



Wyznaczanie przybliżonej wartości miejsca zerowego funkcji metodą stycznych w języku Python

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



Wyznaczanie przybliżonej wartości miejsca zerowego funkcji metodą stycznych w języku Python

Źródło: Chris Barbalis, domena publiczna.

Rozwiązując niektóre problemy matematyczne i informatyczne, możemy stosować algorytmy numeryczne wykorzystujące operacje na liczbach. Jednym z takich algorytmów jest metoda stycznych, zwana także metodą Newtona-Raphsona, która służy do wyszukiwania wartości miejsca zerowego funkcji z pewną dokładnością. Zagadnienie to omówiliśmy w e-materiale [Wyznaczanie przybliżonej wartości miejsca zerowego funkcji metodą stycznych](#).

Teraz zajmiemy się implementacją algorytmu wyznaczania przybliżonej wartości miejsca zerowego funkcji metodą stycznych 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:

- [Wyznaczanie przybliżonej wartości miejsca zerowego funkcji metodą stycznych w języku C++](#),
- [Wyznaczanie przybliżonej wartości miejsca zerowego funkcji metodą stycznych w języku Java](#).

Więcej zadań? Sięgnij do [Wyznaczanie przybliżonej wartości miejsca zerowego funkcji metodą stycznych – zadania maturalne](#).

Twoje cele

- Prześledzisz implementację w języku Python algorytmu wyznaczania przybliżonej wartości miejsca zerowego funkcji metodą stycznych.
- Wyjaśnisz, jak dobierać punkt startowy, aby program wykorzystujący metodę stycznych zwrócił wynik.
- Zaimplementujesz program znajdujący przybliżoną wartość miejsca zerowego funkcji metodą stycznych w języku Python.

Przeczytaj

W e-materiale [Wyznaczanie przybliżonej wartości miejsca zerowego funkcji metodą stycznych](#) poznaliśmy algorytm wyszukiwania wartości miejsca zerowego funkcji z pewną dokładnością. Napiszmy zatem program w języku Python, który pozwoli dokonać obliczenia przybliżenia takiego [pierwiastka](#).

Pamiętajmy, że kolejne przybliżenia (pierwiastki) wyznaczone są na podstawie następującego wzoru:

$$x_2 = x_1 - \frac{f(x_1)}{f'(x_1)}$$

gdzie:

- x_1 – punkt startowy; przybliżona wartość miejsca zerowego funkcji $f(x)$ w przedziale $[a, b]$;
- $f(x)$ – przybliżona wartość miejsca zerowego funkcji $f(x)$ w przedziale $[a, b]$; punkt przecięcia stycznej do wykresu funkcji $f(x)$ w punkcie x_1 z osią OX ;
- $f(x_1)$ – wartość funkcji $f(x)$ w punkcie x_1 ;
- $f'(x_1)$ – pochodna funkcji $f(x)$ w punkcie x_1 .

Styczna jest geometryczną interpretacją pochodnej.

Napiszmy program realizujący algorytm wyznaczania miejsca zerowego funkcji $f(x)$ w przedziale $[1, 3]$ z dokładnością do trzech miejsc po przecinku.

$$f(x) = x^3 - 5x$$

Przykład 1

Tworzymy definicję funkcji o nazwie `funkcja`. Jej rolą jest obliczenie i zwrócenie wartości analizowanej funkcji w punkcie `x` zgodnie z wcześniej podanym wzorem.

```
1 def funkcja(x):  
2     return x ** 3 - 5 * x
```

Następnie tworzymy kolejną funkcję obliczającą pochodną funkcji, której miejsce zerowe chcemy wyznaczyć. Pochodna funkcji $f(x)$ ma następujący wzór:

$$f'(x) = 3x^2 - 5$$

W programie nadajemy jej nazwę `pochodna`.

```
1 def pochodna(x):
2     return 3 * x ** 2 - 5
```

Najważniejszą funkcją będzie `metoda_stycznych` o następujących parametrach: punkt startowy x_1 , wartość błędu bezwzględnego ex oraz dokładność ey , czyli maksymalna różnica pomiędzy kolejnymi przybliżeniami, przy której możemy zakończyć algorytm. W trakcie pisania programu będziemy korzystali z funkcji `fabs` umożliwiającej obliczenie wartości bezwzględnej wyrażenia.

```
1 from math import fabs
2 def metoda_stycznych(x1, ex, ey):
3     x2 = x1
4     while True:
5         x1 = x2
6         x2 = x1 - funkcja(x1) / pochodna(x1)
7         if (fabs(x2 - x1) <= ex or funkcja(x2) <= ey):
8             break
9
10    return x2
```

Na początek tworzymy dodatkową zmienną lokalną x_2 , której przypisujemy wartość zmiennej x_1 . W pętli nieskończonej `while True` przypisujemy zmiennej x_1 wartość zmiennej x_2 . W tym momencie wartość zapisana w zmiennej x_2 staje się nowym punktem startowym. Następnie zgodnie z przedstawionym wcześniej wzorem obliczamy miejsce zerowe i zapisujemy do zmiennej x_2 . Jeżeli różnica pomiędzy kolejnymi przybliżeniami (wartość bezwzględna, którą obliczamy funkcją `fabs()`) jest mniejsza lub równa wartości błędu bezwzględnego bądź wartość funkcji dla podanego argumentu jest mniejsza lub równa dokładności, wychodzimy z pętli i na koniec zwracamy wartość zmiennej x_2 , czyli przybliżoną wartość miejsca zerowego funkcji.

W omawianym przykładzie wykonanie takiego programu będzie wyglądać następująco (użyjemy specjalnego formatowania liczb rzeczywistych):

```
1 from math import fabs
```

```

2
3 def funkcja(x):
4     return x ** 3 - 5 * x
5
6
7 def pochodna(x):
8     return 3 * x ** 2 - 5
9
10
11 def metoda_stycznych(x1, ex, ey):
12     x2 = x1
13     while True:
14         x1 = x2
15         x2 = x1 - funkcja(x1) / pochodna(x1)
16         if (fabs(x2 - x1) <= ex or funkcja(x2) <= ey):
17             break
18
19     return x2
20
21
22 przyblizenie = metoda_stycznych(3, 0.0001, 0.0001) * 1000
23 print(f"Wartość pierwiastka wynosi {przyblizenie:.4f}")
24 wynik = przyblizenie / 1000
25 print(f"Wartość pierwiastka wynosi {wynik:.4f}")
26
27 # efekt działania programu
28 # Wartość pierwiastka wynosi 2236.0681
29 # Wartość pierwiastka wynosi 2.2361

```

Dobieranie punktu startowego oraz zbieżność

Ważne!

Zbieżność nie zawsze musi zachodzić. Jeżeli źle dobierzemy punkt startowy, algorytm stycznej nie zwróci wyniku. W takim wypadku program może nigdy się nie skończyć, ponieważ w każdej iteracji będzie się oddalał od znalezienia miejsca zerowego. Taką zbieżność nazywamy **lokalną**. Aby uniknąć możliwej nieskończonej pętli, możemy do funkcji wprowadzić parametr, który ograniczy liczbę prób, np.:

```

1 from math import fabs

```

```

2
3 def funkcja(x):
4     return x ** 3 - 5 * x
5
6
7 def pochodna(x):
8     return 3 * x ** 2 - 5
9
10
11 def metoda_stycznych(x1, ex, ey, max_iter=100):
12     x2 = x1
13     iteracja = 0
14     while True:
15         iteracja += 1
16         if iteracja > max_iter:
17             raise Exception("Zbyt wiele operacji.")
18         x1 = x2
19         x2 = x1 - funkcja(x1) / pochodna(x1)
20         if (fabs(x2 - x1) <= ex or funkcja(x2) <= ey):
21             break
22
23     return x2

```

Efekt wykonania takiego kodu dla dwóch wartości parametru x1:

- `x1 = 999999999999999`
- `x1 = 9999999999999999999999999999999`

```
1 print("Test dla wartości : 9999999999999")  
2 przyblizenie = metoda_stycznych(9999999999999, 0.0001, 0.0001)  
3 print(f"Wartość pierwiastka wynosi {przyblizenie:.4f}")  
4 wynik = przyblizenie / 1000  
5 print(f"Wartość pierwiastka w przybliżeniu 4 cyfr wynosi {wynik  
6 print()  
7 print("Test dla wartości : 9999999999999999999999999")  
8 przyblizenie = metoda_stycznych(9999999999999999999999999, 0.0001  
9 print(f"Wartość pierwiastka wynosi {przyblizenie:.4f}")  
10 wynik = przyblizenie / 1000  
11 print(f"Wartość pierwiastka w przybliżeniu 4 cyfr wynosi {wynik  
12  
13 # Efekt działania takiego wywołania funkcji:
```

[illegible]

O metodzie tej mówimy, że jest **szybko zbieżna**, ponieważ jej zbieżność jest kwadratowa, a nie liniowa; zależy ona jednak od różnych czynników, m.in. od przybliżenia założonego na starcie oraz od postaci funkcyjnej równania.

Słownik

algorytm numeryczny

metoda rozwiązywania problemów matematycznych za pomocą operacji na liczbach

błąd bezwzględny

bezwzględna różnica pomiędzy dokładną wartością a jej uzyskanym przybliżeniem (np. w wyniku obliczeń lub pomiarów)

funkcja wielomianowa

funkcja wyrażana przez wielomian, czyli sumę co najmniej dwóch **jednomianów**
jednomian

iloczyn litery oznaczającej zmienną oraz liczby (nazywanej stałą)

pierwiastek funkcji

inaczej: miejsce zerowe funkcji, czyli argument, dla którego wartość funkcji wynosi zero

miara szybkości, z jaką zmieniają się wartości funkcji; w ujęciu geometrycznym pochodną stanowi styczna do rozpatrywanej funkcji w ustalonym punkcie

styczna

linia prosta mająca jeden punkt wspólny z linią krzywą, np. wartością funkcji w punkcie, nieprzecinająca linii krzywej

wartość bezwzględna

odległość danej liczby od zera na osi liczbowej

zbieżność

zmierzanie do pewnej docelowej wartości granicznej wraz ze zwiększaniem się liczby iteracji

Film samouczek

Polecenie 1

Napisz program wyznaczający przybliżoną wartość miejsca zerowego metodą stycznych (zwaną także algorytmem Newtona-Raphsona). Przetestuj jego działanie dla funkcji $f(x) = x^3 - 2x^2 - 3$.

```
1 # Tutaj dodaj własny kod.
```

```
1
```

Polecenie 2

Porównaj swoje rozwiązanie z przedstawionym w filmie.



Algorytmy numeryczne i przybliżone III

Implementacja algorytmu Newtona-Raphsona, założenia i zbieżność metody w języku Python



Film dostępny pod adresem </preview/resource/RdkiwJuiieDV4>

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

Film nawiązujący do algorytmów numerycznych i przybliżonych.

Plik o rozmiarze 265.00 B w języku polskim

Polecenie 3

Porównaj implementację zaprezentowaną w filmie z tą z sekcji „Przeczytaj”. Omów, co je różni.

Sprawdź się

Pokaż ćwiczenia:   

Ćwiczenie 1



Mamy daną funkcję $f(x)$ oraz jej pochodną $f'(x)$. Wiemy, że funkcja $f(x)$ ma jedno miejsce zerowe znajdujące się w przedziale (x_1, x_2) .

Napisz program, który za pomocą metody stycznych wyznaczy przybliżoną wartość miejsca zerowego funkcji $f(x)$ w danym przedziale i wypisze ją z dokładnością do trzeciego miejsca po przecinku (nie zaokrąglaj ostatniej cyfry).

Przetestuj swój program dla następujących danych:

- $f(x) = x \sin(x) + x^3 - 5$
- $f'(x) = x \cos(x) + \sin(x) + 3x^2$
- $x_1 = 1$
- $x_2 = 2$

Specyfikacja problemu:

Dane:

- funkcja – funkcja f , której przybliżoną wartość miejsca zerowego należy obliczyć
- pochodna – funkcja będąca pochodną funkcji f
- x_1 – początek przedziału; liczba rzeczywista
- x_2 – koniec przedziału; liczba rzeczywista

Wynik:

- wynik – przybliżona wartość miejsca zerowego funkcji; liczba rzeczywista

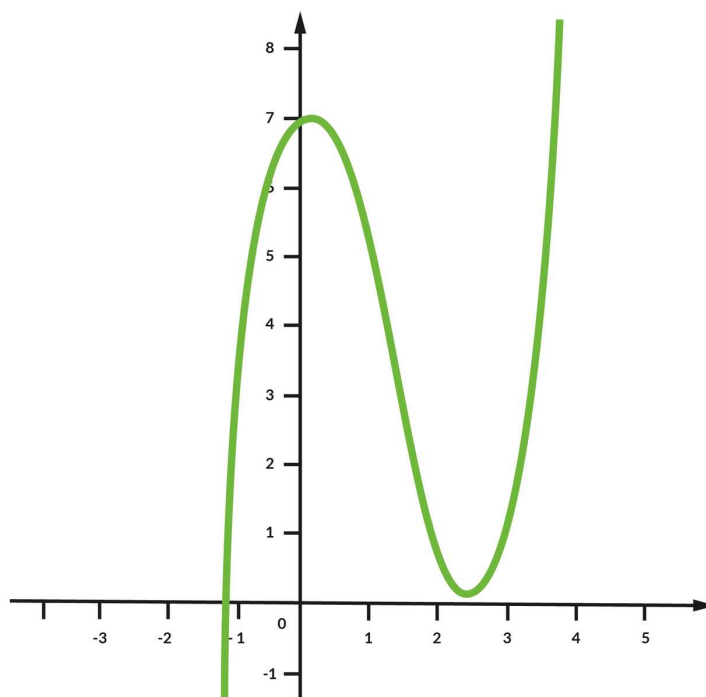
Twoje zadania

1. Program oblicza przybliżoną wartość miejsca zerowego funkcji $f(x)$ w danym przedziale (x_1, x_2) i wypisuje ją na standardowe wyjście z dokładnością do trzech miejsc po przecinku, bez zaokrąglania.

Ćwiczenie 2



Zapoznaj się z wykresem funkcji $f(x) = x^3 - 4x^2 + x + 7$ i wykonaj polecenie.



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

Zastanów się, dlaczego w przedstawionym programie odnalezienie miejsca zerowego zajmuje tak wiele iteracji. Zmodyfikuj tylko parametr x_1 w taki sposób, aby program wypisał „Sukces” oraz w następnej linii wynik będący przybliżoną wartością z dokładnością do drugiego miejsca po przecinku (nie zaokrąglaj wyniku).

Specyfikacja problemu:

Dane:

- f – funkcja, której miejsce zerowe należy obliczyć
- f' – pochodna funkcji f
- ϵ – dokładność rozwiązania; liczba rzeczywista
- x_1 – wstępne przybliżenie miejsca zerowego; liczba rzeczywista

Wynik:

- program wypisuje na standardowe wyjście napis „Sukces” jeśli liczba iteracji jest mniejsza niż 10 lub „Liczba iteracji” oraz liczbę iteracji; w kolejnej linii powinien zostać wypisany wynik będący przybliżoną wartością, z dokładnością do drugiego miejsca po przecinku (bez zaokrąglania)

Przykładowe wyjście:

```
1 Sukces
2 -1.07
```

Twoje zadania

1. Program metodą stycznych oblicza przybliżoną wartość miejsca zerowego funkcji $f(x) = x^2 - 2x + 1$ w mniej niż 10 iteracji.



Ćwiczenie 3



Napisz program, który za pomocą metody stycznych obliczy rozwinięcie dziesiętne liczby π z dokładnością do piątego miejsca po przecinku. Wykorzystaj odpowiednie funkcje z biblioteki `math`.

Pamiętaj, że $\sin(\pi) = 0$.

Rozwiązanie wypisz z dokładnością do pięciu miejsc po przecinku, bez zaokrąglania.

Specyfikacja problemu:

Dane:

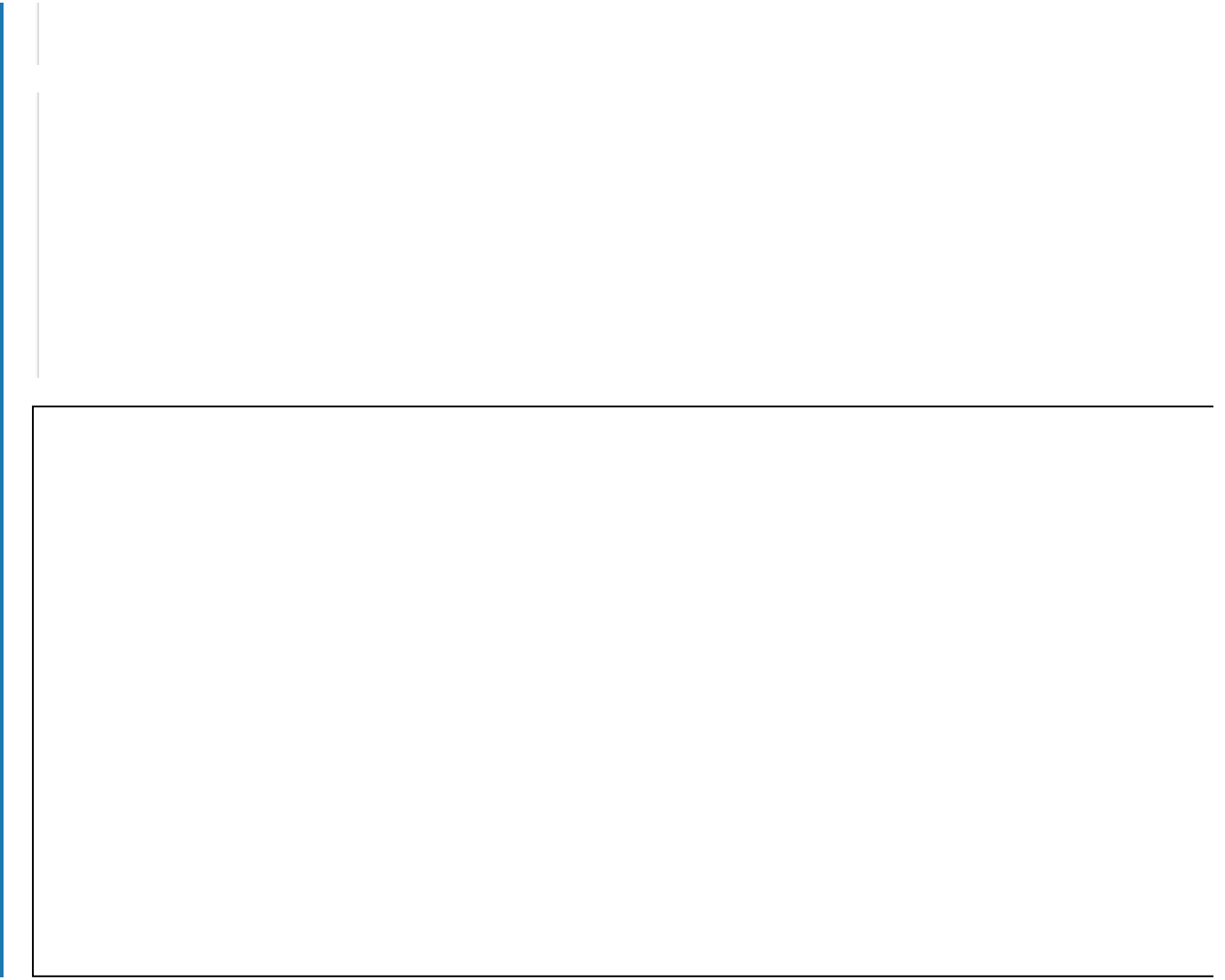
- π – liczba niewymierna

Wynik:

- `wynik` – rozwinięcie dziesiętne liczby π z dokładnością do pięciu miejsc po przecinku; liczba rzeczywista

Twoje zadania

1. Program oblicza przybliżone do piątego miejsca po przecinku (bez zaokrąglania) rozwinięcie liczby π .



Dla nauczyciela

Autor: Adam Jurkiewicz

Przedmiot: Informatyka

Temat: Wyznaczanie przybliżonej wartości miejsca zerowego funkcji metodą stycznych 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

I. Rozumienie, analizowanie i rozwiązywanie problemów.

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

1) w zależności od problemu rozwiązuje go, stosując metodę wstępującą lub zstępującą;

3) objaśnia dobrany algorytm, uzasadnia poprawność rozwiązania na wybranych przykładach danych i ocenia jego efektywność;

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

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

1) projektuje i tworzy rozbudowane programy w procesie rozwiązywania problemów, wykorzystuje w programach dobrane do algorytmów struktury danych, w tym

struktury dynamiczne i korzysta z dostępnych bibliotek dla tych struktur;

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

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 implementację w języku Python algorytmu wyznaczania przybliżonej wartości miejsca zerowego funkcji metodą stycznych.
- Wyjaśnisz, jak dobierać punkt startowy, aby program wykorzystujący metodę stycznych zwrócił wynik.
- Zaimplementujesz program znajdujący przybliżoną wartość miejsca zerowego funkcji metodą stycznych w języku Python.

Strategie nauczania:

- konstruktywizm;
- konektywizm.

Metody i techniki nauczania:

- dyskusja;
- rozmowa nauczająca z wykorzystaniem multimediu i ćwiczeń interaktywnych;
- ćwiczenia praktyczne;
- burza mózgów;
- mapa myśli.

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: „Wyznaczanie przybliżonej wartości miejsca zerowego funkcji metodą stycznych w języku Python”. Uczniowie zapoznają się z treściami w sekcji „Przeczytaj” w kontekście programowania.

Faza wstępna:

1. Nauczyciel wyświetla temat i cele zajęć zawarte w sekcji „Wprowadzenie”. Następnie wspólnie z uczniami ustala kryteria sukcesu.
2. Uczniowie metodą burzy mózgów przypominają najważniejsze informacje o algorytmie wyznaczania przybliżonej wartości miejsca zerowego funkcji metodą stycznych. Wybrana osoba zapisuje je na tablicy w formie mapy myśli.

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”.
2. **Praca z multimediami.** Uczniowie wykonują polecenie 1 z sekcji „Film samouczek”. Chętna lub wybrana osoba prezentuje klasie swój kod. Nauczyciel omawia go, uczniowie dyskutują na jego temat. Następnie uczniowie w parach porównują swoje kody z filmem. Dodają do nich komentarze.
3. **Ćwiczenie umiejętności.** Uczniowie rozwiązują ćwiczenia 1-3 z sekcji „Sprawdź się”.

Faza podsumowująca:

1. Nauczyciel omawia rozwiązanie ćwiczeń z sekcji „Sprawdź się”.
2. Nauczyciel wyświetla na tablicy temat lekcji i cele zawarte w sekcji „Wprowadzenie”. W kontekście ich realizacji podsumowuje przebieg zajęć, a także wskazuje mocne i słabe strony pracy uczniów.

Praca domowa:

1. Uczniowie wykonują polecenie 3 z sekcji „Film samouczek”.

Materiały pomocnicze:

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

Wskazówki metodyczne:

- Uczniowie mogą wykorzystać multimedium w sekcji „Film samouczek” do przygotowania się do lekcji powtórkowej.