

Szoftver laboratórium 2. 2. ellenőrző dolgozat. 2012.03.13. Kurz/Terem: L4/	15 perc
Név:	Neptun:
	Összpont:

1.feladat

6 pont

Valósítson meg C++ nyelven egy olyan osztályt (**Datum**), ami hónap, nap formátumban tárolja a dátumot! A konstruktor ellenőrizze, hogy a paraméterként megadott nap érvényes-e. Ha nem lehetséges a megadott nap a megadott hónapban, akkor a hónap utolsó napját állítsa be! Tételezze fel, hogy létezik egy **Honapok** nevű osztály, aminek az index operátora (operator[](int)) megmondja, hogy a paraméterként kapott hónap hány napos.

Az osztály valósítsa meg a következő műveleteket!

- Két **Datum** összehasonlítása az == művelettel (operator==()).
- hónap és nap kiírása egy ostream típusú objektumra a szokásos << operátorral. A kiírás formátumát Ön határozza meg, és ezt adja meg az alábbi kódrészletben. (A vonalakra írja rá, hogy mit ír ki! Jelölje a szóközőket is!)

Az adattagok kívülről közvetlenül ne legyenek elérhetőek! Működjön helyesen az alábbi kódrészlet! **Ne írjon felesleges** tagfüggvényt (olyat, ami automatikusan keletkezik és jó a feladathoz)!

```
const Datum d0; // hónap=1, nap=1
const Datum d0401(4); // hónap=4, nap=1
const Datum d1111(11, 11); // hónap=11, nap=11
const Datum d1130(11, 52); // hónap=11, nap=30
Datum* np = new Datum[5];
np[0] = d0;
np[1] = d1130;
if (np[0] == Datum())
    std::cout << np[1] << std::endl; // 11.30.
```

A megfelelő utasítások után a vonalra írja rá, hogy **mit** ír ki a fenti kódrészlet!

Hányszor hívódott meg a fenti kódrészletben a **paraméter nélküli** konstruktor¹? Csak azután válaszoljon, miután megírta a saját kódját! **7**

```
class Datum {
    int ho, nap;
public:
    Datum(int h = 1, int n = 1) :ho(h), nap(n) {
        Honapok honap;
        if (nap > honap[ho]) nap = honap[ho];
    }
    bool operator==(const Datum& const);
    friend ostream& operator<<(ostream&, const Datum&);
};
bool Datum::operator==(const Datum& d) const {
    return ho == d.ho && nap == d.nap;
}
std::ostream& operator<<(ostream& os, const Datum &d) {
    os << setfill('0') << setw(2) << d.ho << '.'
        << setfill('0') << setw(2) << d.nap << '.';
    return os;
}
```

¹ A kérdés a paraméter nélkül **hívható** (default) konstruktorra vonatkozott. Paraméter nélküli konstruktora az osztálynak valójában nincs, mivel ott vannak a default argumentumok. Aki ez utóbbi formában értelmezte a kérdést annak a 0-át is elfogadtuk válaszként.