



Ciąg Fibonacciego w języku Python

- [Wprowadzenie](#)
- [Schemat interaktywny](#)
- [Przeczytaj](#)
- [Sprawdź się](#)
- [Dla nauczyciela](#)



Ciąg Fibonacciego w języku Python

Źródło: Jason Leung, domena publiczna.

Ciąg Fibonacciego to ciąg liczb naturalnych opisany w sposób rekurencyjny: pierwszy element jest równy 0, drugi równy jest 1, a każdy następny jest sumą dwóch poprzednich. Ciąg ten po raz pierwszy został omówiony w 1202 roku przez Leonarda z Pizy, zwanego również Fibonaccim. Więcej informacji znajdziesz w e-materiale [Ciąg Fibonacciego](#).

Z tego e-materiału dowiesz się, jak w języku Python zaimplementować algorytm służący do obliczania wyrazów ciągu Fibonacciego.

Implementację algorytmów wykorzystujących ciąg Fibonacciego w wybranych językach programowania znajdziesz w e-materiałach:

- [Ciąg Fibonacciego w języku C++](#),
- [Ciąg Fibonacciego w języku Java](#).

Więcej zadań? Przejdź do e-materiału [Ciąg Fibonacciego – zadania maturalne](#).

O tym, jak zagadnienie rekurencji wyjaśnia matematyka, przeczytasz w e-materiałach:

- [Ciąg określony rekurencyjnie](#),
- [Ciąg geometryczny określony rekurencyjnie](#),
- [Wzór ogólny ciągu określonego rekurencyjnie](#),

- Ciąg arytmetyczny określony wzorem rekurencyjnym.

Twoje cele

- Przeanalizujesz rekurencyjny oraz iteracyjny algorytm generujący kolejne elementy ciągu Fibonacciego, zapisany za pomocą języka Python.
- Zaimplementujesz rekurencyjny oraz iteracyjny algorytm generowania kolejnych wyrazów ciągu Fibonacciego, w języku Python.
- Stworzysz program obliczający i wizualizujący związek tzw. złotego podziału i ciągu Fibonacciego.

Schemat interaktywny

Polecenie 1

Zapisz algorytm obliczania elementu ciągu Fibonacciego o indeksie n .

Działanie programu przetestuj dla $n = 8$.

Specyfikacja problemu:

Dane:

- n – liczba naturalna

Wynik:

Program na standardowym wyjściu indeks ciągu Fibonacciego o indeksie n .

1

1

Polecenie 2

Porównaj swoje rozwiązanie z tym przedstawionym w filmie.



Algorytmy rekurencyjne Ciąg Fibonacciego

Implementacja w języku Python



Film dostępny pod adresem </preview/resource/R1Ca9SV985Qli>

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

Film nawiązujący do treści materiału

Plik o rozmiarze 213.00 B w języku polskim

Polecenie 3

Zapisz algorytm obliczania rekurencyjnie n pierwszych elementów ciągu Fibonacciego. Wykorzystaj do tego schemat interaktywny lub język programowania.

Działanie programu przetestuj dla $n = 8$.

Ze względu na ograniczenia schematu blokowego maksymalna wartość n to 25.

Specyfikacja problemu:

Dane:

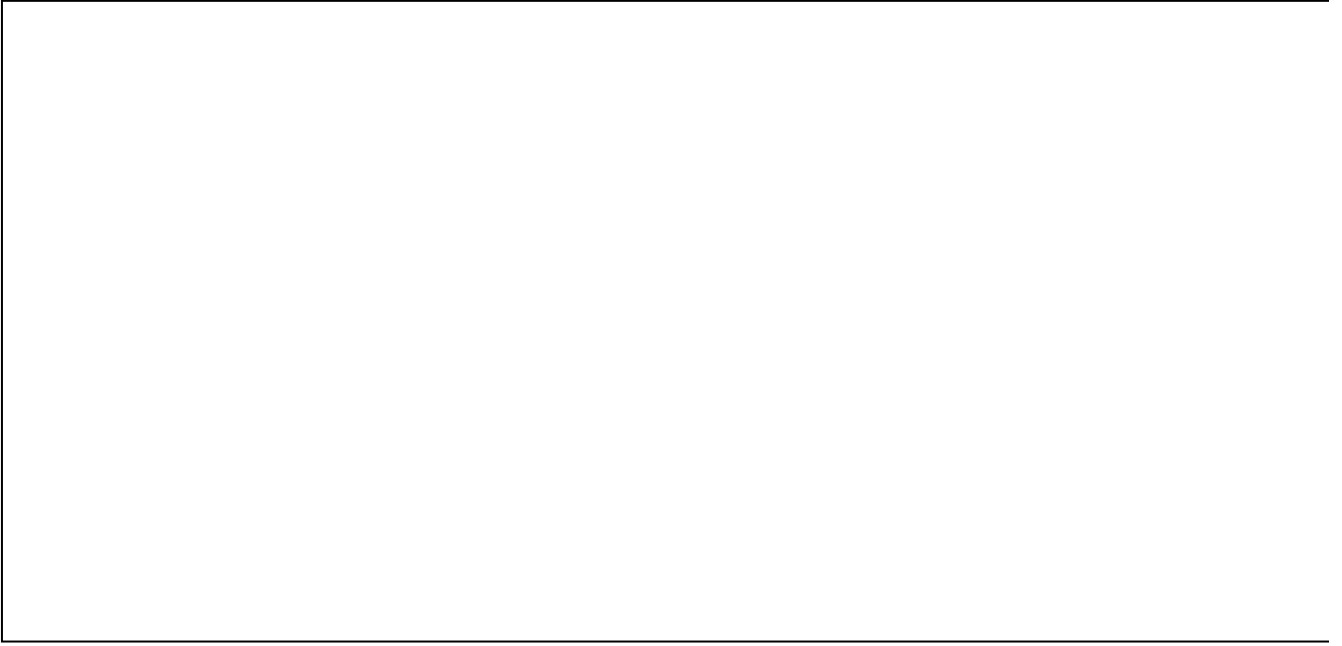
- n – liczba naturalna

Wynik:

Program na standardowym wyjściu wypisze n pierwszych elementów ciągu Fibonacciego.

1

1



Przeczytaj

Ciąg Fibonacciego – podsumowanie

Ciąg Fibonacciego to ciąg złożony z [liczb naturalnych](#) o następujących cechach:

- pierwszy element ciągu jest równy 0,
- drugi element ciągu jest równy 1,
- każdy kolejny element ciągu jest sumą dwóch elementów go poprzedzających.

Definicja ciągu Fibonacciego ma następującą postać:

$$F_n = \begin{cases} 0 & \text{dla } n = 0 \\ 1 & \text{dla } n = 1 \\ F_{n-1} + F_{n-2} & \text{dla } n > 1 \end{cases}$$

Rekurencja

Przykład 1

Oto zapisana w postaci pseudokodu funkcja rekurencyjna, która oblicza wyraz ciągu Fibonacciego o indeksie n:

```
1 rek_fib(n)
2     jeśli n wynosi 0 zwróć 0
3     jeśli n wynosi 1 zwróć 1
4     w innym przypadku zwróć (rek_fib(n - 1) + rek_fib(n - 2))
```

Na podstawie pseudokodu możemy zdefiniować w języku Python funkcję, która zwróci element ciągu o indeksie n. Przetestujmy jej działanie dla indeksu równego 5:

```
1 def rek_fibonacci(n):
2     if n == 0:
3         return 0
4     elif n == 1:
5         return 1
6     else:
7         return rek_fibonacci(n - 1) + rek_fibonacci(n - 2)
8
9 print(rek_fibonacci(5))
```

Wywołując przedstawioną funkcję w pętli, wygenerujemy określoną liczbę elementów ciągu Fibonacciego:

```
1 def rek_fibonacci(n):
2     if n == 0:
3         return 0
4     elif n == 1:
5         return 1
6     else:
7         return rek_fibonacci(n - 1) + rek_fibonacci(n - 2)
8
9 ciag = ''
10 for element in range(10):
11     ciag += str(rek_fibonacci(element)) + ' '
12
13 print(ciag)
```

Polecenie 1



Możesz również przygotować funkcję zwracającą ciąg Fibonacciego jako listę dla dalszych obliczeń. Przygotuj taki kod.

Iteracja

Przykład 2

Oto zapisana w postaci pseudokodu funkcja iteracyjna, która oblicza wyraz ciągu Fibonacciego o indeksie n:

```
1 funkcja iter_fib(n)
2     jeżeli n == 0 wykonaj:
3         zwróć 0
4     w przeciwnym razie jeżeli n == 1 wykonaj:
5         zwróć 1
6
7     a = 0
8     b = 1
9     i = 2
10
11     dopóki i <= n wykonuj:
12         suma = a + b
13         a = b
```

```
14         b = suma
15         i = i + 1
16     zwróć b
```

Na podstawie pseudokodu możemy zdefiniować w języku Python funkcję, która zwróci element ciągu o indeksie n. Przetestujmy jej działanie dla indeksu równego 5:

```
1 def iter_fibonacci(n):
2     if n == 0:
3         return 0
4     elif n == 1:
5         return 1
6
7     a = 0
8     b = 1
9     i = 2
10
11     while i <= n:
12         suma = a + b
13         a = b
14         b = suma
15         i = i + 1
16     return b
17
18 print(iter_fibonacci(5))
```

Wywołując przedstawioną funkcję w pętli, wygenerujemy określoną liczbę elementów ciągu Fibonacciego:

```
1 def iter_fibonacci(n):
2     if n == 0:
3         return 0
4     elif n == 1:
5         return 1
6
7     a = 0
8     b = 1
9     i = 2
10
11     while i <= n:
```

```

12         suma = a + b
13         a = b
14         b = suma
15         i = i + 1
16     return b
17
18 ciag = ''
19 for element in range(10):
20     ciag += str(iter_fibonacci(element)) + ' '
21 print(ciag)

```

Polecenie 2



Możesz również przygotować funkcję zwracającą ciąg Fibonacciego jako listę dla dalszych obliczeń. Przygotuj taki kod.

Złota liczba

Ciąg Fibonacciego jest powiązany z tzw. [złotą liczbą](#). W przybliżeniu wynosi ona 1,618033988... Dzieląc wyraz ciągu o indeksie n przez wyraz o indeksie $n - 1$, otrzymamy iloraz oscylujący wokół 1,618. W miarę tego, jak zwiększają się wartości dzielnej i dzielnika, zmniejsza się odchylenie od wartości złotej liczby.

Przykład 3

Napiszmy program, który dla zadanej liczby elementów ciągu Fibonacciego wypisze kolejne przybliżenia złotej liczby oraz przygotuje wykres obrazujący, jak zmienia się odchylenie od wartości złotej liczby dla kolejnych wyrazów ciągu Fibonacciego, wykorzystując moduł [matplotlib](#).

```

1 def fibonacci_zlota(n):
2     # wczytanie modułu rysującego
3     import matplotlib.pyplot as plt
4
5     # definicja funkcji pomocniczej
6     def rek_fib(n):
7         if n == 0:
8             return 0
9         elif n == 1:
10            return 1
11        else:
12            return rek_fib(n - 1) + rek_fib(n - 2)

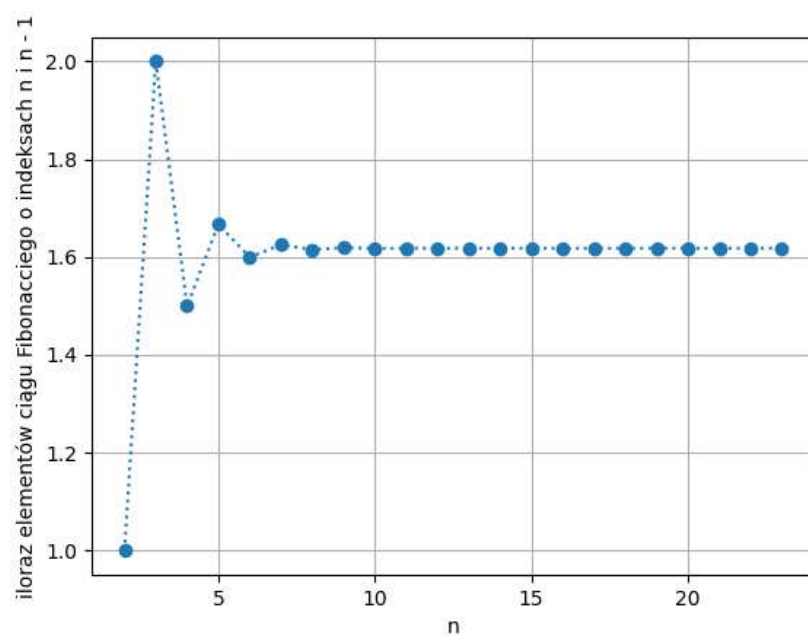
```

```

13
14 lista_x = [ x for x in range(2,n) ]
15 fibonacci = [ rek_fib(x) for x in range(n) ]
16 zlota = [ (fibonacci[x] / fibonacci[x-1]) for x in range(2,
17
18 # wypisujemy na ekranie listę wartości
19 print(zlota)
20
21 # wyświetlamy okno z wykresem
22 # "o:" - punkty połączone przerywaną linią
23 plt.plot(lista_x,zlota,"o:")
24 plt.grid(True)
25 plt.xlabel('n')
26 plt.ylabel('stosunek wyrazu ciągu Fibonacciego o indeksie n')
27 plt.show()
28
29 # przykład wykonania
30 fibonacci_zlota(24)
31 [1.0, 2.0, 1.5, 1.6666666666666667, 1.6, 1.625,
32 1.6153846153846154, 1.619047619047619,
33 1.6176470588235294, 1.6181818181818182,
34 1.6179775280898876, 1.6180555555555556,
35 1.6180257510729614, 1.6180371352785146,
36 1.618032786885246, 1.618034447821682,
37 1.6180338134001253, 1.618034055727554,
38 1.6180339631667064, 1.6180339985218033,
39 1.618033985017358, 1.6180339901755971]

```

Wynikiem będzie wykres prezentujący zależność odchylenia od wartości złotej liczby, dla kolejnych wyrazów ciągu Fibonacciego:



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

Ćwiczenie 1

Napisz w języku Python program, który będzie wyznaczał wyraz ciągu Fibonacciego o indeksie n .

Specyfikacja problemu:

Dane:

- n – liczba naturalna, indeks wyznaczanego elementu

Wynik:

Na wyjściu standardowym program wypisze wyraz ciągu Fibonacciego o indeksie n .



Ćwiczenie 2

Uruchom funkcję generującą wyraz ciągu Fibonacciego o indeksie 45, wykorzystując funkcję rekurencyjną i iteracyjną. Zmierz czas potrzebny na wykonanie obliczeń programu. Porównaj otrzymane wyniki i zapisz wnioski.

Słownik

Fibonacci

włoski matematyk, ur. ok. 1175 r., znany jako Leonardo Fibonacci, Filius Bonacci (syn Bonacciego), Leonardo Pisano (Leonardo z Pizy)

liczby naturalne

liczby całkowite nieujemne

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 Pythona; należy go zainstalować, korzystając z mechanizmu `pip`

przestrzeń nazw

(ang. *namespace*) miejsce w pamięci operacyjnej, w której jest przechowywana dana zmienna; najczęściej przestrzeń nazw jest związana z funkcją, a zmienne w niej używane są widoczne tylko w obrębie tej przestrzeni – jest to „zakres obowiązywania” tych zmiennych

złota liczba

inaczej boska proporcja (łac. *divina proportio*) – liczba, której wartość wynosi w przybliżeniu $1.618033988\dots$; w geometrii podział odcinka na dwie części, tak by

stosunek długości dłuższej z nich do krótszej był taki sam, jak całego odcinka do części dłuższej

Sprawdź się

Pokaż ćwiczenia:   

Ćwiczenie 1



Napisz funkcję, która zwróci liczbę wywołań rekurencyjnych funkcji `fibonacci` dla obliczenia elementu ciągu Fibonacciego o indeksie `n`. Przetestuj działanie programu dla indeksu równego 11.

Specyfikacja problemu:

Dane:

- `n` – liczba naturalna, indeks elementu ciągu

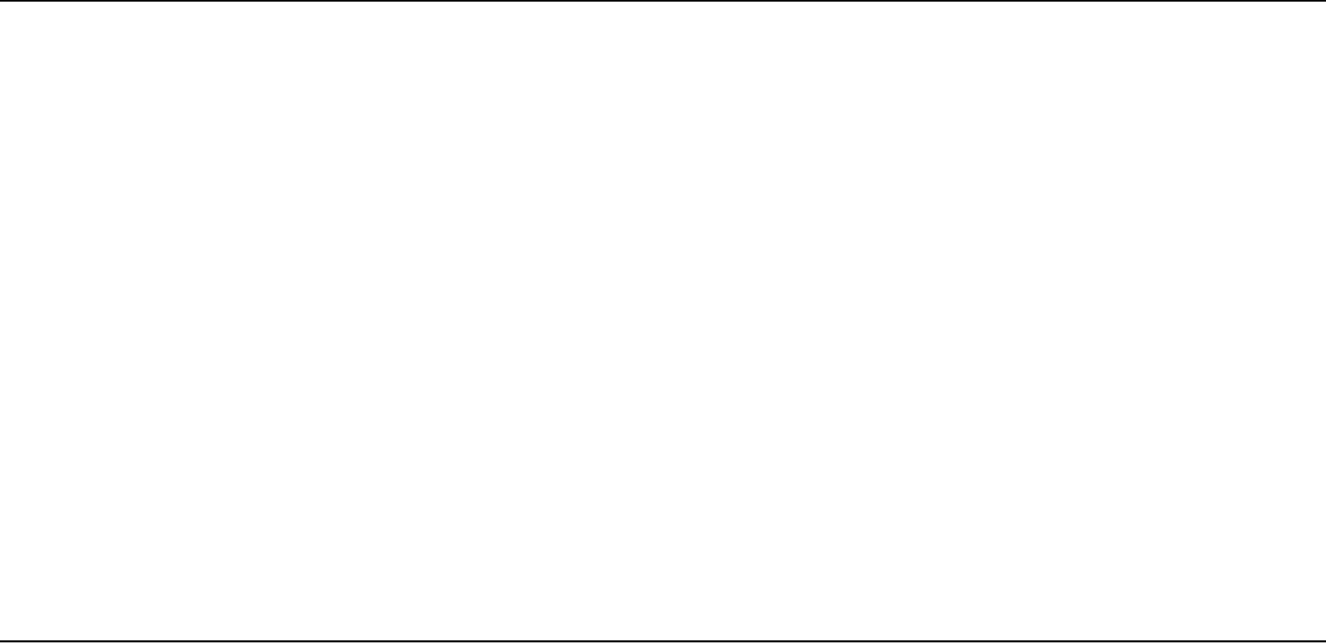
Wynik:

Program zwróci liczbę wywołań rekurencyjnych funkcji `fibonacci`.

Twoje zadania

1. Zdefiniowanie funkcji `ile_wywolan`.
2. Obliczenie, ile razy funkcja wywołuje się dla danego indeksu `n`.

```
1 licznik = 0
2
3 def fibonacci(n):
4     # Tu uzupełnij kod
5     pass
6
7 def ile_wywolan(n):
8     # Tu uzupełnij kod
9     pass
```





Napisz program, który obliczy element ciągu Fibonacciego o indeksie n , a następnie sprawdzi, czy ten element jest liczbą pierwszą i wypisze odpowiedni komunikat. Przetestuj działanie swojego programu dla elementu ciągu Fibonacciego o indeksie równym 11.

Specyfikacja:

Dane:

- n – liczba naturalna, indeks elementu ciągu Fibonacciego

Wynik:

Program wyświetla element ciągu Fibonacciego o indeksie n oraz komunikat
to liczba pierwsza, jeżeli element ciągu Fibonacciego o indeksie n jest liczbą pierwszą
lub to nie liczba pierwsza, jeżeli element o indeksie n nie jest liczbą pierwszą.

Twoje zadania

1. Program wypisuje element ciągu Fibonacciego o indeksie n oraz po znaku spacji komunikat to liczba pierwsza, jeżeli ten element ciągu jest liczbą pierwszą lub to nie liczba pierwsza, jeżeli ten element ciągu nie jest liczbą pierwszą.

```
1 def fibonaccini):
2     // Tu uzupełnij kod
3
4 def czyPierwsza(wynik):
5     // Tu uzupełnij kod
6
7 n = 11
8 wynik = fibonaccini)
9 if czyPierwsza(wynik):
10     print(wynik, "to liczba pierwsza")
11 else:
12     print(wynik, "to nie liczba pierwsza")
```




Napisz program, który obliczy liczbę wywołań rekurencyjnych funkcji `fibonacci()` z argumentami 2, 1 oraz 0 przy obliczaniu elementu ciągu Fibonacciego o indeksie `n`. Przetestuj działanie swojego programu dla elementu ciągu o indeksie równym 13.

Specyfikacja:

Dane:

- `n` – liczba naturalna, indeks elementu ciągu Fibonacciego

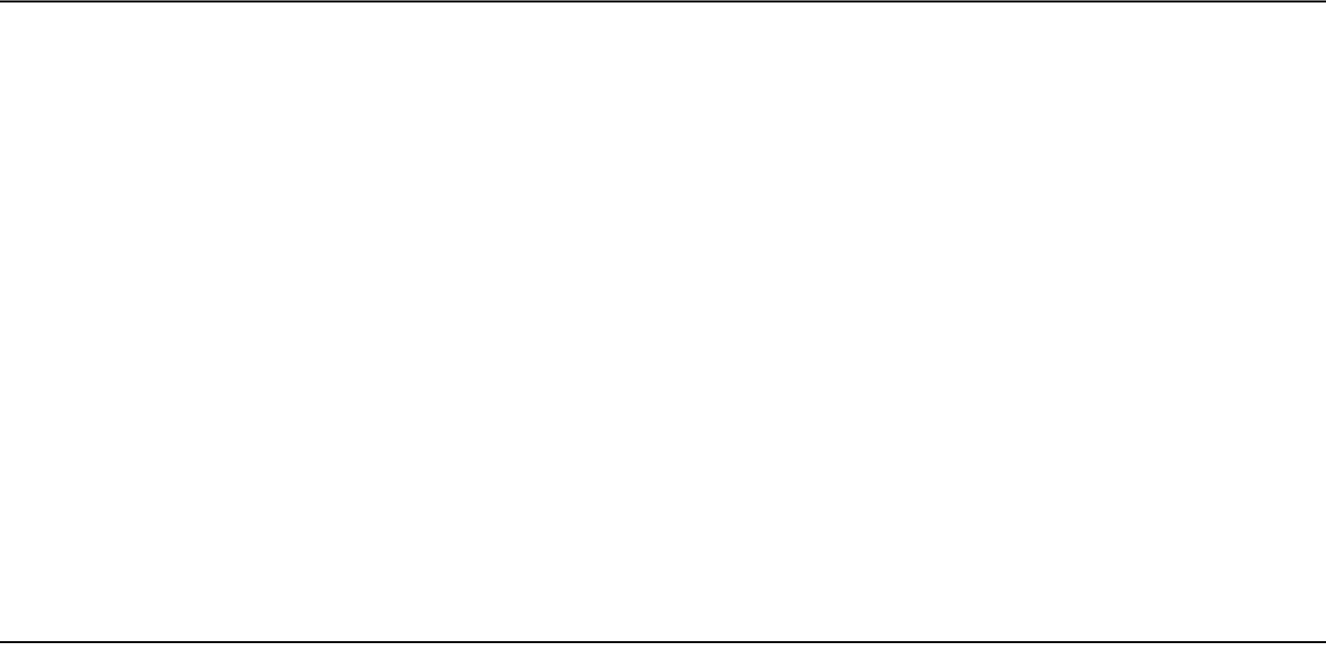
Wynik:

Program wyświetla liczbę wywołań rekurencyjnych `fibonacci(2)`, `fibonacci(1)` oraz `fibonacci(0)` przy obliczaniu elementu ciągu Fibonacciego o indeksie `n`.

Twoje zadania

1. Program wypisuje trzy wartości: liczbę wywołań `fibonacci(2)`, `fibonacci(1)` oraz `fibonacci(0)` przy obliczaniu elementu ciągu Fibonacciego o indeksie równym `n`.

```
1 licznik0 = 0
2 licznik1 = 0
3 licznik2 = 0
4
5 def fibonacci(n):
6     // Tu uzupełnij kod
7
8 ile_wywolan(13)
```



Dla nauczyciela

Autor: Zespół autorski Contentplus.pl sp. z o.o.

Przedmiot: Informatyka

Temat: Ciąg Fibonacciego w języku Python

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.

Treści nauczania – wymagania szczegółowe

I. Rozumienie, analizowanie i rozwiązywanie problemów.

Zakres podstawowy. Uczeń:

2) stosuje przy rozwiązywaniu problemów z różnych dziedzin algorytmy poznane w szkole podstawowej oraz algorytmy:

e) obliczania wartości elementów ciągu metodą iteracyjną i rekurencyjną, w tym wartości elementów ciągu Fibonacciego.

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):

- Przeanalizujesz rekurencyjny oraz iteracyjny algorytm generujący kolejne elementy ciągu Fibonacciego, zapisany za pomocą języka Python.
- Zaimplementujesz rekurencyjny oraz iteracyjny algorytm generowania kolejnych wyrazów ciągu Fibonacciego, w języku Python.

- Stworzysz program obliczający i wizualizujący związek tzw. złotego podziału i ciągu Fibonacciego.

Strategie nauczania:

- konstruktywizm;
- konektywizm.

Metody i techniki nauczania:

- dyskusja;
- rozmowa nauczająca z wykorzystaniem multimediu 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 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: „Ciąg Fibonacciego w języku Python”. Uczniowie zapoznają się z treściami w sekcji „Schemat interaktywny” w kontekście programowania.

Faza wstępna:

1. Przedstawienie tematu zajęć oraz wspólne z uczniami ustalenie kryteriów sukcesu.
2. Prowadzący prosi uczniów, aby zgłaszali swoje propozycje pytań do tematu. Jedna osoba może zapisywać je na tablicy. Gdy uczniowie wyczerpią swoje pomysły, a pozostały jakieś ważne kwestie do poruszenia, nauczyciel je dopowiada.

Faza realizacyjna:

1. **Praca z multimediu.** Nauczyciel wyświetla zawartość sekcji „Schemat interaktywny”. Uczniowie rozwiązują Polecenie 1 i w parach budują schemat lub indywidualnie piszą

program, wykorzystując język programowania.

2. **Praca z tekstem.** Chętna lub wybrana osoba referuje najważniejsze informacje związane z ciągiem Fibonacciego. Uczniowie wykonują ćwiczenie 2 z sekcji „Przeczytaj”.
3. **Praca z multimediami.** Uczniowie rozwiązują Polecenie 2 indywidualnie pisząc program, wykorzystując język programowania. Porównują swoją odpowiedź z filmem.
4. **Ćwiczenie umiejętności.** Uczniowie indywidualnie rozwiązują ćwiczenia 1 i 2 z sekcji „Sprawdź się”.
5. Uczniowie w parach analizują napisane przez siebie programy. Chętne lub wybrane osoby przedstawiają je na forum klasy. Nauczyciel odpowiada na ewentualne pytania.

Faza podsumowująca:

1. Wybrany uczeń podsumowuje zajęcia z programowania w Pythonie, zwracając uwagę na nabyte umiejętności.

Praca domowa:

1. Uczniowie wykonują Polecenie nr 1 z sekcji „Przeczytaj”. Ich zadaniem jest przygotowanie funkcji zwracającej ciąg Fibonacciego jako listę.

Materiały pomocnicze:

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

Wskazówki metodyczne:

- Treści w sekcji „Schemat interaktywny” można wykorzystać jako podsumowanie i utrwalenie wiedzy uczniów.