

Programozás alapjai 2.	3. ellenőrző dolgozat.	2014.04.09.	Kurz/Terem: G2/
Név:	Neptun:	Összpont:	

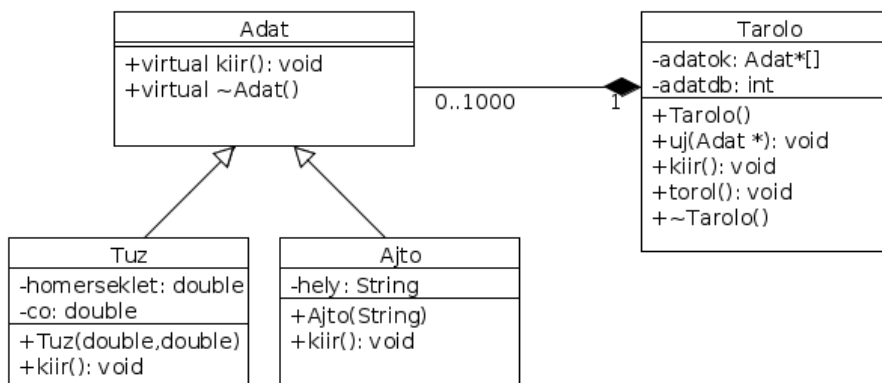
1. feladat

2.5 pont

Egy biztonsági rendszerben különféle érzékelők adatait (*Adat*) kell fogadni és eltárolni. Az érzékelők teljesen eltérő adatokat adnak. A tűzjelző (*Tuz*) pl. hőfokot és CO koncentrációt (2 db *double*), az ajtónyitás érzékelő (*Ajto*) pedig egy szöveget (*String*) ad, ami az érzékelő helyét azonosítja. A rendszert később bővíteni fogjuk újabb, jelenleg még nem ismert érzékelőkkel. Követelmény, hogy a beérkező adatok a beérkezés sorrendjében, egyetlen tárolóban (*Tarolo*) legyenek eltárolva. A tárolóban levő adatokat (azok attribútumait is) naponta egy *std::ostream* típusú objektumra kiírják (*kiir*), majd törlik (*torol*). Egy nap maximum 1000 adat keletkezik. A rendszerben megvalósítandó műveleteket egy kódrészlettel mutatjuk be:

```
Tarolo naplo; // Ez lesz a tároló
naplo.uj(new Tuz(98, 0.55)); // Tűzjelző adatainak letárolása és felvétele
naplo.uj(new Ajto("IB28")); // Ajtónyitás érzékelő adatának hozzáadása
naplo.kiir(std::cout); // Tárolóban levő adatok kiírása
naplo.torol(); // Összes adat törlése a nap végén
naplo.uj(new Ajto("IB26")); // Ajtónyitás érzékelő adatának hozzáadása
```

Feltételezheti, hogy a *Tarolo* osztályból példányosított objektumot nem akarjuk paraméterként átadni és értékadás jobb, ill. bal oldalán sem szerepel. **Tervezzen** meg és **rajzoljon** fel egy olyan osztályhierarchiát, ami alkalmas a biztonsági rendszer adatainak keletkezési sorrendben való tárolására és könnyen bővíthető! Az osztálydiagramban jelölje az adattagok és metódusok láthatóságát is, valamint a virtuális függvényeket is! A *String* osztályt nem kell lerajzolni, arra típusként hivatkozzon!



2. feladat

3.5 pont

Deklarálja a *Tarolo*, *Adat*, és *Ajto* osztályokat! **Csak** a *Tarolo* osztály konstruktorát, valamint az *uj()*, *torol()* és *kiir()* metódusát **valósítsa** meg!

```
class Adat {
public:
    virtual void kiir();
    virtual ~Adat();
};
```

```
class Ajto : public Adat {
public:
    String hely;
    Ajto(String);
    void kiir();
};
```

```
class Tarolo {
    Adat* adatok[1000];
    int adatdb;
public:
    Tarolo() : adatdb(0) {}
    void uj(Adat* a) {
        adatok[adatdb++] = a;
    }
    void kiir() {
        for (int i = 0; i < adatdb; i++)
            adatok[i]->kiir();
    }
    void torol() {
        for (int i = 0; i < adatdb; i++)
            delete adatok[i];
        adatdb = 0;
    }
    ~Tarolo();
};
```