

Programowanie 1R – Ćwiczenia 8

Klasy (dziedziczenie)

Mikołaj Marcinkowski

28 kwietnia 2026

Zadania możesz wykonywać jako oddzielne skrypty `.py` lub w notatniku jupytera `.ipynb`, chyba że treść zadania wskazuje inaczej.

Zadania wzorowane na Serii 8 zadań autorstwa Bartłomieja Zglinickiego (<https://www.fuw.edu.pl/~bzglinicki/teaching/plr/zadania/seria-8/>), udostępnionej na licencji CC BY 4.0.

Zadanie 1. Wielomiany

Napisz klasę `Wielomian` reprezentującą wielomian zmiennej rzeczywistej. Klasa ta powinna zawierać:

- pole `c`, będące listą przechowującą współczynniki wielomianu. Niech n -ty wyraz `c` odpowiada współczynnikowi przy x^n ;
- konstruktor, przyjmujący jako argumenty współczynniki wielomianu (możesz użyć `*args`);
- metodę `Deg`, zwracającą stopień wielomianu;
- operator wywołania `()` (metoda `__call__`), przyjmujący jako argument liczbę x i zwracający wartość wielomianu dla tej liczby;
- metodę `__str__`, zwracającą wielomian napisany jako napis:
 $c_0x^0 + c_1x^1 + c_2x^2 + \dots$;
- operator `[]`, umożliwiający odczyt (metoda `__getitem__`) i modyfikację (metoda `__setitem__`) współczynników wielomianu.
 - Jeśli spróbujemy odczytać współczynnik powyżej stopnia wielomianu, operator ten powinien zwrócić zero.

- Jeśli spróbujemy nadpisać współczynnik powyżej stopnia wielomianu, operator ten powinien odpowiednio dodać elementy do listy `c`, podnosząc stopień wielomianu;
- operator dodawania wielomianów `+`;
- operator mnożenia wielomianu przez liczbę `*` (powinien działać niezależnie od kolejności argumentów);
- metodę `D`, zwracającą nowy obiekt klasy `Wielomian` reprezentujący pochodną wielomianu.

Z klasy `Wielomian` wywiedź klasy pochodne `WielomianHermitea` i `WielomianLegendrea`, reprezentujące odpowiednio wielomiany Hermite’a $H_n(x)$ i Legendre’a $L_n(x)$. W każdej z nich zaimplementuj konstruktor, który zamiast listy współczynników przyjmuje liczbę całkowitą n reprezentującą stopień wielomianu. W razie potrzeby przeciąż odpowiednie metody z klasy bazowej.

Przetestuj działanie tych klas. Możesz np. sprawdzić wzory rekurencyjne na $H_n(x)$ i $L_n(x)$, jak $\frac{d}{dx}H_n(x) = 2nH_{n-1}(x)$.

Wielomiany Legendre’a dane są wzorem:

$$L_n(x) = 2^n \sum_{k=0}^n \binom{n}{k} \binom{\frac{n+k-1}{2}}{n} x^k, \quad (1)$$

gdzie:

$$\binom{\alpha}{k} = \frac{\alpha(\alpha-1)(\alpha-2)\cdots(\alpha-(k-1))}{k(k-1)(k-2)\cdots 1}. \quad (2)$$

Wielomiany Hermite’a dane są wzorem:

$$H_n(x) = n! \sum_{m=0}^{n//2} \frac{(-1)^m \cdot 2^{n-2m}}{m!(n-2m)!} x^{n-2m}, \quad (3)$$

gdzie $n//2$ to wynik dzielenia n przez 2 bez reszty. Tak się składa, że wszystkie współczynniki wielomianów Hermite’a są całkowite, więc możesz zastąpić wszystkie dzielenia we wzorze (3) dzieleniem całkowitym `//`.

Dla chętnych: Zadanie 4 z serii 7 polegało na zrobieniu klasy liczb wymiernych. Klasę `Wielomian` można dość łatwo zmodyfikować, by wspierała współczynniki będące liczbami wymiernymi. Przyda się to zwłaszcza w wielomianach Legendre’a, które mają wymierne współczynniki. Możesz skorzystać z gotowego rozwiązania pod adresem https://github.com/mmarcinkowski/Programowanie-IR/blob/main/tydzien_8/Z_4_wymierne.py.

Zadanie 2. Średnie

Napisz abstrakcyjną klasę **Sredna** reprezentującą uogólnioną średnią zbioru liczb. Po tej klasie dziedziczyły będą przyszłe klasy implementujące konkretne rodzaje średnich. Klasa **Sredna** powinna zawierać

- pole **x** będące listą liczb, których średnia jest obliczana;
- konstruktor, przyjmujący jako argument listę liczb, których średnia jest obliczana;
- metodę **N**, zwracającą liczbę tych liczb;
- wirtualny bezargumentowy operator wywołania **()** (metoda **__call__**). W klasach dziedziczących wywołanie **obiekt()** ma zwracać odpowiednio obliczoną średnią, ale w tej (abstrakcyjnej) klasie powinno wyrzucać błąd **NotImplementedError**.

Z klasy **Sredna** wywiedź trzy klasy pochodne: **SredniaArytmetyczna**, **SredniaGeometryczna** i **SredniaHarmoniczna**, reprezentujące odpowiednio średnie arytmetyczną, geometryczną i harmoniczną. W każdej z nich zaimplementuj przeciążony operator **()** zwracający odpowiednio obliczoną średnią z liczb w **x**. W razie potrzeby przeciąż odpowiednie metody z klasy bazowej.

Korzystając z tych klas napisz program, który oblicza te średnie z listy liczb.

Na wszelki wypadek, średnie te dane są wzorami:

$$\text{średnia arytmetyczna} = \frac{1}{N} \sum_{i=0}^N x_i, \quad (4)$$

$$\text{średnia geometryczna} = \sqrt[N]{\prod_{i=0}^N x_i}, \quad (5)$$

$$\text{średnia harmoniczna} = \frac{1}{\sum_{i=0}^N \frac{1}{x_i}}, \quad (6)$$

gdzie x_i to kolejne liczby.