132
Pisanie funkcji mandelbrot()	135
Dodawanie koloru do zbioru Mandelbrota	139
Tworzenie zbioru Julii	141
Pisanie funkcji julia()	141
Podsumowanie	143

## 8

### **Wykorzystywanie macierzy do grafiki komputerowej i układów równań** **145**

Co to jest macierz?	146
Dodawanie macierzy	146
Mnożenie macierzy	147
Kolejność ma znaczenie w mnożeniu macierzy	151
Rysowanie kształtów 2D	151
Przekształcanie macierzy	154
Transpozycja macierzy	156
Obracanie macierzy w czasie rzeczywistym	160
Tworzenie kształtów 3D	161
Tworzenie macierzy obrotu	162
Rozwiązywanie układów równań przy użyciu macierzy	166
Metoda eliminacja Gaussa	167
Pisanie funkcji gauss()	168
Podsumowanie	172

## CZĘŚĆ III: PRZECIERAMY SZLAKI

### 9

#### Budowanie obiektów przy użyciu klas 175

Program z odbijającymi się piłkami . . . . .	177
Wprawianie piłki w ruch . . . . .	178
Odbijanie piłki od ścian . . . . .	179
Tworzenie wielu piłek bez użycia klas . . . . .	181
Tworzenie obiektów przy użyciu klas . . . . .	182
Program z pasącymi się owcami . . . . .	186
Pisanie klasy owcy . . . . .	186
Programowanie ruchu owcy . . . . .	187
Tworzenie właściwości energii . . . . .	189
Tworzenie trawy przy użyciu klas . . . . .	189
Zmiana koloru trawy na brązowy po jej zjedzeniu . . . . .	192
Nadawanie każdej owcy losowego koloru . . . . .	194
Programowanie rozmnażania się owiec . . . . .	196
Odrastanie trawy . . . . .	197
Zapewnianie przewagi ewolucyjnej . . . . .	198
Podsumowanie . . . . .	200

### 10

#### Tworzenie fraktali przy użyciu rekurencji 201

Długość wybrzeża . . . . .	202
Co to jest rekurencja? . . . . .	203
Pisanie funkcji factorial() . . . . .	203
Budowanie drzewa fraktalnego . . . . .	204
Płatek Kocha . . . . .	209
Pisanie funkcji segment() . . . . .	210
Trójkąt Sierpińskiego . . . . .	214
Fraktal bazujący na kwadracie . . . . .	216
Krzywa smocza . . . . .	220
Podsumowanie . . . . .	224

### 11

#### Automaty komórkowe 225

Tworzenie automatu komórkowego . . . . .	226
Pisanie klasy Cell . . . . .	228
Zmienianie rozmiaru wszystkich komórek . . . . .	230
Programowanie rozwoju automatu komórkowego . . . . .	231
Umieszczanie komórek w macierzy . . . . .	232
Tworzenie listy komórek . . . . .	233
Listy Pythona są dziwne . . . . .	234
Notacja indeksów listy . . . . .	235
Automatyczny rozwój automatu komórkowego . . . . .	238
Zasady gry w życie . . . . .	238
Elementarny automat komórkowy . . . . .	241
Podsumowanie . . . . .	246

## 12

<b>Rozwiązywanie problemów przy użyciu algorytmów genetycznych</b>	<b>247</b>
Wykorzystywanie algorytmów genetycznych do zgadywania fraz . . . . .	248
Pisanie funkcji makelist() . . . . .	248
Testowanie funkcji makelist() . . . . .	249
Pisanie funkcji score() . . . . .	250
Pisanie funkcji mutate() . . . . .	250
Generowanie liczby losowej . . . . .	251
Rozwiązywanie problemu komiwojażera . . . . .	254
Wykorzystywanie algorytmów genetycznych . . . . .	254
Pisanie metody calclength() . . . . .	260
Testowanie metody calclength() . . . . .	261
Losowe trasy . . . . .	262
Stosowanie idei mutacji z programu do odgadywania fraz . . . . .	265
Modyfikowanie dwóch numerów z listy . . . . .	265
Krzyżowanie w celu ulepszenia tras . . . . .	269
Podsumowanie . . . . .	272

<b>Indeks</b>	<b>273</b>
---------------	------------

## PODZIĘKOWANIA

Chciałbym podziękować: Donowi „The Mathman” Cohenowi za pokazanie mi, ile zabawy i wyzwań może dostarczyć nauka prawdziwej matematyki, Seymourowi Papertowi za udowodnienie, że programowanie powinno wchodzić w zakres lekcji matematyki, Markowi Millerowi za szansę wcielenia pomysłów w życie, Hanselowi Lynn i Waynowi Teng z theCoderSchool, dzięki którym mogłem kontynuować zabawę, programując ze studentami, a także Kenowi Hawthornowi za udostępnienie moich projektów w swojej szkole. Dziękuję redaktorom No Starch, Annie Choi, Liz Chadwick oraz Meg Sneeringer, za pomoc w ulepszaniu tej książki oraz Paddiemu Gauntowi, którego wkład jest widoczny w całej książce. Bez Was ta książka nigdy by nie powstała. Dziękuję również wszystkim, którzy mi odmówili – daliście mi energię do kontynuowania pracy. Na zakończenie dziękuję Lucy za to, że zawsze we mnie wierzy.





## WPROWADZENIE

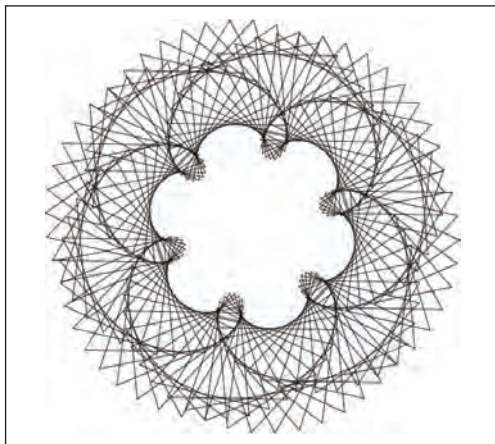


Które podejście do nauki przedstawione na rysunku 1 bardziej Ci odpowiada? Po lewej mamy przykład tradycyjnej metody nauczania matematyki – za pomocą definicji, twierdzeń i dowodów. Ta metoda wymaga długich opisów i dziwnych symboli. Nic nie wskazuje na to, że mają one cokolwiek wspólnego z figurami geometrycznymi. W rzeczywistości ten tekst zawiera wyjaśnienie, jak znaleźć *centroid*, czyli środek trójkąta. Jednak tego typu tradycyjne metody nie tłumaczą, *dlatego* powinniśmy być w ogóle zainteresowani szukaniem środka trójkąta.

8. Dany jest graniastosłup prawidłowy sześciokątny o wysokości  $h$  i krawędzi podstawy  $e$ ,  $h = e$ . Z tego graniastosłupa usuwamy dwa przystające ostrosłupy o podstawach przystających do podstaw graniastosłupa. Znajdź wysokość dowolnego z ostrosłupów, jeśli pozostała bryła ma objętości graniastosłupa.

9. **Centroidy.** Przez środek ciężkości bryły rozumiemy ogólnie miejsce, w którym skupia się waga tej bryły. Gdy potraktujemy prostą lub krzywą linię jako bardzo cienką, jednorodny pręt, środek ciężkości pręta będzie nazywany **centroidem** linii. Na przykład centroidem prostego odcinka jest jego środkowy punkt. Analogicznie, jeśli potraktujemy figurę płaską domkniętą jako bardzo cienką jednorodną blachę o jednolitej grubości, jej środek ciężkości będzie nazywany **centroidem** figury płaskiej. Podobnie, **centroid** figury przestrzennej jest definiowany jako identyczny ze środkiem ciężkości tej figury, traktowanej jako bryła z jednorodnego materiału. W związku z tym centroid sfery znajduje się w jej środku.

Korzystając z prawa dźwigni (Plane Geometry, Ex. 3, str. 234), można dość prosto pokazać, że środek bryły złożonej z dwóch części leży na odcinku łączącym środki ciężkości tych części i dzieli odcinek w sposób odwrotnie proporcjonalny do wagi tych części. Można łatwo rozszerzyć ten wynik, aby udowodnić następującą podstawową regułę: *jeśli figura płaska lub przestrzenna jest podzielona na części,*



Rysunek 1. Dwie metody uczenia o centroidzie

Obok tekstu znajduje się rysunek z dynamicznym szkicem mniej więcej stu obracających się trójkątów. Jest to dość trudny projekt programistyczny i aby dokonać obrotu w odpowiedni sposób (i osiągnąć ciekawy efekt wizualny), trzeba znaleźć centroid trójkąta. W wielu sytuacjach tworzenie ciekawych grafik jest prawie niemożliwe bez znajomości odpowiednich dziedzin matematyki np. geometrii. Jak będzie można zobaczyć w tej książce, poznanie pewnych zagadnień matematycznych dotyczących trójkątów, takich jak centroid, ułatwi nam tworzenie własnej grafiki. Uczeń, który zna matematykę i potrafi tworzyć ciekawe projekty, będzie bardziej skłonny nauczyć się nieco geometrii i zaakceptować konieczność obliczenia kilku pierwiastków kwadratowych czy funkcji trygonometrycznych. Uczeń, który nie widzi żadnych korzyści i jedynie odrabia zadanie domowe z podręcznika, prawdopodobnie nie będzie zbyt zmotywowany do nauki geometrii.

Podczas ośmiu lat pracy jako nauczyciel matematyki i trzech jako nauczyciel informatyki spotkałem dużo więcej uczniów, którzy wolą wizualne techniki nauczania matematyki od tych ściśle akademickich. Proces tworzenia czegoś ciekawego pomaga zrozumieć, że matematyka nie sprowadza do opanowywania technik rozwiązywania równań. Poznawanie matematyki przez programowanie daje wiele możliwości rozwiązywania interesujących problemów, z mnóstwem nieprzewidzianych błędów i możliwości ulepszeń.

Na tym polega różnica między szkolną a prawdziwą matematyką.

## Problemy ze szkolną matematyką

Co dokładnie mamy na myśli, używając określenia „szkolna matematyka”? W Stanach Zjednoczonych, w latach 60. XX wieku, szkolna matematyka miała przygotowywać do pracy na stanowisku urzędniczym, czyli do ręcznego dodawania wielu liczb. Obecnie praca wygląda inaczej i w związku z tym trzeba zmienić sposób przygotowywania do niej.

Najlepszą metodą uczenia się jest działanie. Jednak nie znajduje to odzwierciedlenia w codziennym funkcjonowaniu szkół, które mają tendencję do faworyzowania pasywnych metod nauczania. W ramach „pracy” na lekcjach języka ojczystego czy historii uczniowie mogą pisać artykuły lub przygotowywać

prezentację, na naukach przyrodniczych przeprowadzać eksperymenty, ale co z matematyką? Kiedyś jedynymi rodzajami aktywnej „pracy” na lekcjach matematyki było rozwiązywanie równań, rozkładanie wielomianów na czynniki i rysowanie wykresów funkcji. Jednak obecnie komputery mogą wyręczyć nas przy wykonywaniu większości tego typu obliczeń, więc takie ćwiczenia nie wystarczą.

Celem końcowym nie jest nauczenie się, jak automatyzować procesy rozwiązywania równań, rozkładania na czynniki czy rysowania wykresów. Po opanowaniu technik automatyzacji procesu uczeń może pogłębić swoją wiedzę w stopniu, który wcześniej był nieosiągalny.

Rysunek 2 przedstawia typowe, podręcznikowe zadanie matematyczne, który polega na wyznaczeniu wartości zdefiniowanej funkcji  $f(x)$  dla szeregu różnych wartości.

Ćwiczenia 1-22 dotyczą poniższych funkcji. Znajdź określoną wartość funkcji:

$$f(x) = \sqrt{x+3} - x + 1$$
$$g(t) = t^2 - 1$$
$$h(x) = x^2 + \frac{1}{x} + 2$$

1.  $f(0)$
2.  $f(1)$
3.  $f(\sqrt{2})$
4.  $f(\sqrt{2} - 1)$

Rysunek 2. Tradycyjny sposób uczenia o funkcjach

Kolejne 18 pytań ma taką samą postać! Ten rodzaj ćwiczenia stanowi banalne zadanie dla języka programowania takiego jak Python. Wystarczy po prostu zdefiniować funkcję  $f(x)$ , a następnie wstawiać do niej wartości, iterując po liście w następujący sposób:

```
import math

def f(x):
    return math.sqrt(x + 3) - x + 1

# lista wartości do wstawienia
for x in [0,1,math.sqrt(2),math.sqrt(2)-1]:
    print("f({:.3f}) = {:.3f}".format(x,f(x)))
```

Ostatni wiersz służy jedynie do sformatowania wyświetlanych danych i zaokrąglenia wszystkich wyników do trzech cyfr po przecinku<sup>1</sup>, jak pokazano poniżej:

```
f(0.000) = 2.732
f(1.000) = 2.000
f(1.414) = 1.687
f(0.414) = 2.434
```

<sup>1</sup> A dokładniej po kropce, ponieważ w Pythonie, jak w języku angielskim, część dziesiętna jest oddzielana od części całkowitej kropką, a nie przecinkiem (przyp. tłum.).