

Szoftver laboratórium 2. 3. ellenőrző dolgozat. 2012.03.27. Kurz/Terem: L4/	15 perc
Név:	Neptun:
	Összpont:

1.feladat

4.5+1.5 pont

Valósítson meg C++ nyelven egy olyan osztályt (*Polinomok*), ami tetszőleges számú *Polinom* típusú objektumot képes tárolni! Tételezze fel, hogy a *Polinom* osztály létezik, és másodfokú polinom együtthatóit tárolja! A *Polinomok* osztályt úgy valósítsa meg, hogy másolható legyen és értékadás bal és jobb oldalán is szerepelhessen. A függvényhívás operátora számítsa ki a tárolt polinomok helyettesítési értékek összegét a paraméterként kapott helyen.

Megadtuk a *Polinomok* osztály deklarációját, és a tagfüggvények definícióját, de ez több helyen hiányos. A **pontozott** részekre írva **egészítse ki** az alábbi kódrészletet úgy, hogy a megadott főprogram az elvárásoknak megfelelően működjön, és ne lépjen fel memóriakezelési hiba! Nem kell minden pontozott részre írnia! Feltételezheti, hogy a **másoló konstruktor**, amely külön állományban van megvalósítva, **helyesen működik**, ezért azt **nem kell** megírnia!

A feladatlap hátoldalán **adja meg**, hogy **megoldása** minimálisan **milyen elvárásokat támaszt** a *Polinom* osztállyal szemben! (pl: van olyan konstruktora, ami 3 valós számot kap paraméterként; van ... operátora; stb.) (1.5p)

```
class Polinomok {
    Polinom* pPoli; // dinamikus adatterület kezdőcíme, ahol tárolunk
    int db; // tárolt Polinomok száma (pontosan ennyi polinom van )
public:
    // Konstruktor: n darab valós számhármast (3*n darab számot) kap, amiből
    // n darab Polinom objektumot készít és letárolja
    Polinomok(const double a[], int n = 0) .....:db(n)..... {
        pPoli = new Polinom[db];
        for (int i = 0; i < 3*db; i += 3)
            pPoli[i/3] = Polinom(a[i], a[i+1], a[i+2]);
    }
    .....Polinomok(const Polinomok&) .....;
    Polinomok& operator=(const Polinomok&) .....;
    double operator()(double x) const;
    .....~Polinomok() { delete [] pPoli; }
};

Polinomok& Polinomok::operator=(const Polinomok& p) ..... {
    if (this != &p) {
        delete[] pPoli;
        db = p.db;
        pPoli = new Polinom[db];
        for (int i = 0; i < db; i++)
            pPoli[i] = p.pPoli[i];
    }
    return *this;
}

double Polinomok::operator()(double x) const {
    double sum = 0;
    for (int i = 0; i < db; i++)
        sum += pPoli[i](x);
    return sum;
}

int main() {
    double av[] = { 1, 2, 3, 4, 5, 6, 7, 8, 9 };
    Polinomok p1(av, 3); // három darab valós számhármast (9 darab szám)
    Polinomok p2 = p1;
    cout << p1(1); // helyettesítés értékek összege az 1 helyen
    p2 = p1 = p1;
    cout << p2(3.14); // helyettesítés értékek összege a 3.14 helyen
}
```

**Pont osztállyal szemben támasztott elvárások:**

- legyen létrehozható 3 db valós számból
- legyen default konstruktora
- legyen értékadás (operator=) operátora
- legyen függvényhívás operátora, ami a helyettesítési értéket számolja