

11. Típuskonverziók

A gyakorlat célja a konverzió gyakorlása egy – már ismerős – sztring osztályon keresztül. A többértelműség előadáson nem kapott hangsúlyt, ez a gyakorlat anyaga.

1. Feladat. Írjunk *String* osztályt, amely a szokásos operátorokon kívül felhasználható C stílusú nullterminált sztringként.

- Tervezzük meg közösen az osztályt, főleg konverzió szempontjából. Miért a *const char** konverziós operátort definiáljuk felül, és miért nem a *char ** operátort? Miért érdemes reprezentációt váltani az előzőhöz képest?

A megoldásban szereplő reprezentáció megegyezik a régivel (pointer+ hossz), de a végére rakunk egy '\0' karaktert a konverziós operátor hatékonysága érdekében. Ugyanakkor megtartottuk a hossz – mostmár származtatott – adatot, hogy ne kelljen mindig bejárni a sztringet, ha a hosszát akarjuk, valamint biztonságosabb a memóriakezelés. Ez nem feltétlenül az egyedül üdvözítő megoldás.

- Teszteljük az osztályt a megadott *TheUltimateString.cpp* állománnyal!
- Helyezzünk el töréspontokat az összes konstruktor és konverziós operátor elején! Haladjunk át a *main()* függvény minden során, és indokoljuk meg, miért hívta meg azt a fordító. (Ezzel átismételjük a másoló konstruktorról tanultakat is).
- Miért kell a *printf()* függvény második hívásánál explicit típuskonverziót kiírni, és miért nem kell az elsőnél?
- Ha kihagyjuk az explicit típuskonverziót (*,(String)*) az alábbi kifejezésből, hibaüzenetet kapunk.

```
str1==(String)"My name is Bond. James Bond."
```

Mi okozza? Találjuk ki a többértelműség okát! Ha fejből nem megy, egyszerűsítsük le a *main()* függvény kódját, és kezdjük „kikommentezni” a konverziós függvényeket, amíg egyértelmű nem lesz!

Ez utóbbi feladat önálló, arra szolgál, hogy gyakoroltassa a hiba „lokalizálását”, és az önállóságot.

Házi Feladat. Írjuk bele a hiányzó függvényeket a *String* osztályba, amik a C könyvtárban megvannak! A C könyvtári függvények hívása megengedett.

Gyakorlófeladatok

1. [Referencia helytelen/helyes használata](#) f), g)