

Szoftver laboratórium 2.	4. Ellenőrző dolgozat.	2014.04.14.	Kurz/Terem: L3/
Név:	Neptun:	Összpont:	

Tételezze fel, hogy rendelkezésére áll az előző laboron elkészített *Tartozek* osztály:

```
class Tartozek {
    String gyszam;
    String megnevezes;
public:
    Tartozek(const char*, const char*);
    virtual void print(std::ostream&) const;
    virtual ~Tartozek();
};
```

1. feladat

2 pont

A *Tartozek* osztály felhasználásával, annak módosítása nélkül hozzon létre egy olyan kijelző (*LedTV*) osztályt, ami együtt tárolható a már meglévő tartozékokkal! A *LedTV* jellemzője a képátló (*double*) amit a konstruktorban adunk meg. Valósítsa meg az új osztály minden szükséges tagfüggvényét úgy, hogy a *print* tagfüggvény a kijelző jellemzőjét is írja ki a kijelző egyéb adataival (gyári szám, megnevezés, ...) együtt!

2. feladat

2 pont

Alakítsa át a *LedTV* osztályt sablonná (*GenTV*) úgy, hogy a kijelző jellemzője tetszőleges típusú lehessen! Elegendő, ha csak a sablon lényegi sorait írja le, és a 1. feladat megoldásával azonos sorokra ... -tal hivatkozik! Adja meg, hogy a megoldás minimálisan milyen követelményeket támaszt a sablonban használt generikus típussal szemben (pl. , van & operátora)! Ne írjon olyan követelményt, amit a megoldás nem követel meg!

3. feladat

2 pont

Hozzon létre egy saját osztályt (*MyClass*) tetszőleges adattaggal! Valósítsa meg az osztály azon tagfüggvényeit, amelyek alkalmassá teszik a *GenTV* sablonban való használatra! Mutassa be az elkészített sablon használatát ezzel az osztállyal!

```
class LedTV : public Tartozek {
    double jellemzo;
public:
    LedTV(const char* gysz, const char* megn, double jellemzo)
        : Tartozek(gysz, megn), jellemzo(jellemzo) {}
    void print(std::ostream& os) const {
        Tartozek::print(os);
        os << " jellemzo: " << jellemzo;
    }
};
```

```
template<class T>
class GenTV : public Tartozek {
    T jellemzo;
public:
    GenTV(const char* gysz, const char* megn, T jellemzo)
        : Tartozek(gysz, megn), jellemzo(jellemzo) {}
    ...
};
```

T-vel szemben támasztott követelmény:

- legyen másolható,
- legyen olyan `std::ostream& operator<<(std::ostream&, const T&) operátor`, vagy ezzel kompatibilis, amivel kiírható.

```
struct MyClass {
    int adat;
    MyClass(int adat) : adat(adat) {}
};
std::ostream& operator<<(std::ostream& os, const MyClass& my) {
    return os << my.adat;
}
```

```
GenTV<MyClass> demo("345", "Generikus TV", MyClass(123));
demo.print(cout);
```