

Programowanie II R

Zadania – seria 10.

Programowanie obiektowe – dziedziczenie.

Zadanie 1. polynomial – Wielomiany.

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

Zaimplementuj:

- chroniony wektor `c`, przechowujący współczynniki wielomianu,
- konstruktor jednoargumentowy, przyjmujący jako argument wektor współczynników wielomianu,
- publiczną 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 dla tej liczby,
- operator dodawania wielomianów `+`,
- operator mnożenia wielomianu przez liczbę rzeczywistą `*`, działający niezależnie od kolejności argumentów,
- publiczną metodę `D`, zwracającą nową instancję klasy `Polynomial` reprezentującą pochodną wielomianu.

Z klasy `Polynomial` wywiedź dwie klasy pochodne: `TaylorSin` i `TaylorCos`, reprezentujące wielomiany zbudowane z pierwszych wyrazów rozwinięcia w szereg Taylora wokół $x = 0$ funkcji, odpowiednio, sinus i cosinus. W każdej z tych klas zaimplementuj konstruktor jednoargumentowy, przyjmujący jako argument stopień wielomianu (wielomian stopnia n zawiera pierwszych $n + 1$ wyrazów rozwinięcia), oraz – w razie potrzeby – przeciążenia odpowiednich metod z klasy bazowej.

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

$$S_n(x), \quad C_n(x), \quad S'_n(x) + C'_n(x) + 3(S_n(x) + C_n(x)),$$

gdzie S_n i C_n oznaczają wielomiany Taylora stopnia n dla funkcji, odpowiednio, sinus i cosinus.

Zadanie 2. mean – Średnie.

Napisz abstrakcyjną klasę `Mean` reprezentującą średnią zbioru liczb.

Zaimplementuj:

- chroniony zbiór `x`, przechowujący liczby, których średnia jest obliczana,

- konstruktor jednoargumentowy, przyjmujący jako argument kolekcję liczb, których średnia ma być obliczana,
- publiczną metodę `N`, zwracającą ilość liczb,
- wirtualny bezargumentowy operator wywołania `()`.

Z klasy `Mean` wywiedź trzy klasy pochodne: `ArithmeticMean`, `GeometricMean` i `HarmonicMean`, reprezentujące średnie, odpowiednio, arytmetyczną, geometryczną i harmoniczną. W każdej z tych klas zaimplementuj przeciążony operator `()`, zwracający odpowiednią 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.

Zadanie 3. sequence – Ciągi liczbowe.

Napisz abstrakcyjną klasę `Sequence` reprezentującą ciąg liczb rzeczywistych.

Zaimplementuj:

- wirtualny operator indeksowania `[]`,
- publiczną metodę `Sum`, zwracającą sumę pierwszych wyrazów ciągu w ilości przekazanej jako argument metody.

Z klasy `Sequence` wywiedź dwie klasy pochodne:

- klasę `FibonacciSequence`, reprezentującą ciąg Fibonacciego $(F_n)_{n=0}^{\infty}$ określony wzorem

$$F_n \stackrel{\text{def}}{=} \begin{cases} n, & \text{gdy } n = 0 \text{ lub } n = 1, \\ F_{n-1} + F_{n-2}, & \text{w pozostałych przypadkach;} \end{cases}$$

zaimplementuj w niej konstruktor bezargumentowy oraz przeciążony operator indeksowania `[]`, zwracający zadany wyraz ciągu Fibonacciego (nie pozwalający jednak na zmianę jego wartości),

- klasę `CollatzSequence`, reprezentującą ciąg Collatza $(c_n^k)_{n=0}^{\infty}$, $k \in \mathbb{N}$, określony wzorem

$$c_n^k \stackrel{\text{def}}{=} \begin{cases} k, & \text{gdy } n = 0, \\ \frac{1}{2}c_{n-1}, & \text{gdy } c_{n-1} \text{ jest liczbą parzystą,} \\ 3c_{n-1} + 1, & \text{gdy } c_{n-1} \text{ jest liczbą nieparzystą;} \end{cases}$$

zaimplementuj w niej konstruktor jednoargumentowy, przyjmujący jako argument wartość parametru k , przeciążony operator indeksowania `[]`, zwracający zadany wyraz ciągu Collatza (nie pozwalający jednak na zmianę jego wartości), oraz publiczną metodę `Length`, zwracającą liczbę wyrazów ciągu Collatza aż do pierwszego wyrazu o wartości 1 włącznie.

Korzystając z tych klas napisz program `sequence` przyjmujący jako argumenty wywołania dodatnie liczby całkowite m i k , a następnie wypisujący na standardowe wyjście, w kolejnych liniach: m początkowych wyrazów ciągu Fibonacciego, sumę tych wyrazów, początkowe wyrazy ciągu Collatza z parametrem k aż do pierwszego wyrazu o wartości 1 włącznie oraz sumę tych wyrazów.

Opracowanie: Bartłomiej Zglinicki.