

Programozás alapjai II. 5. ellenőrző dolgozat. 2013.05.02. Kurz/Terem: G3/	20 perc
Név:	Összpont:
Neptun:	

1. feladat

3 pont

Tételezze fel, hogy rendelkezésére áll egy verem tulajdonságokkal rendelkező generikus tároló (*Stack*)! A tároló a szokásos verem műveletekkel rendelkezik (*push*, *pop*, *top*, *empty*), és csak default konstruktora van. Ezen osztály **felhasználásával** készítsen egy olyan generikus vermet (*MyStack*), ami a *Stack* osztállyal **teljesen azonos** módon viselkedik, de van egy *reverse()* tagfüggvénye is, ami megfordítja az elemek sorrendjét a veremben!

**Örökléssel egyszerűbb, de tartalmazott objektummal is megoldható volt.**

Ez utóbbi esetben a *Stack* osztály minden függvényét delegálni kellett.

```
template <typename T>
class MyStack :public Stack<T> {
public:
    void reverse() {
        MyStack<T> tmp;
        while (!Stack<T>::empty()) {
            tmp.push(this->top());
            this->pop();
        }
        *this = tmp;
    }
};
```

2. feladat

3 pont

Írjon függvénysablont (*myequal*), ami az *std::equal* sablonhoz hasonlóan összehasonlítja két sorozatot! Ha a két sorozat összes eleme azonos, akkor igaz, ellenkező esetben hamis értékkel tér vissza. A függvény első két iterátor paramétere (*InputIterator*) az egyik sorozatot jelöli ki (jobbról nyílt intervallum kezdetét és végét), a 3. paramétere pedig a másik sorozat elejét. A függvény 4. paramétere egy predikátum függvény (kétparaméteres), amivel az elemek egyezősége tesztelhető. Amennyiben helyesen oldja meg a feladatot, akkor az alábbi kódrészlet a következőt írja ki: **true**

```
using namespace std;
int v[] = { 1, 2, 3, 4, 1, 2, 3 };
cout << boolalpha << myequal(v, v+3, v+4, std::equal_to<int>()) << endl;
```

```
template <class Iter, class Func>
bool myequal(Iter first, Iter last, Iter first2, Func func) {
    while (first != last) {
        if (!func(*first, *first2))
            return false;
        ++first;
        ++first2;
    }
    return true;
}
```