

Szoftver laboratórium 2.	3. Ellenőrző dolgozat.	2014.04.01.	Kurz/Terem: L4/
Név:	Neptun:	Összpont:	

1.feladat

2 pont

Tervezzen egy olyan osztályt (*Ellenallas2*), ami két ellenállást tartalmaz és képes azok soros, és párhuzamos kapcsolásával nyert eredő ellenállását kiszámolni. A konstruktorban legyen megadható mindkét ellenállás értéke! Ön döntheti el, hogy a paraméter nélkül hívott konstruktor milyen értéket állít be. A soros és a párhuzamos eredő a *getS* és a *getP* tagfüggvényekkel legyen lekérdezhető! Az ellenállások tárolásához használja fel a korábban házi feladatként elkészített *Resistor* osztályt! Osztálydiagram segítségével ábrázolja a két osztály kapcsolatát! A *Resistor* osztály tagfüggvényeit és adattagjait nem kell részleteznie a rajzon. (emlékeztető: soros kapcs: op+, párhuzamos kapcs: op%)

Valósítsa meg C++ nyelven az *Ellenallas2* osztályt!

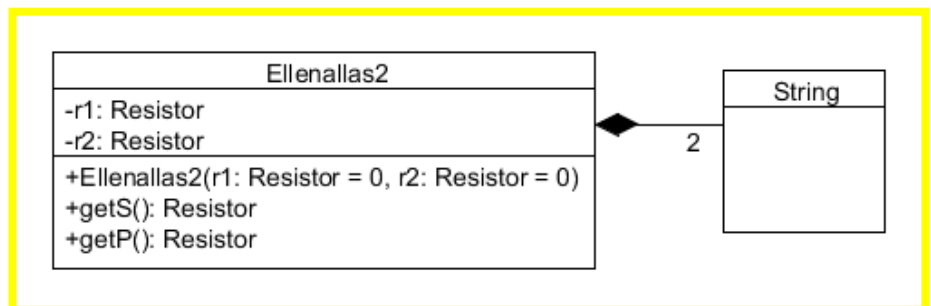
2.feladat

4 pont

Tervezzen egy olyan tároló jellegű osztályt (*Doboz*), ami tetszőleges mennyiségű *Ellenallas2* típusú objektumot képes tárolni! Az osztálynak legyen olyan tagfüggvénye, amivel új *Ellenallas2* példányt rakhathunk a dobozba (*berak*), és olyan is, amivel lekérdezhetjük, a dobozban levő *Ellenallas2* objektumok számát (*getN*)! Elvárás az osztállyal szemben, hogy másolható legyen, értékadás bal és jobb oldalán is szerepelhessen, valamint támogassa a többszörös értékadást is.

Deklarálja C++ nyelven a *Doboz* osztályt! Valósítsa meg (definiálja) az osztály paraméter nélkül hívható konstruktorát, destruktort, valamint az *berak* tagfüggvényét! Az alábbi kódrészlet az osztályok használatára mutat példát:

```
....
Ellenallas2 kicsi(5, 8), nagy;
Doboz d1, d2, d3;
d1.berak(nagy);
d1.berak(kicsi);
d2 = d3 = d1;
std::cout << d3.getN();
```



```
class Ellenallas2 {
    Resistor r1, r2;
public:
    Ellenallas2(double r1 = 0, double r2 = 0) :r1(r1), r2(r2) {}
    Resistor getS() const { return r1 + r2; }
    Resistor getP() const { return r1 % r2; }
};
```

```
class Doboz {
    Ellenallas2 *rp;
    int db;
public:
    Doboz() :db(0), rp(0) {}
    Doboz(const Doboz&);
    int getN() const;
    void berak(const Ellenallas2&);
    Doboz& operator=(const Doboz&);
    ~Doboz() { delete[] rp; }
};
```

```
void Doboz::berak(const Ellenallas2& r) {
    Ellenallas2 *tmp = new Ellenallas2[db+1];
    for (int i = 0; i < db; i++)
        tmp[i] = rp[i];
    delete[] rp;
    rp = tmp;
    rp[db++] = r;
}
```