

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 `wielomian` do przechowywania wielomianu o określonym stopniu. 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
{
private:
    int n; // stopień wielomianu
    double *a; // współczynniki wielomianu
    // ...
};
```

Pierwszym współczynnikiem `a[0]` niech będzie a_0 .

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, const double wsp[]); // konstruktor tworzący wielomian
    wielomian (initializer_list<double> wsp); // lista współczynników
    wielomian (const wielomian &w); // konstruktor kopijący
    wielomian (wielomian &&w); // konstruktor przenoszący
    wielomian& operator = (const wielomian &w); // przypisanie kopijące
    wielomian& operator = (wielomian &&w); // przypisanie przenoszące
    ~wielomian (); // destruktor
    // ...
};

```

W destruktorze zwolnij pamięć przydzieloną do pamiętania współczynników wielomianu.

Nie zapomnij też o operatorach czytania ze strumienia wejściowego i pisanie do strumienia wyjściowego.

```

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

```

Operatory te powinny się przyjaźnić z klasą `wielomian`.

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, ale to tylko w przypadku jak będziesz miał czas na dodatkowe eksperymenty) 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.

```

class wielomian
{
public:
    friend wielomian operator + (const wielomian &u, const wielomian &v);
    friend wielomian operator - (const wielomian &u, const wielomian &v);
    friend wielomian operator * (const wielomian &u, const wielomian &v);
    friend wielomian operator * (double c);
    wielomian& operator += (const wielomian &v);
    wielomian& operator -= (const wielomian &v);
    wielomian& operator *= (const wielomian &v);
    wielomian& operator *= (double c);
    double operator () (double x) const; // obliczenie wartości wielomianu dla x
    double operator [] (int i) const; // odczytanie i-tego współczynnika
    // ...
};

```

Wielomian powinien umożliwiać łatwe odczycanie współczynników stojących przy jednomianach za pomocą operatora indeksowania.

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`.

Elementy w programie, na które należy zwracać uwagę.

- Podział programu na pliki nagłówkowe i pliki źródłowe (wyodrębniony osobny plik z funkcją `main()`).
- Implementacja semantyki kopiowania i przenoszenia dla wielomianu.
- Definicje operatorów dla wielomianu (dodawanie, odejmowanie i mnożenie).
- W funkcji `main()` należy przetestować wszystkie funkcje składowe i operatory związane z wielomianem.