

Szoftver laboratórium 2.	5. Ellenőrző dolgozat.	2014.04.28.	Kurz/Terem: L2/
Név:	Neptun:	Összpont:	

1. feladat

2.5 pont

Írjon generikus függvényt (*forEach*), ami egy sorozattároló minden elemére meghív egy paraméterként kapott egyoperandusú függvényt/függvényobjektumot! A sablon és a függvény további paramétereit, visszatérési értékét Önnek kell meghatározni úgy, hogy a 2. feladatban azt hatékonyan fel tudja használni!

```
template <typename Iterator, class Func>
void forEach(Iterator first, Iterator last, const Func& func) {
    while (first != last)
        func(*first++);
}
```

2. feladat

3.5 pont

Tételezze fel, hogy rendelkezésére áll egy generikus tároló (pl. a korábbi laborgyakorlatokon elkészített *Array<T>*)! Mutassa be ennek felhasználásával az elkészített sablon használatát a következő kódrészletek megírásával:

1. Hozzon létre egy *egész* és egy *std::string* elemeket tartalmazó tárolót!
2. Tételezze fel, hogy létezik egy 300 elemű vektor (*int vektor[300]*)! Töltse fel az egész elemek tartalmazó tárolót a vektor elmeivel iterátorok használatával!
3. Készítsen olyan generikus függvényobjektumot, amit az 1. feladatban elkészített generikus algoritmussal használva ki tudja írni egy *std::ostream* típusú objektumra egy tároló összes elemét! Írja ki az egész tárolóból az összes elemet a hibakimenetre (*std::cerr*)!
4. Az elkészített sablonok felhasználásával írja ki a szabványos kimenetre az *std::string* elemeket tároló tárolóban levő összes stringet!

```
template <class T>
class ostreamFunctor {
    std::ostream& os;
public:
    ostreamFunctor(std::ostream& os) :os(os) {}
    void operator()(T& d) const { os << d << std::endl;}
};
```

```
Array<int> intTar(vektor, vektor+300);
```

```
Array<std::string> strTar;
```

```
forEach(intTar.begin(), intTar.end(), ostreamFunctor<int>(std::cerr));
```

```
forEach(strTar.begin(), strTar.end(), ostreamFunctor<std::string>(std::cout));
```

// a tömb feltöltése assign tagfüggvénnyel is elvégezhető:

```
Array<int> intTar;
```

```
intTar.assign(vektor, vektor+300);
```