

Programozás alapja 2.	2. ellenőrző dolgozat.	2015.03.25.	Kurz/Terem: G2/	Elért pontszám:
Név:			Neptun:	

1. Egy útvonal-nyilvántartó programban méterben és yard-ban megadott távolságokat is kezelünk kell. Ehhez **tervezzen** egy olyan osztályt (*Tavolsag*), amelynek a konstruktorában megadható, hogy az adott objektumpéldány mely rendszerben (metrikus/angolszász) működik, azaz a getter/setter függvényei mit (méter/yard) várnak, illetve mit adnak. Legyen az osztálynak 7p

- olyan tagfüggvénye, amivel a távolság beállítható, ill. lekérdezhető (*get/set*);
- olyan tagfüggvénye, amivel mindig méterben kérdezhető le a távolság;
- olyan tagfüggvénye, amivel lekérdezhető, hogy metrikus rendszerű-e az objektum (*is_metric*);
- olyan tagfüggvénye, amivel meg lehet állapítani egy távolságról, hogy kisebb-e mint a másik (<); Az összehasonlítás működjön helyesen még akkor is, ha az összehasonlítás bal és jobb oldalán eltérő rendszerben működő objektumok állnak (1 yard 0,9144 méternek felel meg)!

Valósítsa meg az osztályt C++ nyelven úgy, hogy ne lehessen az adattagokhoz kívülről hozzáférni, és az osztálynak legyen legalább egy nem inline tagfüggvénye! Működjön az alábbi kódrészlet!

```
const bool metrikus = true;
const bool angolszasz = false;
Tavolsag m1(metrikus), a1(angolszasz);
m1.set(10.1); a1.set(11);
std::cout << std::boolalpha << m1.is_metric(); // true-t ír ki
std::cout << std::boolalpha << (m1 < a1); // false-t ír ki
```

2. Az előző feladatban megtervezett osztályhoz készítsen inserter operátort (<<), ami képes a tárolt távolságot kiírni egy ostream típusú objektumra! Méter esetén a kiírt érték után „m” betűt, yard esetében pedig „yd” karaktersorozatot írjon! Rövid kódrészleten mutassa be az inserter használatát! 3p

megoldás:

```
class Tavolsag {
    bool metrikus;
    double tav;
    static const double yd2m = 0.9144;
public:
    Tavolsag(bool m) :metrikus(m) {}
    bool is_metric() const { return metrikus; }
    double get() const { return tav; }
    double getMeter() const { return metrikus ? tav : tav * yd2m; }
    void set(double d) { tav = d; }
    bool operator<(const Tavolsag& d) const;
};
bool Tavolsag::operator<(const Tavolsag& d) const {
    return getMeter() < d.getMeter();
}
std::ostream& operator<<(std::ostream& os, const Tavolsag& d) {
    os << d.get() << (d.is_metric() ? "m" : "yd");
    return os;
}
std::cout << m1 << std::endl << a1 << std::endl;
```

Programozás alapja 2.	1. ellenőrző dolgozat.	2015.03.35.	Kurz/Terem: G2/	Elért pontszám:
Név:			Neptun:	

1. Egy adatgyűjtő programban Celsius-fokban és Fahrenheit-fokban megadott hőmérsékleteket is kezelni kell. Ehhez **tervezzen** egy olyan osztályt (*Homerseklet*), amelynek a konstruktorában megadható, hogy az adott objektumpéldány mely skálán (Celsius/Fahrenheit) működik, azaz a getter/setter függvényei mit várnak, illetve mit adnak. Legyen az osztálynak

7p

- olyan tagfüggvénye, amivel a hőmérséklet beállítható, ill. lekérdezhető (*get/set*);
- olyan tagfüggvénye, amivel mindig Celsius-fokban kérdezhető le a hőmérséklet;
- olyan tagfüggvénye, amivel megkérdezhető, hogy Celsius skálájú-e az objektum (*is_celsius*);
- olyan tagfüggvénye, amivel meg lehet állapítani egy hőmérsékletről, hogy nagyobb-e mint a másik (>); Az összehasonlítás működjön helyesen még akkor is, ha az összehasonlítás bal és jobb oldalán eltérő skálán működő objektumok állnak ($X \text{ } ^\circ\text{F} = \frac{[(X - 32) \times 5]}{9} \text{ } ^\circ\text{C}$)!

Valósítsa meg az osztályt C++ nyelven úgy, hogy ne lehessen az adattagokhoz kívülről hozzáférni, és az osztálynak legyen legalább egy nem inline tagfüggvénye! Működjön jól az alábbi kódrészlet:

```
const bool celsius = true;
const bool fahrenheit = false;
Homerseklet c1(celsius), f1(fahrenheit);
c1.set(10); f1.set(32);
std::cout << std::boolalpha << c1.is_celsius(); // true-t ír ki
std::cout << std::boolalpha << (c1 > f1); // true-t ír ki
```

2. Az előző feladatban megtervezett osztályhoz készítsen inserter operátort (<<), ami képes a tárolt hőmérsékletet kiírni egy ostream típusú objektumra! Celsius esetén a kiírt érték után „C” betűt, Fahrenheit esetében pedig „F” betűt írjon!

Egy rövid kódrészleten mutassa be az inserter használatát!

3p

megoldás:

```
class Homerseklet {
    const bool celsius;
    double ho;
public:
    Homerseklet(bool m) :celsius(m) {}
    bool is_celsius() const { return celsius; }
    double get() const { return ho; }
    double getCels() const { return celsius ? ho : (ho-32) * 5 / 9; }
    void set(double t) { ho = t; }
    bool operator>(const Homerseklet& t) const;
};

bool Homerseklet::operator>(const Homerseklet& t) const {
    return getCels() > t.getCels();
}

std::ostream& operator<<(std::ostream& os, const Homerseklet& d) {
    os << d.get() << (d.is_celsius() ? "C" : "F");
    return os;
}

std::cout << c1 << std::endl << f1 << std::endl;
```