

GEMINI

```
def kalkulator():
```

```
    """Funkcja definiująca działanie kalkulatora."""
```

```
    while True:
```

```
        # Pobranie działania od użytkownika
```

```
        dzialanie = input("Wybierz działanie (+, -, , /, ^, pierwiastek kwadratowy, pierwiastek sześcienny, logarytm): ")
```

```
        # Sprawdzenie poprawności działania
```

```
        if dzialanie not in "+-/*^":
```

```
            print("Niepoprawne działanie. Spróbuj ponownie.")
```

```
            continue
```

```
        # Pobranie liczb od użytkownika
```

```
        liczba1 = float(input("Podaj pierwszą liczbę: "))
```

```
        liczba2 = float(input("Podaj drugą liczbę: "))
```

```
        # Wykonanie obliczeń
```

```
        if dzialanie == "+":
```

```
            wynik = liczba1 + liczba2
```

```
        elif dzialanie == "-":
```

```
            wynik = liczba1 - liczba2
```

```
        elif dzialanie == "*":
```

```
            wynik = liczba1 * liczba2
```

```
        elif dzialanie == "/":
```

```
            if liczba2 == 0:
```

```
                print("Dzielenie przez zero jest niedozwolone!")
```

```

        continue

    else:

        wynik = liczba1 / liczba2

    elif dzialanie == "^":

        wynik = liczba1 ** liczba2

    elif dzialanie == "s":

        wynik = liczba1 ** (1/2)

    elif dzialanie == "l":

        wynik = math.log(liczba1, liczba2)

    else:

        wynik = math.pow(liczba1, (1/3))


# Wyświetlenie wyniku

print(f"{liczba1} {dzialanie} {liczba2} = {wynik}")


# Zapytanie o kontynuację

kontynuacja = input("Czy chcesz wykonać kolejne działanie? (t/n): ")

if kontynuacja.lower() != "t":

    break

```

Uruchamianie kalkulatora

kalkulator()

GPT

import math

def dodawanie(a, b):

return a + b

```
def odejmowanie(a, b):
```

```
    return a - b
```

```
def mnozenie(a, b):
```

```
    return a * b
```

```
def dzielenie(a, b):
```

```
    if b != 0:
```

```
        return a / b
```

```
    else:
```

```
        return "Błąd: Nie można dzielić przez zero!"
```

```
def potegowanie(a, b):
```

```
    return a ** b
```

```
def pierwiastek_kwadratowy(a):
```

```
    return math.sqrt(a)
```

```
def logarytm(a, b):
```

```
    if a > 0 and b > 0:
```

```
        return math.log(a, b)
```

```
    else:
```

```
        return "Błąd: Podstawa logarytmu i argument muszą być większe od zera!"
```

```
print("Prosty kalkulator naukowy")
```

```
print("Wybierz działanie:")
```

```
print("1. Dodawanie")
```

```
print("2. Odejmowanie")
```

```
print("3. Mnożenie")
```

```
print("4. Dzielenie")
```

```
print("5. Potęgowanie")
```

```
print("6. Pierwiastek kwadratowy")
```

```
print("7. Logarytm")
```

```
wybor = input("Twój wybór (wpisz odpowiedni numer): ")
```

```
if wybor in ['1', '2', '3', '4', '5', '7']:
```

```
    liczba1 = float(input("Podaj pierwszą liczbę: "))
```

```
    liczba2 = float(input("Podaj drugą liczbę: "))
```

```
    if wybor == '1':
```

```
        print("Wynik:", dodawanie(liczba1, liczba2))
```

```
    elif wybor == '2':
```

```
        print("Wynik:", odejmowanie(liczba1, liczba2))
```

```
    elif wybor == '3':
```

```
        print("Wynik:", mnozenie(liczba1, liczba2))
```

```
    elif wybor == '4':
```

```
        print("Wynik:", dzielenie(liczba1, liczba2))
```

```
    elif wybor == '5':
```

```
        print("Wynik:", potegowanie(liczba1, liczba2))
```

```
    elif wybor == '7':
```

```
        print("Wynik:", logarytm(liczba1, liczba2))
```

```
else:
```

```
    if wybor == '6':
```

```
        liczba = float(input("Podaj liczbę: "))
```

```
        print("Wynik:", pierwiastek_kwadratowy(liczba))
```

else:

```
print("Błąd: Niepoprawny wybór działania!")
```

Porównanie i analiza dwóch kodów:

Oba zaprezentowane kody mają na celu implementację prostego kalkulatora naukowego w Pythonie, jednak różnią się w podejściu do struktury, interakcji z użytkownikiem i obsługi błędów. Oto kluczowe różnice i podobieństwa między nimi oraz sugestie dotyczące usprawnień.

Kluczowe różnice

Interfejs użytkownika:

Pierwszy kod (GEMINI): Użytkownik wprowadza operacje jako tekst (np. "+", "-", "pierwiastek kwadratowy").

Drugi kod (GPT): Użytkownik wybiera operacje z menu, wpisując odpowiedni numer.

Struktura i modularność:

Pierwszy kod: Monolityczna funkcja kalkulator, w której wszystkie operacje i interakcje są zdefiniowane.

Drugi kod: Modularna struktura z osobnymi funkcjami dla każdej operacji matematycznej.

Obsługa błędów:

Pierwszy kod: Podstawowa walidacja działań i obsługa dzielenia przez zero.

Drugi kod: Obsługuje dzielenie przez zero oraz waliduje argumenty logarytmu.

Interakcja wielokrotna:

Pierwszy kod: Umożliwia użytkownikowi wykonywanie wielu działań w ramach jednej sesji kalkulatora.

Drugi kod: Pierwotnie umożliwia wykonanie tylko jednej operacji, po czym program się kończy.