Klasówka z programowania obiektowego i C++

Zadanie

Zaprojektuj w C++ klasę Wielomian. Obiekty tej klasy powinny reprezentować wielomiany (o współczynnikach rzeczywistych) jednej zmiennej rzeczywistej x. Obiekty tej klasy powinny udostępniać następujące operacje:

- *wartosc*: policzenie wartości w punkcie,
- *pochodna*: obliczenie swojej pochodnej (efektem tej operacji jest wygenerowanie nowego wielomianu, obiekt dla którego wykonano tę operację nie zmienia swojej wartości),
- *dodaj*: dodawanie do siebie drugiego wielomianu (efektem tej operacji jest zmiana wartości tego obiektu, dla którego wykonano tę operację).

Ponadto należy zdefiniować odpowiedni zestaw konstruktorów oraz destruktor. Wśród konstruktorów powinien być (co najmniej) konstruktor bezargumentowy, kopiujący oraz konstruktor inicjujący nowy wielomian na podstawie tablicy współczynników typu double i rozmiaru tej tablicy. Współczynnik w elemencie tablicy o indeksie k odpowiada współczynnikowi przy x do potęgi k. Kopiujący operator przypisania (choć tu potrzebny), nie jest w tym zadaniu wymagany.

Klasa Wielomian ma pamiętać wielomian w postaci uporządkowanej listy następujących struktur:

```
struct Elt {
    double wart;
    int wykl;
    Elt *nast;
    Elt(double, int, Elt *);
    Elt(const Elt &);
};
Elt::Elt(double wart, int wykl, Elt *nast) :
   wart(wart), wykl(wykl), nast(nast) {
}
Elt::Elt(const Elt &elt) : wart(elt.wart), wykl(elt.wykl), nast(elt.nast)
{}
```

To oznacza, że inna implementacja (np. w tablicy, czy wektorze), nie będzie zaliczana. Struktury Elt nie trzeba już implementować w swoim rozwiązaniu.
Dla przypomnienia: przejście do następnego elementu listy zapisuje się tak:

```
*Elt p;
// ... m.in. inicjalizacja p ...
p = p->nast;  // przejście do następnego elementu listy
```

Życzymy powodzenia!

Uwagi:

- Zamianę liczby (typu int, double, itp.) na napis można zapisać tak: std::to_string(liczba).
- Podnoszenie liczby do potęgi wykonuje funkcja pow(double wartość, int wykładnik) z modułu cmath. W tym zadaniu można się łatwo bez niej obejść, obie wersje (z pow i własnym liczeniem potęgi) będą tak samo oceniane.

Załącznik – przykład działania programu:

Frgament programu:

```
WielomianLst w0;
double tab1[1] = {1.0};
double tab2[3] = {1.0, -3.0, 5.0};
double tab3[4] = {1.0, 0.0, 0.0, 6.0};
WielomianLst w1(tab1, 1);
WielomianLst w2(tab2, 3);
WielomianLst w3(tab3, 4);
WielomianLst w4(w3);
cout << "Wartość Wielomianu "; w0.wypisz(); cout << " w punkcie 2.0: " <<
w0.wartosc(2.0) << endl;
cout << "Wartość Wielomianu "; w1.wypisz(); cout << " w punkcie 2.0: " <<
w1.wartosc(2.0) << endl;
cout << "Wartość Wielomianu "; w2.wypisz(); cout << " w punkcie 2.0: " <<
w2.wartosc(2.0) << endl;
cout << "Wartość Wielomianu "; w3.wypisz(); cout << " w punkcie 2.0: " <<
w3.wartosc(2.0) << endl;
cout << "Wartość Wielomianu "; w4.wypisz(); cout << " w punkcie 2.0: " <<
w4.wartosc(2.0) << endl;
cout << "Wielomian "; w2.wypisz(); cout << " po dodaniu Wielomianu ";
w4.wypisz(); cout << " to: "; w2.dodaj(w4).wypisz(); cout << endl;
cout << "Wielomian "; w3.wypisz(); cout << " po dodaniu Wielomianów ";
w1.wypisz(); cout << " oraz "; w2.wypisz(); cout << " to: ";
w3.dodaj(w1).dodaj(w2).wypisz(); cout << endl;
cout << "Pochodna Wielomianu "; w1.wypisz(); cout << " to: ";
w1.pochodna().wypisz(); cout << endl;
cout << "Pochodna Wielomianu "; w2.wypisz(); cout << " to: ";
w2.pochodna().wypisz(); cout << endl;
cout << "Pochodna Wielomianu "; w3.wypisz(); cout << " to: ";
w3.pochodna().wypisz(); cout << endl;
cout << "Pochodna Wielomianu "; w4.wypisz(); cout << " to: ";
w4.pochodna().wypisz(); cout << endl;
```

powinien wypisać (z dokładnością do formatu liczb rzeczywistych):

```
Wartość Wielomianu [] w punkcie 2.0: 0
Wartość Wielomianu [1*x^0] w punkcie 2.0: 1
Wartość Wielomianu [1*x^0-3*x^1+5*x^2] w punkcie 2.0: 15
Wartość Wielomianu [1*x^0+6*x^3] w punkcie 2.0: 49
Wartość Wielomianu [1*x^0+6*x^3] w punkcie 2.0: 49
Wielomian [1*x^0-3*x^1+5*x^2] po dodaniu Wielomianu [1*x^0+6*x^3] to: [2*x^0-3*x^1+5*x^2+6*x^3]
Wielomian [1*x^0+6*x^3] po dodaniu Wielomian├┤w [1*x^0] oraz [2*x^0-3*x^1+5*x^2+6*x^3] to:
[4*x^0-3*x^1+5*x^2+12*x^3]
Pochodna Wielomianu [1*x^0] to: []
Pochodna Wielomianu [2*x^0-3*x^1+5*x^2+6*x^3] to: [-3*x^0+10*x^1+18*x^2]
Pochodna Wielomianu [4*x^0-3*x^1+5*x^2+12*x^3] to: [-3*x^0+10*x^1+36*x^2]
Pochodna Wielomianu [1*x^0+6*x^3] to: [18*x^2]
```