

kurs języka C++

wielomiany

Instytut Informatyki
Uniwersytetu Wrocławskiego

Paweł Rzechonek

Prolog

Wielomian to wyrażenie algebraiczne będące sumą jednomianów. Wielomiany, ze względu na swoją prostotę i dobrze poznane własności, są używane w wielu działach matematyki. 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 $a_n \neq 0$.

Zadanie

Zdefiniuj klasę `wielom` do przechowywania wielomianu o określonym stopniu oraz określonych współczynnikach. Zaprojektuj tę klasę tak, aby stopień wielomianu oraz jego współczynniki były niepubliczne (współczynniki pamiętaj w tablicy utworzonej na stercie). 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 odczytania i inny do wpisania nowej wartości określonego współczynnika). Gdyby program usiłował ustawić współczynnik przy najwyższej potędze na 0, to należy zgłosić wyjątek (za wyjątkiem sytuacji, gdy stopień wielomianu jest równy 0).

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

Współczynnik a_i niech będzie pamiętany na pozycji i -tej w tablicy `a` (czyli w komórce `a[i]`) dla $i = 0 \dots n$.

W klasie `wielom` zdefiniuj konstruktor bezargumentowy, konstruktor z listą inicjalizującą (ze współczynnikami), zaimplementuj kopiowanie i przenoszenie (dotyczy konstruktorów i operatorów przypisania) oraz destruktor (usunięcie tablicy współczynników wielomianu).

```

class wielom {
public:
    wielom (int st=0, double wsp=1.0); // konstruktor jednomianu
    wielom (int st, const double wsp[]); // konstruktor wielomianu
    wielom (initializer_list<double> wsp); // lista współczynników
    wielom (const wielom &w); // konstruktor kopiujący
    wielom (wielom &&w); // konstruktor przenoszący
    wielom& operator = (const wielom &w); // przypisanie kopiujące
    wielom& operator = (wielom &&w); // przypisanie przenoszące
    ~wielom (); // destruktor
    // ...
};

```

Nie zapomnij też o operatorze odczytu wielomianu ze strumienia oraz zapisu wielomianu do strumienia. Operatory te mogą przyjaźnić się z klasą wielomianu, aby mieć bezpośredni dostęp do współczynników.

```

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

```

Zdefiniuj operatory dodawania i odejmowania wielomianów, operator mnożenia wielomianu przez stałą i mnożenia przez inny wielomian (możesz do kompletu zdefiniować operatory dzielenia wielomianów, jeśli chcesz uzupełnić definicję, ale to nie jest wymagane) 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 także składowe operatory przypisania połączone ze wspomnianymi operatorami arytmetycznymi, które będą zwracały jako wynik referencję do bieżącego obiektu reprezentującego wielomian.

```

class wielom {
public:
    friend wielom operator + (const wielom &u, const wielom &v);
    friend wielom operator - (const wielom &u, const wielom &v);
    friend wielom operator * (const wielom &u, const wielom &v);
    friend wielom operator * (double c);
    wielom& operator += (const wielom &v);
    wielom& operator -= (const wielom &v);
    wielom& operator *= (const wielom &v);
    wielom& operator *= (double c);
    double operator () (double x) const; // wartość wielomianu dla x
    double operator [] (int i) const; // do odczytu współczynnika
    double& operator [] (int i); // do zapisu współczynnika
    // ...
};

```

Na koniec napisz program, który bardzo rzetelnie przetestuje całą funkcjonalność zaprogramowaną w klasie reprezentującej 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 (definicja klasy) i źródłowe (definicje metod zadeklarowanych w klasie). Funkcję `main()` z testami umieść w osobnym pliku źródłowym.

Ważne elementy programu

- Implementacja semantyki kopiowania i przenoszenia dla wielomianu.
- Definicje operatorów dla wielomianu (dodawanie, odejmowanie i mnożenie).
- Zgłaszanie wyjątków w konstruktorach i funkcjach składowych.
- Przetestowanie klasy `wielomian` w funkcji `main()`.