

Programowanie I

Zajęcia nr 12

Rafał Masełek

26. maja 2022r.

Zadanie 1 – silnia za pomocą generatora

Zaimplementuj liczenie silni za pomocą generatora. Używając "next" wyświetl $n!$ dla $n \in [0, 10)$.

Zadanie 2 – ciąg Fibonacciego

Stwórz generator, który będzie generował kolejne liczby z ciągu Fibonacciego. Wypisz pierwsze 15 wyrazów.

Zadanie 3 – palindrom

Napisz generator "infinite_sequence" generujący kolejne liczby całkowite począwszy od 0. Napisz funkcję "is_palindrome", która przyjmuje jedną liczbę całkowitą i sprawdza, czy dana liczba jest palindromem. Przyjmij, że palindrom to liczba co najmniej dwucyfrowa, która po odwróceniu kolejności cyfr jest taka sama, np. 1771 jest palindromem. Korzystając ze zdefiniowanego generatora wypisz pierwsze 50 palindromów.

Zadanie 4 – generator expressions

Wykorzystując "generator expressions" napisz generator zwracający kolejne wyrazy rozwinięcia funkcji $\sin x$ w szereg Taylora. Korzystając z tego generatora, przyjmij $x = \pi$ i wypisz pierwsze 10 wyrazów wraz z ich numerami. Dodatkowo wypisz jak zmienia się suma szeregu po uwzględnieniu kolejnych wyrazów i zaobserwuj jak zbiega do 0.

Zadanie 5 – Lista jednokierunkowa

Lista jednokierunkowa jest strukturą danych zachowującą pewien porządek i umożliwiającą iterowanie. Podstawowym elementem listy jest węzeł (ang. node). Każdy węzeł zawiera dane, jakie chcemy przechowywać, oraz referencję (odnośnik, link) do kolejnego węzła. Jeżeli mamy dostęp do jakiegoś węzła, to za pomocą referencji możemy przejść do następnego, który ma referencję do kolejnego, i tak dalej aż dojdziemy do końca listy. W przypadku listy jednokierunkowej każdy węzeł ma tylko jedną referencję do sąsiedniego węzła (w "przód" lub w "tył"). Węzły listy dwukierunkowej mają dwie referencje w obu kierunkach.

Stwórz klasę Node reprezentującą węzeł listy. Klasa ma dwa pola, jeden na "dane" a drugi będący referencją do następnego węzła. Dane powinny być podawane jako argument konstruktora i dostępne na zewnątrz klasy poprzez metodę "get_data". Referencja powinna być domyślnie "None", odczytywana za pomocą metody "get_next", ustawiana za pomocą metody "set_next".

Stwórz klasę "LinkedList". Klasa ma jedno pole, przechowujące jeden z końców listy (Node). Zdefiniuj metodę "_iter_" oraz metodę "add". Ta druga tworzy i dodaje do listy nowy węzeł z danymi podanymi jako argument metody.

Stwórz klasę "LinkedListIterator", która implementuje funkcjonalność iteratora. Klasa ma pole przechowujące informację o "aktualnym" elemencie listy, tworzone w konstruktorze. Klasa posiada metody "_iter_" oraz "_next_".

Druga z metod zwraca dane z aktualnego węzła i "przesuwa" iterator, a w przypadku końca listy rzuca wyjątek "StopIteration".

W programie stwórz listę, dodaj do niej kilka elementów i wypisz je iterując po liście (for item in linked_list).

Zadanie 6 – ciągi matematyczne (autor: B. Zglinicki)

Zadeklaruj klasę "Sequence", reprezentującą ciąg matematyczny i zawierającą iterator oraz operator [], który zwraca żądany wyraz ciągu, a także jednoargumentową metodę "addElement", dopisującą do ciągu nowy wyraz, i jednoargumentową metodę "sum", zwracającą sumę początkowych wyrazów ciągu w liczbie określonej jej argumentem. Z klasy wyprowadź klasę pochodną Arithmetic, reprezentującą ciąg arytmetyczny określony wzorem

$$a_n = a_0 + n\Delta \quad (1)$$

Zdefiniuj dla niej konstruktor umożliwiający nadanie wartości parametrom a_0 i Δ oraz przeciąż te metody klasy bazowej "Sequence", które tego wymagają.

Z klasy Sequence wywiedź ponadto klasę pochodną "Fibonacci", reprezentującą ciąg Fibonacciego, określony zależnością rekurencyjną

$$a_0 = 0, a_1 = 1, a_n = a_{n-1} + a_{n-2} \quad (2)$$

W przypadku tej klasy również zdefiniuj odpowiedni konstruktor oraz przeciąż te metody klasy bazowej Sequence, które tego wymagają.

Wykorzystując te klasy, napisz program "sequences", który wypisuje na ekranie początkowe 10 wyrazów ciągu arytmetycznego o $a_0 = 2$ i $\Delta = 3$ korzystając z operatora [] klasy Arithmetic oraz początkowe 10 wyrazów ciągu Fibonacciego, używając iteratora klasy Fibonacci.