

Programozás alapjai II. 4. ellenőrző dolgozat. 2013.04.18. Kurz/Terem: G1/	20 perc
Név:	Összpont:
Neptun:	

1. feladat

3.5 pont

Készítsen sablont kulcs-érték párok tárolására (*KulcsErtek*)! Mind a kulcs, mind az érték típusa legyen sablonparaméterként megadható! Minden adat legyen külön-külön lekérdezhető és beállítható, de közvetlenül egyik adat se legyen elérhető! Legyen összehasonlítható két ilyen generikus adat az == operátorral! Két *KulcsErtek* típusú adat akkor azonos, ha adataik azonosak. Működjön az elvárásoknak megfelelően az alábbi kódrészlet!

```
KulcsErtek<string, int> par4("4.kZH", 5);
par4.ertek() += 1; // megnöveli az értéket
std::cout << par4.kulcs() << ": " << par4.ertek() << std::endl; //
KulcsErtek<int, double> parok[100], minta; // 2. feladatban ezeket kell használni
parok[0].kulcs() = 12; // beállítja a kulcsot
if (parok[0] == minta ) std::cout << "azonos" << std::endl;
```

```
template<class T1, class T2>
class KulcsErtek {
    T1 _kulcs;
    T2 _ertek;
public:
    KulcsErtek(const T1& k = T1(), const T2& e = T2()) :_kulcs(k), _ertek(e) {}
    T1& kulcs() { return _kulcs; }
    T2& ertek() { return _ertek; }
    bool operator==(const KulcsErtek& rhs) const {
        return _kulcs == rhs._kulcs && _ertek == rhs._ertek;
    }
};
```

2. feladat

2.5 pont

Készítsen függvénysablont, ami iterátorokkal megadott adatsorozatban megkeres egy adott értéket! A függvény paraméterként vegye át a két iterátort és a keresett értéket. A függvény visszatérési értéke a megtalált elemre, vagy a sorozat végére (utolsó elem után) mutató iterátor legyen! (1.5p) **Mutassa** be egy kódrészlettel, hogy a függvénysablonnal hogyan lehet megkeresni az előző feladat *parok* tömbjében a *minta* értékpárt! (1p)

```
template<class Iter, class T>
Iter keres(Iter elso, Iter utolso, const T& ertek) {
    while (elso != utolso) {
        if(*elso == ertek) return elso;
        ++elso;
    }
    return utolso;
}

if (keres(parok, parok+100, minta) == parok+100)
    cout << "nincs ilyen" << endl;
```

BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB

Programozás alapjai II. 4. ellenőrző dolgozat. 2013.04.18. Kurz/Terem: G1/	20 perc
Név:	Neptun:
Összpont:	

1. feladat

3.5 pont

Készítsen sablont név-jelszó párok tárolására (*NevJelszo*)! Mind a név, mind az jelszó típusa legyen sablonparaméterként megadható! Minden adat legyen külön-külön lekérdezhető és beállítható, de közvetlenül egyik adat se legyen elérhető! Legyen összehasonlítható két ilyen generikus adat az == operátorral! Két *NevJelszo* típusú adat akkor azonos, ha adataik azonosak. Működjön az elvárásoknak megfelelően az alábbi kódrészlet!

```

NevJelszo<string, string> login("user1", "123456");
login.jelszo() = "titok"; // beállítja a jelszót
std::cout << login.nev() << " : " << login.jelszo() << std::endl;
NevJelszo<long, int> jelszavak[100], rootpw; // 2. feladatban ezeket kell használni
jelszavak[0].nev() = 12; // beállítja a nevet
if (jelszavak[0] == rootpw ) std::cout << "azonos" << std::endl;
    
```

```

template<class T1, class T2>
class NevJelszo {
    T1 _nev;
    T2 _jelszo;
public:
    NevJelszo(const T1& n = T1(), const T2& j = T2()) :_nev(n), _jelszo(j) {}
    T1& nev() { return _nev; }
    T2& jelszo() { return _jelszo; }
    bool operator==(const NevJelszo& rhs) const {
        return _nev == rhs._nev && _jelszo == rhs._jelszo;
    }
};
    
```

2. feladat

2.5 pont

Készítsen függvénysablont, ami iterátorokkal megadott adatsorozatban megkeres egy adott értéket. A függvény paraméterként vegye át a két iterátort és a keresett értéket. A függvény visszatérési értéke a megtalált elemre, vagy a sorozat végére (utolsó elem után) mutató iterátor legyen! (2p) **Mutassa** be egy kódrészlettel, hogy a függvénysablonnal hogyan lehet megkeresni az előző feladat *jelszavak* tömbjében a *rootpw* név-jelszó párt! (1p)

```

template<class Iter, class T>
Iter keres(Iter elso, Iter utolso, const T& ertek) {
    while (elso != utolso) {
        if(*elso == ertek)
            break;
        ++elso;
    }
    return elso;
}

if (keres(jelszavak, jelszavak+100, rootpw) == jelszavak+100)
    cout << "nincs ilyen" << endl;
    
```