Obliczanie przybliżonej wielkości pola obszarów zamkniętych w języku Python

Źródło: Markus Spiske, domena publiczna.

W e-materiale Obliczanie przybliżonej wielkości pola obszarów zamkniętych dowiedzieliśmy się, jak obliczyć pole powierzchni pod wykresem funkcji.

W tym e-materiale zaimplementujemy w języku Python metodę trapezów do obliczania pola obszaru ograniczonego wykresem funkcji.

Implementacje w pozostałych językach programowania znajdziesz w e-materiałach:

- Obliczanie przybliżonej wielkości pola obszarów zamkniętych w języku C++,
- Obliczanie przybliżonej wielkości pola obszarów zamkniętych w języku Java.

Więcej zadań? Przejdź do e-materiału Obliczanie przybliżonej wielkości pola obszarów zamkniętych – zadania maturalne.

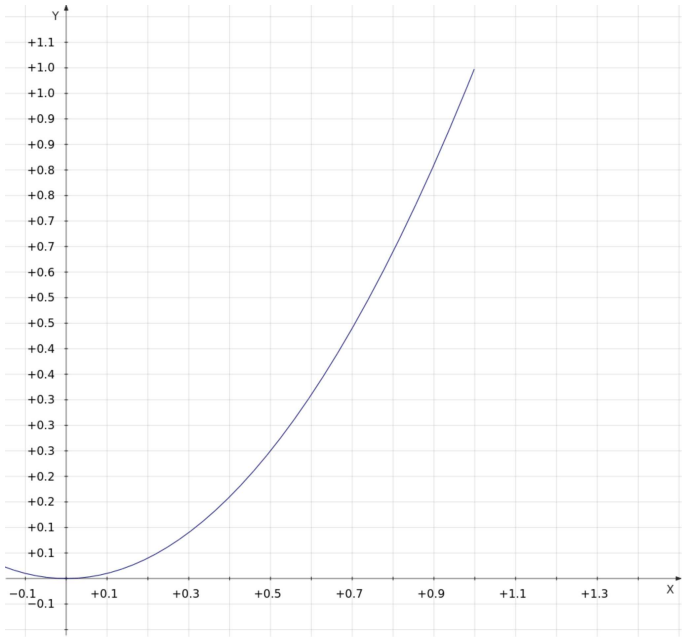
> **Twoje cele**
>
> - Prześledzisz i powtórzysz wiadomości na temat działania funkcji `eval()`.
> - Przygotujesz funkcje pomocne podczas obliczenia przybliżonej wielkości pola obszarów zamkniętych.
> - Zaimplementujesz w języku Python algorytm obliczający pole powierzchni ograniczonej wykresem funkcji.

Prezentacja multimedialna

Polecenie 1

Przeanalizuj krok po kroku obliczanie przybliżonej wielkości pola obszarów zamkniętych z wykorzystaniem metody trapezów.

1



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

Materiał audio dostępny pod adresem:

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

Dana jest funkcja:

$f\left(x\right) = x^2$

Wykonamy obliczenia metodą trapezów w przedziale od `a = 0.1` do `b = 0.6`, podzielonym na pięć odcinków.

2

Materiał audio dostępny pod adresem:

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

Wyznaczamy `dx` oraz przyjmujemy kolejne elementy dla przedziału. Inicjujemy zmienną `wynik`, do której będziemy dodawać wyniki częściowe.

```

```

3

Materiał audio dostępny pod adresem:

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

Obliczamy wartość pierwszego elementu `x (elem)` z przedziału:

```

```

4

Materiał audio dostępny pod adresem:

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

Obliczamy f(x) dla tego elementu:

```

```

5

Materiał audio dostępny pod adresem:

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

Obliczamy wartość drugiego elementu `x (elem + dx)`:

```

```

6

Materiał audio dostępny pod adresem:

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

Obliczamy f(x) dla tego elementu:

7

Materiał audio dostępny pod adresem:

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

Korzystając ze wzoru na powierzchnię trapezu, obliczamy częściowy wynik i dodajemy wartość do zmiennej `wynik`:

8

Materiał audio dostępny pod adresem:

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

Przedstawione kroki powtarzamy jeszcze cztery razy, uzyskując częściowe wyniki:

9

Materiał audio dostępny pod adresem:

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

Sumujemy wszystkie wyniki cząstkowe i uzyskujemy wynik:

10

Materiał audio dostępny pod adresem:

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

Funkcja zwraca wynik. Ten liczony „na kartce" nie jest identyczny z komputerowym (wynika to ze sposobu, w jaki komputer przechowuje w pamięci operacyjnej liczby rzeczywiste).

Polecenie 2

Opracuj notatkę podsumowującą najważniejsze informacje przedstawione w prezentacji.

Przeczytaj

> **Definicja: Całkowanie numeryczne**
>
> Całkowanie numeryczne to przybliżenie całki, poprzez sumy ważone wartości całkowanej funkcji w wielu punktach. Przedział całkowania należy podzielić na niewielkie fragmenty, a wynikiem całkowania jest suma oszacowań całek w tych obszarach.

Znamy różne sposoby obliczania wartości całki. Są to m.in.:

- metoda prostokątów (omówiona w innym e-materiale z tej serii),
- metoda trapezów.

Podczas stosowania metody trapezów, będziemy korzystać ze wzoru na pole trapezu:

$$P = \frac{(podstawa_1 + podstawa_2) \cdot h}{2}$$

Wysokość h obliczamy jako:

$$h = \frac{b - a}{i}$$

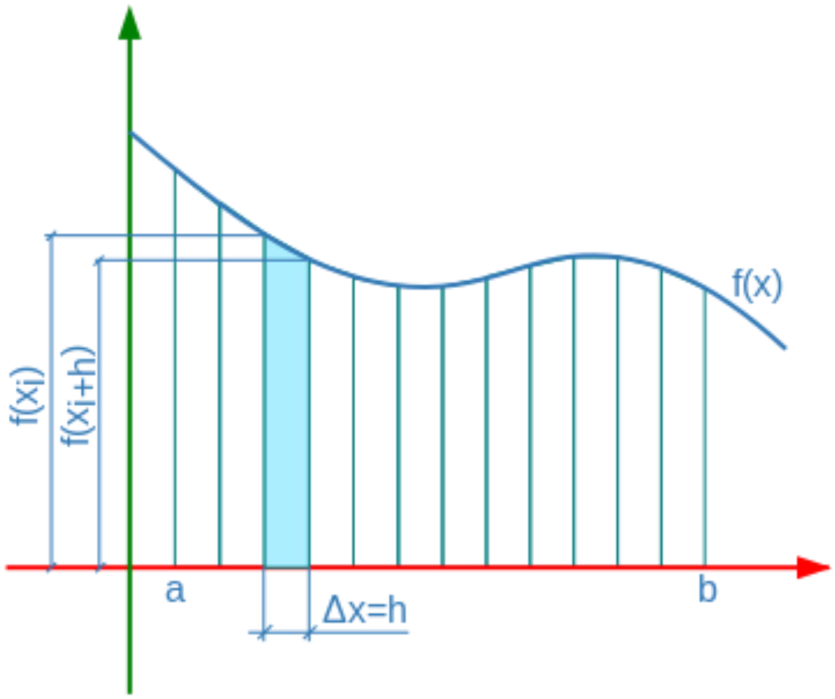
- a – początek przedziału całkowania,
- b – koniec przedziału całkowania,
- i – liczba podprzedziałów, w których będziemy liczyć pole trapezu (im większe i, tym większa dokładność algorytmu, ponieważ będziemy liczyć pola większej liczby trapezów na mniejszych przedziałach).

Długość pierwszej podstawy obliczamy jako:

$$podstawa_1 = f\left(x_i\right)$$

Długość drugiej podstawy obliczamy jako:

$$podstawa_2 = f\left(x_{i+1}\right)$$



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

Przykład 1

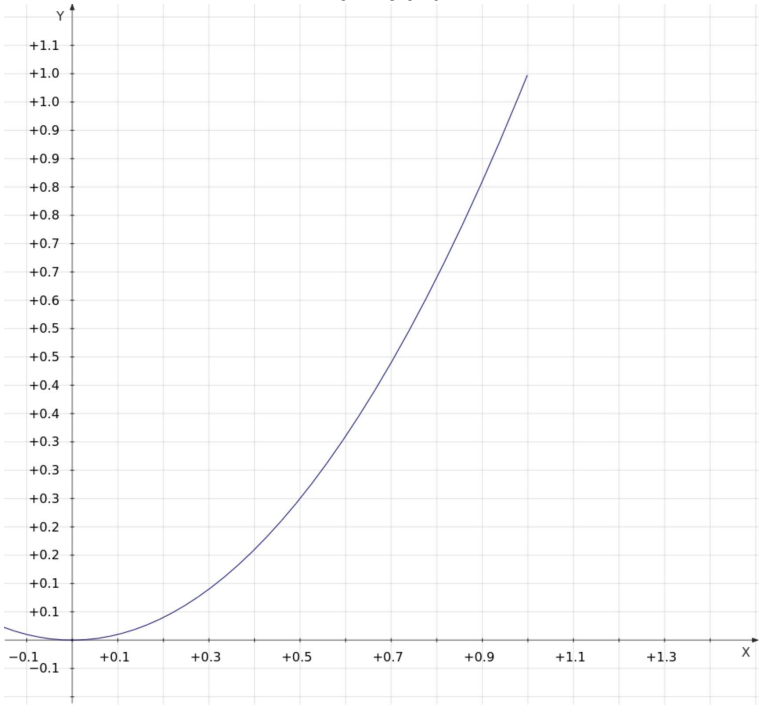
Spróbujmy stworzyć program, który może posłużyć do obliczania całki funkcji o podanym przepisie. W tym przypadku niech to będzie funkcja kwadratowa `f(x) = x`^{2}.

```
def calka_trapez(a, b, i):

    def f(x):
        return x*x # tu możemy zdefiniować funkcję, której całk

    dx = (b - a) / i
    calka = 0
    for elem in range(i):
        elem = elem * dx + a
        fx1 = f(elem)
        fx2 = f(elem+dx)
        calka += 0.5 * dx * (fx1 + fx2)
    return calka
```

Dla przykładowej funkcji:

$$f\left(x\right) = x^2 \text{ dla } x \in \left(0, 1\right) \text{ oraz dla } 10 \text{ przedziałów}$$

jej wykres możemy zaprezentować w następujący sposób:



Teraz możemy wywołać funkcję dla określonego przedziału liczb `(a, b)` i wyświetlić wynik:

```
wynik = calka_trapez(0, 1, 10)
print(wynik)

# 0.3350000000000001
# dokładny wynik to 1/3, więc uzyskaliśmy dość dobre przybliżen
```

Przykład 2

Możemy wykorzystać wbudowaną funkcję `eval()` do stworzenia funkcji, która jako parametr będzie przyjmować kod funkcji do wykonania w momencie obliczania. Nazwa zmiennej w tym kodzie musi być identyczna z użytą do enumeracji w pętli `for`. Przygotujmy zatem kod, który zrealizuje taki sposób obliczania.

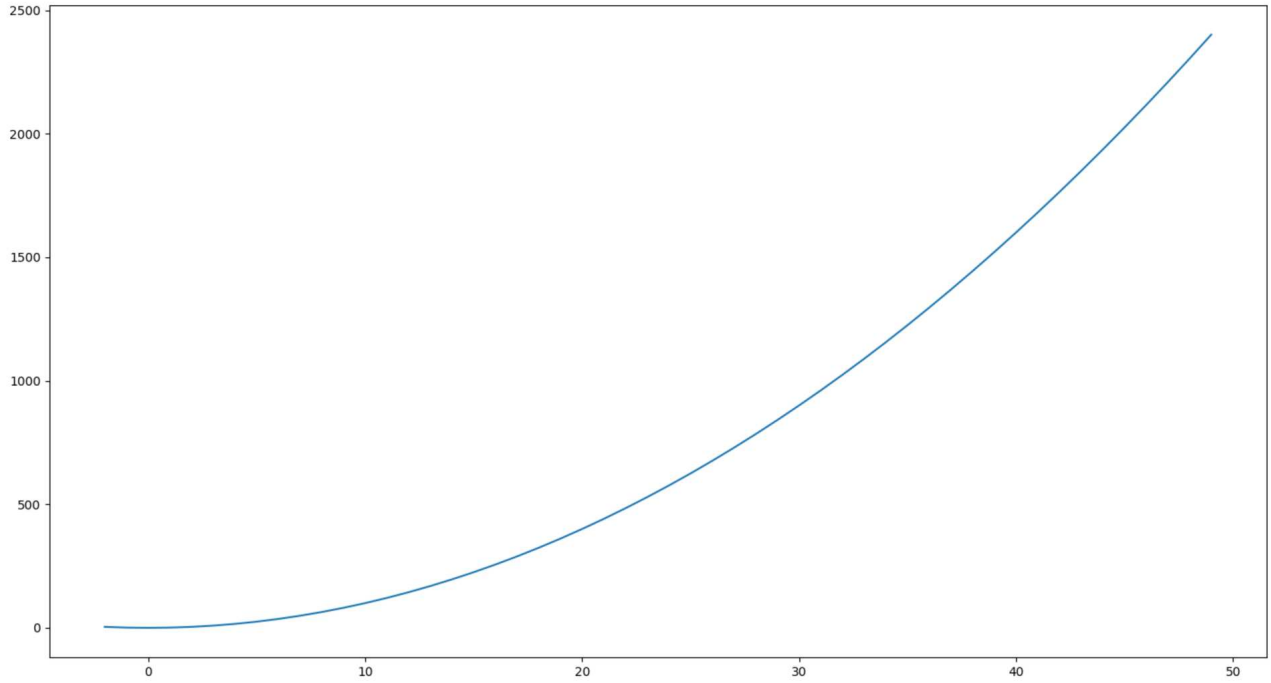
```
def calka_trapez_eval(fun_txt, a, b, i):
    dx = (b - a) / i
    calka = 0
    for elem in range(i):
        elem = elem * dx + a
        fx1 = eval(fun_txt)
```

```
        elem += dx
        fx2 = eval(fun_txt)
        calka += 0.5 * dx * (fx1 + fx2)
    return calka

# przykład wywołania:
wynik = calka_trapez_eval('elem**2', 0, 1, 100)
print(wynik)

# 0.33335
```

W języku Python istnieją możliwości wygenerowania wykresu funkcji i zaprezentowania go na ekranie. Oto przykład z wykorzystaniem biblioteki `matplotlib`:

```
import matplotlib.pyplot as plt

def wykres_funkcji(a, b, i):
    """
    a - punkt początkowy
    b - punkt końcowy
    i - krok
    """

    def f(x):
        return x**2

    X = [ x for x in range(a,b,i) ]
    Y = [ f(x) for x in range(a,b,i) ]
    plt.plot(X, Y)
    plt.show()

wykres_funkcji(-2, 50, 1)
```

Wynikiem działania takiego programu będzie wykres:



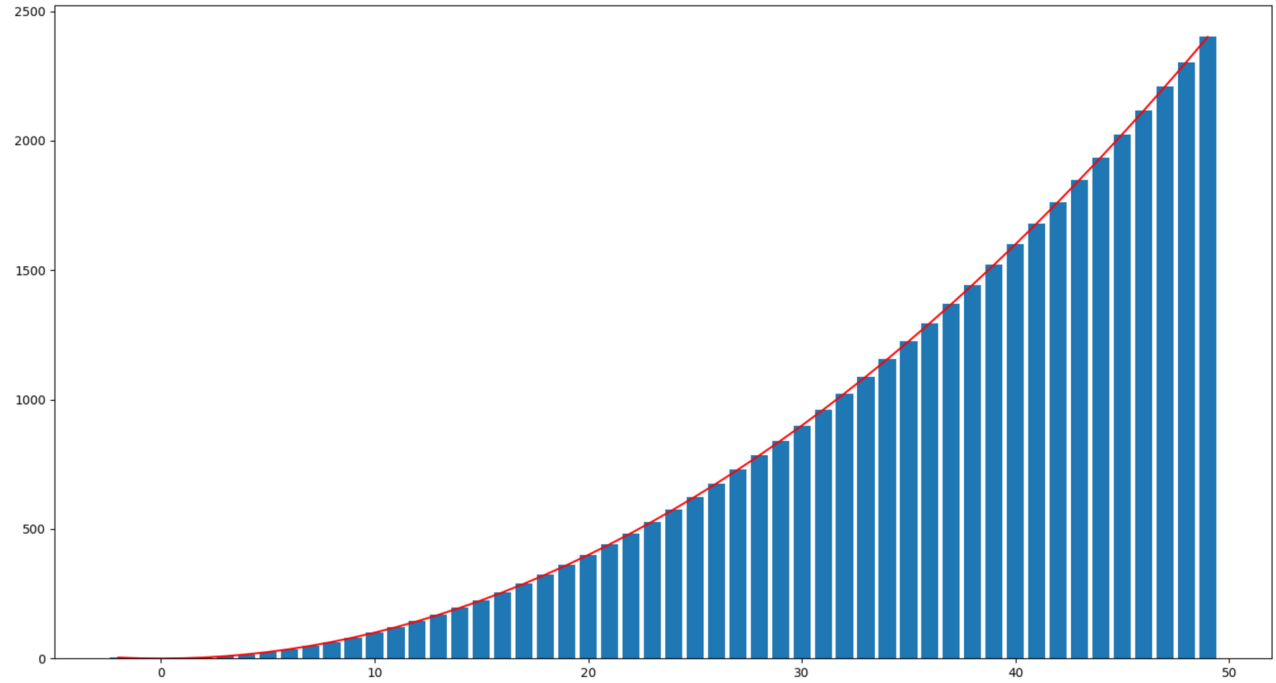
Źródło: Contentplus.pl Sp. z o.o., licencja: CC BY-SA 3.0.

Moduł `matplotlib` możemy również wykorzystać w celu nałożenia różnych typów wykresów. Tym samym wyświetlimy jednocześnie wykres słupkowy (otrzymany za pomocą metody `bar()`) oraz wykres liniowy (otrzymany za pomocą metody `plot()`). Używając wykresu słupkowego, możemy zobaczyć, na jakich przedziałach nasza funkcja liczyła pola trapezów. Oto zdefiniowana funkcja:

```
1  import matplotlib.pyplot as plt
2
3  def wykres_funkcji(a, b, i):
4      """
5      a - punkt początkowy
6      b - punkt końcowy
7      i - krok
8      """
9
10     def f(x):
11         return x**2
12
13     X = [ x for x in range(a,b,i) ]
14     Y = [ f(x) for x in range(a,b,i) ]
15
16     # bar - wykres słupkowy
17     plt.bar(X, Y)
18
19     # plot - wykres liniowy
```

```
    plt.plot(X, Y, 'r-')

    plt.show()

wykres_funkcji(-2, 50, 1)
```

Wynikiem działania takiego programu będzie wykres:



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

Już wiesz

- Wartość całki możemy obliczyć, wykorzystując metodę trapezów.
- W celu obliczenia wartości różnych funkcji, możemy użyć funkcji `eval()`.

Słownik

całka

uogólnione pojęcie sumy; można ją sobie wyobrazić jako sumę nieskończonej liczby nieskończenie małych wartości funkcji `f(x)`

eval()

wbudowana funkcja w języku Python; może służyć do dynamicznego wyliczania wartości wyrażenia w programie

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 języka Python – należy go zainstalować, korzystając z mechanizmu `pip`

Sprawdź się

Pokaż ćwiczenia:

Ćwiczenie 1

Zaznacz wszystkie poprawne stwierdzenia.

- [ ] Możemy polepszyć dokładność całkowania, zmniejszając liczbę podprzedziałów.
- [ ] Funkcja `eval()` jest wbudowaną funkcją języka Python.
- [ ] Możemy polepszyć dokładność całkowania, zwiększając liczbę podprzedziałów.
- [ ] Funkcja `eval()` nie jest wbudowaną funkcją języka Python.

Ćwiczenie 2

Zdefiniuj funkcję `pole(f1, f2, a, b, i)`, która obliczy pole powierzchni zawartej między krzywymi będącymi wykresami funkcji `f1` i `f2` w przedziale $\langle a,\ b\rangle$. Skorzystaj z metody trapezów na `i` podprzedziałach i funkcji `eval()`. Wynik zaokrąglij do siedmiu miejsc po przecinku.

Swoje rozwiązanie przetestuj dla podanych funkcji na przedziale $\langle 0,\ 1\rangle$, podzielonym na 200 trapezów.

$$f1\left(x\right) = x^4 + x^3 + \frac{x^2}{25}$$

$$f2\left(x\right) = x^4 + x^2 + \frac{x}{4}$$

Specyfikacja problemu:

Dane:

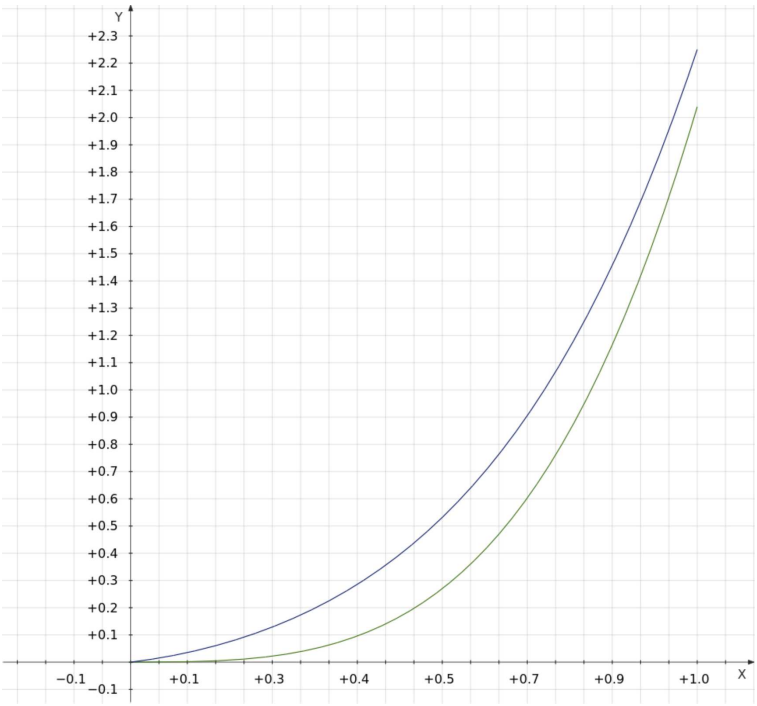
- `f1, f2` – ciąg znaków przedstawiający funkcję ze zmienną x
- `a, b` – liczby całkowite wyznaczające przedział całkowania
- `i` – liczba całkowita; liczba podprzedziałów

Wynik:

- `wynik` – liczba zmiennoprzecinkowa; wynik całkowania

Obraz poglądowy:

- $f1(x)$ – wykres zielony
- $f2(x)$ – wykres niebieski

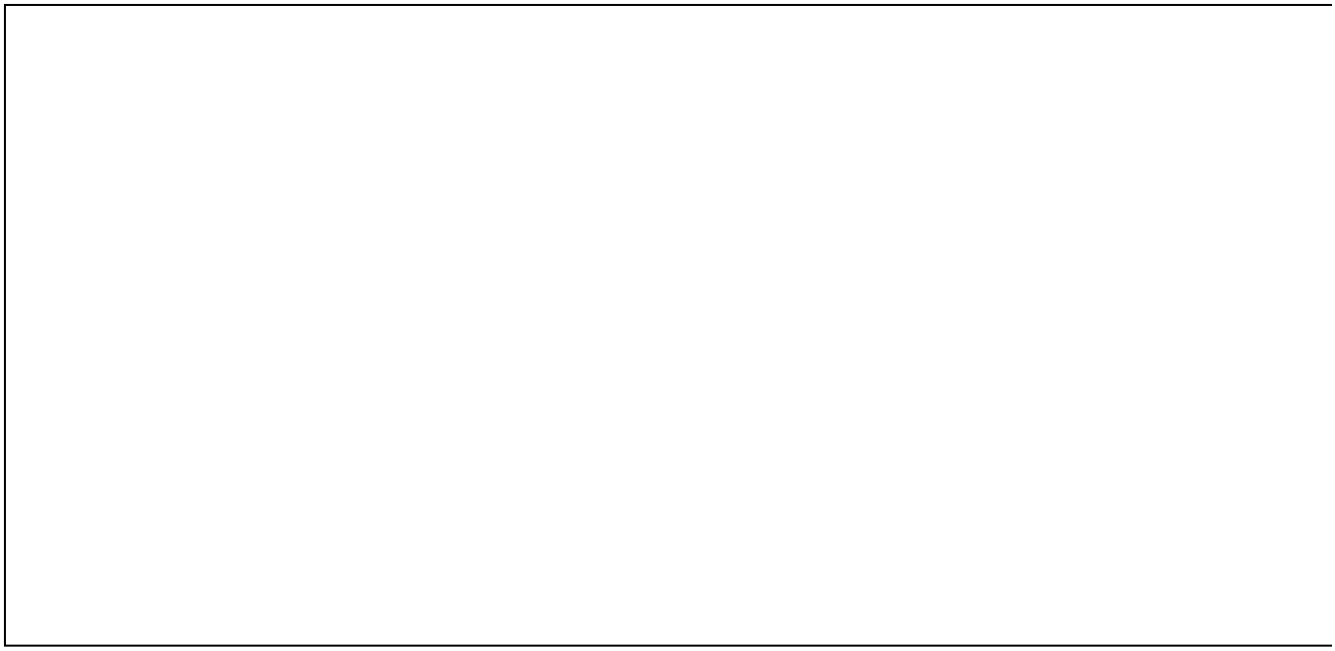


Twoje zadania

1. Program, za pomocą metody trapezów, oblicza pole obszaru pomiędzy wykresami funkcji `f1` i `f2`.

```
def pole(f1, f2, a, b, i):
    # tutaj Twój kod
    return None

wynik = pole('x**4+x**3+x**2/25','x**4+x**2+x/4',0,1,200)
print(wynik)
```

```
1
```



Dla nauczyciela

Autor: Adam Jurkiewicz
Przedmiot: Informatyka

Temat: Obliczanie przybliżonej wielkości pola obszarów zamkniętych 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:

2) do realizacji rozwiązania problemu dobiera odpowiednią metodę lub technikę algorytmiczną i struktury danych;

4) ilustruje i wyjaśnia rolę pojęć, obiektów i operacji matematycznych w projektowaniu rozwiązań problemów informatycznych i z innych dziedzin, posługuje się pojęciem logarytmu;

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:

2) wykorzystuje znane sobie algorytmy przy rozwiązywaniu i programowaniu rozwiązań następujących problemów:

f) obliczanie przybliżonej wielkości pola obszarów zamkniętych;

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 i powtórzysz wiadomości na temat działania funkcji `eval()`.
- Przygotujesz funkcje pomocne podczas obliczenia przybliżonej wielkości pola obszarów zamkniętych.
- Zaimplementujesz w języku Python algorytm obliczający pole powierzchni ograniczonej wykresem funkcji.

Strategie nauczania:

- konstruktywizm;
- konektywizm.

Metody i techniki nauczania:

- dyskusja;
- rozmowa nauczająca z wykorzystaniem multimedium 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.

Przebieg lekcji

Przed lekcją:

1. **Przygotowanie do zajęć.** Nauczyciel loguje się na platformie i udostępnia e-materiał: „Obliczanie przybliżonej wielkości pola obszarów zamkniętych w języku Python". Nauczyciel prosi uczniów o zapoznanie się z multimedium w sekcji „Przeczytaj".

Faza wstępna:

1. Chętna lub wybrana osoba przypomina najważniejsze informacje dotyczące obliczania przybliżonej wielkości pola obszarów zamkniętych.
2. Nauczyciel wyświetla uczniom temat, wskazuje cele zajęć oraz ustala z uczestnikami zajęć kryteria sukcesu.

Faza realizacyjna:

1. Odnosząc się do treści zawartej w sekcji „Prezentacja multimedialna", uczniowie w parach konstruują alternatywny przykład, definiując samodzielnie problem, rozwiązanie i ewentualną implementację. Rezultaty omawiane są na forum klasy.
2. **Praca z multimedium.** Nauczyciel wyświetla zawartość sekcji „Przeczytaj", czyta treść polecenia nr 1 „Możemy również wykorzystać wbudowaną funkcję `eval()` do stworzenia funkcji, która jako parametr będzie przyjmować kod funkcji do wykonania w momencie obliczania. Musimy pamiętać o tym, aby nazwa zmiennej w tym kodzie była identyczna z użytą do enumeracji w pętli `for`." i omawia kolejne kroki rozwiązania.
3. **Ćwiczenie umiejętności.** Uczniowie wykonują ćwiczenia nr z sekcji „Sprawdź się". Nauczyciel sprawdza poprawność wykonanych zadań, omawiając je wraz z uczniami.
4. Praca indywidualna – implementacja poznanej techniki do rozwiązywania problemów informatycznych – wykonywanie ćwiczeń z sekcji „Sprawdź się".

Faza podsumowująca:

1. Wybrany uczeń podsumowuje zajęcia, zwracając uwagę na nabyte umiejętności.

Praca domowa:

1. Uczniowie proponują alternatywny sposób rozwiązania problemu postawionego w sekcji „Przeczytaj".

Materiały pomocnicze:

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

Wskazówki metodyczne:

- Multimedium w sekcji „Przeczytaj" można potraktować jako zadanie domowe dotyczące analizy problemu zawartego w temacie „Obliczanie przybliżonej wielkości

pola obszarów zamkniętych w języku Python”.