Wielomiany gęste
na bazie list Pythona

Autor: Alicja Fedor

08.02.2026

1. Wstęp i przeznaczenie

Projekt udostępnia klasę Polynomial, przeznaczoną do reprezentacji oraz wykonywania operacji na wielomianach o współczynnikach liczbowych. Współczynniki mogą być liczbami całkowitymi, zmiennoprzecinkowymi oraz innymi typami liczbowymi zgodnymi z numbers.Number (np. Fraction).

Wielomiany przechowywane są w reprezentacji gęstej na bazie list Pythona. Klasa implementuje przeciążenia standardowych operatorów arytmetycznych, dzięki czemu działania na wielomianach mogą być wykonywane w sposób analogiczny do operacji na liczbach. Umożliwia to wygodne tworzenie, łączenie i obliczanie wartości wielomianów w kodzie programu.

2. Zawartość projektu

Projekt składa się z dwóch plików:

polynomial.py: Zawiera definicję klasy Polynomial wraz z implementacją reprezentacji danych oraz operacji matematycznych na wielomianach.

test.py: Zbiór testów jednostkowych wykorzystujących moduł unittest, sprawdzający poprawność działań, obsługę przypadków brzegowych oraz współpracę z typem Fraction.

3. Jak uruchomić projekt?

Aby upewnić się, że wszystko działa poprawnie, należy uruchomić plik z testami w terminalu:

python test.py

Przykład użycia w kodzie

Możesz zaimportować klasę do własnego skryptu:

```
from polynomial import Polynomial

p1 = Polynomial([1, 2, 3]) # 3x^2 + 2x + 1
p2 = Polynomial([0, 1])    # x
result = p1 * p2           # 3x^3 + 2x^2 + x
print(result)              # Wyświetli: 3x^3 + 2x^2 + 1x
```

4. Opis działania i algorytmów

- Reprezentacja danych
    - Wielomian przechowywany jest jako lista współczynników self.coeffs.
    - Indeks listy odpowiada potędze zmiennej x.
    - Przykład: [5, 0, 2] reprezentuje 2x^2 + 0x + 5.

- Normalizacja: Klasa automatycznie usuwa końcowe zera z listy (np. [1, 2, 0, 0] staje się [1, 2]), aby stopień wielomianu był zawsze poprawnie określony.
- Konstruktor akceptuje pojedynczą liczbę lub iterowalną kolekcję współczynników (np. lista, krotka)
- Przykłady:

  Polynomial(5) - 5
  Polynomial([1,2,3]) - 3x^2 + 2x + 1
  Polynomial([]) - 0

- Algorytmy operacji
  - Dodawanie i Odejmowanie

    Algorytm wyrównuje długości list współczynników obu wielomianów, dodając wirtualne zera tam, gdzie stopień jednego wielomianu jest mniejszy.

    $$C_i = A_i + B_i$$

    Operacja wykonywana jest w czasie liniowym O(n), gdzie n to maksymalny stopień wielomianu.

  - Mnożenie

    Zastosowano klasyczny algorytm mnożenia "każdy z każdym". Jeśli mnożymy wielomian stopnia n przez wielomian stopnia m, wynik będzie stopnia n+m. Dla każdego elementu “i” z pierwszego wielomianu i elementu “j” z drugiego, iloczyn współczynników dodawany jest do pozycji “i+j” w liście wynikowej.

    Złożoność: O(nm).

  - Obliczanie wartości (Schemat Hornera)

    W metodzie value(x) zastosowano Schemat Hornera, który jest najbardziej efektywnym sposobem obliczania wartości wielomianu. Zamiast potęgować x wielokrotnie, wielomian przekształcany jest do postaci zagnieżdżonych mnożeń:

    $$((a_n * x + a_{n-1}) * x + ...) + a_0$$

    Dzięki temu wykonujemy tylko n mnożeń i n dodawań, unikając kosztownego potęgowania.

5. Kluczowe metody klasy

- __init__, Inicjalizuje wielomian, usuwa zbędne zera i obsługuje inicjalizację pojedynczą liczbą.
- is_zero(), Sprawdza, czy wielomian jest wielomianem zerowym.
- degree(), Zwraca stopień wielomianu (długość listy - 1).
- value(), Oblicza wartość wielomianu dla podanego x przy użyciu Schematu Hornera.

- __getitem__, Operator [], Pozwala pobrać współczynnik przy danej potędze (np. p[2] dla x^2).
- __str__, Formatuje wielomian do czytelnej postaci (np. 3x^2 - 2x + 1).
- __add__, Operator +, Dodaje wielomian do innego wielomianu lub liczby (prawostronnie).
- __sub__, Operator -, Odejmuje wielomian lub liczbę od aktualnego obiektu.
- __mul__, Operator *, Mnoży wielomian przez inny wielomian lub liczbę.
- __radd__, __rsub__, __rmul__, Zapewniają przemienność (np. 5 + p działa tak samo jak p + 5).
- __eq__, Operator ==, Sprawdza równość dwóch wielomianów (porównuje ich różnicę z zerem).
- __ne__, Operator !=, Sprawdza nierówność.