

# Programowanie I R

Zadania – seria 8.

Programowanie obiektowe – dziedziczenie.

## Zadanie 1. `polynomial` – Wielomiany.

Napisz klasę `Polynomial` reprezentującą wielomian zmiennej rzeczywistej nad ciałem liczb rzeczywistych.

Zaimplementuj:

- listę `c`, przechowującą współczynniki wielomianu,
- konstruktor, przyjmujący jako argumenty listę współczynników wielomianu,
- metodę `Deg`, zwracającą stopień wielomianu,
- operator `[]`, umożliwiający dostęp (odczyt wartości i jej modyfikację) do współczynników wielomianu,
- operator wywołania `()`, przyjmujący jako argument liczbę rzeczywistą i zwracający wartość wielomianu od tej liczby,
- operator dodawania wielomianów `+`,
- operator mnożenia wielomianu przez liczbę rzeczywistą `*`, działający niezależnie od kolejności argumentów,
- metodę `D`, zwracającą nową instancję klasy `Polynomial` reprezentującą pochodną wielomianu.

Z klasy `Polynomial` wywiedź dwie klasy pochodne `HermitePolynomial` i `LegendrePolynomial`, reprezentujące, odpowiednio, wielomian Hermite'a i wielomian Legendre'a. W każdej z nich zaimplementuj konstruktor jednoargumentowy, przyjmujący jako argument stopień wielomianu, oraz – w razie potrzeby – przeciążenia odpowiednich metod z klasy bazowej.

Korzystając z tych klas napisz program `polynomial` przyjmujący jako argumenty wywołania liczbę naturalną  $n$  oraz liczbę rzeczywistą  $x$  i wypisujący na standardowe wyjście, w kolejnych liniach, liczby

$$H_n(x), \quad L_n(x), \quad H'_n(x) + L'_n(x) + 3(H_n(x) + L_n(x)),$$

gdzie  $H_n$  i  $L_n$  oznaczają, odpowiednio, wielomian Hermite'a stopnia  $n$  i wielomian Legendre'a stopnia  $n$ .

*Wskazówka.* Współczynniki wielomianów Hermite'a i Legendre'a określonego stopnia można łatwo otrzymać wykorzystując pakiet `SymPy`. Pakiet ten należy najpierw zainstalować, wydając bezpośrednio w Terminalu kolejno następujące polecenia:

```
python3 -m pip install -U pip
python3 -m pip install -U sympy
```

Funkcje

```
def HermiteCoefficients(n):
    from sympy import hermite, Poly, Symbol
    x = Symbol('x')
    return Poly(hermite(n, x), x).coeffs()
```

```
def LegendreCoefficients(n):  
    from sympy import legendre, Poly, Symbol  
    x = Symbol('x')  
    return Poly(legendre(n, x), x).coeffs()
```

zwrócić listę współczynników wielomianu, odpowiednio, Hermite'a lub Legendre'a stopnia  $n$ .

## Zadanie 2. mean – Średnie.

Napisz abstrakcyjną klasę `Mean` reprezentującą średnią zbioru liczb.

Zaimplementuj:

- listę `x`, przechowującą liczby, których średnia jest obliczana,
- konstruktor, przyjmujący jako argumenty listę liczb, których średnia ma być obliczana,
- metodę `N`, zwracającą ilość liczb,
- wirtualny bezargumentowy operator wywołania `()`.

Z klasy `Mean` wywiedź trzy klasy pochodne: `ArithmeticMean`, `GeometricMean` i `HarmonicMean`, reprezentujące, odpowiednio, średnie arytmetyczną, geometryczną i harmoniczną. W każdej z nich zaimplementuj przeciążony operator `()`, zwracający właściwą średnią liczb, oraz – w razie potrzeby – przeciążenia odpowiednich metod z klasy bazowej.

Korzystając z tych klas napisz program `mean` wczytujący ze standardowego wejścia liczby rzeczywiste aż do napotkania znaku końca pliku, a następnie wypisujący na standardowe wyjście, w kolejnych liniach, ich średnie: arytmetyczną, geometryczną i harmoniczną.

*Opracowanie: Bartłomiej Zglinicki.*