
KURS JĘZYKA C++

OPERACJE NA WIELOMIANACH

Instytut Informatyki Uniwersytetu Wrocławskiego

Paweł Rzechonek

Zadanie.

Zdefiniuj klasę `wielomian` do przechowywania wielomianu o określonym stopniu.

Wielomianem stopnia n zmiennej rzeczywistej x nazywamy wyrażenie postaci

$$a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0$$

gdzie $n \in \mathbb{N}$ to stopień wielomianu oraz $a_n, a_{n-1}, \dots, a_1, a_0 \in \mathbb{R}$ to współczynniki wielomianu, przy czym $a_n \neq 0$.

Zaprojektuj tę klasę tak, aby stopień wielomianu oraz jego współczynniki były niepubliczne (współczynniki pamiętaj w tablicy utworzonej na stacku). Oczywiście należy zapewnić metody pozwalające odczytywać i ustawiać te pola: do odczytu stopnia wielomianu zdefiniuj składową funkcję dostępową a do odczytu i zapisu poszczególnych współczynników zdefiniuj operatory indeksowania (inny do czytania i inny do wpisania nowej wartości określonego współczynnika). Gdyby program usiłował ustawić współczynnik przy najwyższej potęgze na 0, to należy zgłosić wyjątek (za wyjątkiem sytuacji, gdy stopień wielomianu jest równy 0).

```
class wielomian
{
    int n; // stopień wielomianu
    double *a; // współczynniki wielomianu
    // ...
};
```

W klasie `wielomian` zdefiniuj konstruktor bezargumentowy, konstruktor z listą współczynników, zaimplementuj kopiowanie i przenoszenie, przypisanie kopiujące i przenoszące oraz destruktor.

```
class wielomian
{
public:
    wielomian (int st=0, double wsp=1.0); // konstruktor tworzący jednomian
    wielomian (int st, double wsp[]); // konstruktor tworzący wielomian
    wielomian (initializer_list<double> wsp); // lista współczynników
    wielomian (const wielomian &w); // konstruktor kopiujący
    wielomian (wielomian &&w); // konstruktor przenoszący
};
```

```

        wielomian& operator = (const wielomian &w); // przypisanie kopijące
        wielomian& operator = (wielomian &&w); // przypisanie przenoszące
        ~wielomian (); // destruktor
        // ...
};

```

Nie zapomnij też o operatorach czytania ze strumienia wejściowego i pisania do strumienia wyjściowego. Operatory te powinny się przyjaźnić z klasą `wielomian`.

```

class Wielomian
{
    friend istream& operator >> (istream &we, wielomian &w);
    friend ostream& operator << (ostream &wy, const wielomian &w);
    // ...
};

```

Do kompletu podefiniuj operatory dodawania i odejmowania wielomianów, operator mnożenia wielomianu przez stałą i mnożenia przez inny wielomian (możesz do kompletu zdefiniować operatory dzielenia wielomianów) oraz operator wywołania funkcji obliczający wartość wielomianu w zadanym punkcie za pomocą *schematu Hornera*. Operatory dodawania, odejmowania i mnożenia wielomianów niech będą operatorami zaprzyjaźnionymi, które zwracają wynik przez wartość. Dodaj jednak składowe operatory przypisania połączone ze wspomnianymi operatorami arytmetycznymi, które będą zwracały jako wynik referencję do bieżącego obiektu.

Na koniec napisz program, który bardzo rzetelnie przetestuje całą funkcjonalność zaprogramowaną w klasie `wielomian`. Dane do programu wczytaj ze standardowego wejścia `cin` za pomocą operatora strumieniowego `>>`. Wyniki wypisz na standardowym wyjściu `cout` za pomocą operatora strumieniowego `<<`. Ewentualne komunikaty o błędach wypisz na standardowym wyjściu dla błędów `cerr`.

Uwaga.

Podziel program na pliki nagłówkowe i źródłowe.