

Előszó

Napjaink gyakorlatilag egyeduralkodó programozási paradigmája az objektumorientált programozás (OOP). A szoftverrendszerekkel szemben támasztott követelmények – főként az átláthatóság és a karbantarthatóság – eddigi tudásunk alapján leginkább ebben a megközelítésben kezelhető. A C++ programozási nyelv nem a legfiatalabb objektumorientált nyelv, az utána következő programozási nyelvek, így a Java és a C# jelentős mértékben egyszerűsítik a nyelvi koncepciókat. Ezek alapján az is elmondható, hogy a C++ nem a legegyszerűbb OO nyelvek közé tartozik. Mindezek ellenére a nyelv nemcsak a különös szakmai kihívásokat öncélúan gyűjtő programozók „sportszerű nehezítésként” használt eszköze, hanem az operációs rendszerek, a POSIX platformok és számos beágyazott rendszer alapja. Ennek oka a C nyelvtől örökölt gépközeli konstrukciók megtartása, ebből adódó sebessége, és a C nyelv „veszélyes” megoldásainak biztonságossá tétele. A sebességnek természetesen ára van: a C++ nyelvben a figyelmetlen programozó nehezen felfedezhető hibákat ejtethet, és a nyelv hiányos ismerete hamar megbosszulja magát valamilyen nem várt vagy rejtélyes működés által.

E könyv azzal kíván segítséget nyújtani, hogy rámutat azokra a részekre, ahol a programozónak komolyan oda kell figyelnie a kódolásra, illetve arra, miként csökkentse különböző megoldásokkal a hibalehetőségeket.

Ezenkívül fontos szempontnak tartjuk, hogy a nyelv logikáját megismerve rögtön tudjuk a szokatlan viselkedés okát, valamint képesek legyünk megelőzni.

A harmadik célkitűzésünk az volt, hogy az objektumorientált szemlélet áthassa a C++ nyelvi konstrukcióinak ismertetését, általános „világnézetbe” helyezve őket ne csak azt írjuk le, hogyan működnek, hanem az adott konstrukció használatát, az átláthatóbb szoftver készítését elősegítő funkcióját is megmutassuk. Ez a célkitűzés indokolja, hogy könyvünk címében nem a programozás, hanem a szoftverfejlesztés szerepel. Ahol tehetjük, az újonnan bemutatott ismereteket példákkal illusztráltuk.

A könyv öt fő egységre tagolódik. A bevezetést a C nyelv ismeretét feltételező C++ tárgyalás hagyományos fejezete követi: témája, hogy miként teszi kényelmesebbé a C++ a C nyelvet. Régebben az erről szóló fejezetek jóval terjedelmesebbek voltak, de a C nyelv fejlődése következtében ma már kevesebb különbséget tapasztalhatunk.

A kilenc fejezetet felölelő második egység a C++ objektumorientáltságot támogató lehetőségeit mutatja be. Itt kerül sorra az egységbe zárás, az adat-rejtés és az öröklés alapelveinek megvalósítása, valamint néhány további lehetőség: az operátorok túlterhelése, a névterek, a kivételkezelés, a sablonok bemutatása.

Az ismét egy fejezetből álló harmadik egységet a szabványos C++ könyvtárnak szenteltük, különös tekintettel a tárolók és az algoritmusok bemutatására példákon keresztül. Ezt a fejezetet nem referenciának szántuk: célunk az volt, hogy a könyvtár dokumentációjával együtt – amely sokszor a fordító dokumentációjának része – az Olvasó képes legyen hatékonyan felhasználni a sablonkönyvtár nyújtotta lehetőségeket. Ebben a részben kapott helyet a magyar nyelvű fejlesztés tárgyalása is.

A negyedik egységben az UML osztálydiagramok C++ implementációjával foglalkozunk, amelyet egy nagyobb esettanulmány zár. Az esettanulmány nemcsak az UML nyelvű modellezést illusztrálja, hanem az egész könyv főbb fejezeteinek mondanivalóját is.

Az ötödik egységet a függelékben elhelyezett útmutatások alkotják. Elsőként bemutatjuk C++ operátorainak precedenciatáblázatát. Ezután azok számára, akik ipari projektet indítanának, bemutatjuk néhány gyakran használt eszköz (Microsoft Visual C++ és GNU Compiler Collection) sajátosságait. Mivel ezek a különbségek zavarnák az egységes és platformfüggetlen tárgyalást, a függelékben kaptak helyet. Végül röviden összefoglaljuk a C programozás meg szokott konstrukcióinak C++-beli megfelelőit.

A könyv alapját a Budapesti Műszaki és Gazdaságtudományi Egyetem Villamosmérnöki és Informatikai Karán választható Szoftverfejlesztés C++ nyelven tantárgy, a műszaki informatika szak alapképzésében oktatott Programozás alapjai II., valamint a villamosmérnöki szak alapképzésében megjelenő Programozás alapjai 2. tantárgy előadásai és kapcsolódó laborfoglalkozásai képezik. Így a könyv első fejezetei feltételezik, hogy az Olvasó tisztában van a C nyelv alapjaival, az alapvető adatstruktúrákkal (dinamikus tömbök, láncolt listák, bináris fák) és algoritmusokkal (keresés, rendezés), emellett rendelkezik elemi programozástechnikai ismeretekkel, amelyek például az [1] irodalomhivatkozásnak megfelelő könyvek áttanulmányozásával szerezhetők meg. Ahol csak tehetjük, az egyes funkciókat egyszerű példákkal illusztráltuk.

Reményeink szerint azok, akik végigolvassák a könyvet, felfedezik a C++ eleinte bonyolultnak tűnő nyelvi elemei mögötti rendszert, a nyelv logikáját, elsajátítják az objektumorientált szemlélet alapjait, valamint képesek lesznek hatékonyan kihasználni a szabványos C++ könyvtár lehetőségeit.

A könyvet tankönyvként használók számára a következő fejezetek feldolgozását javasoljuk:

- C++ alapkursus nem szoftver szakirányos képzésben: 2-5, 6.1–6.5, 7–8, 9.1–9.4, 10–11, 12.1.1, 12.5.3
- C++ alapkursus szoftver szakirányos képzésben: 2–11, 12.1, 12.5, 13, 14.1
- Haladó C++ kurzus: 12–14
- Beágyazott rendszerek programozása kurzus: 2–5, 6.1–6.5, 7–8, 9.1–9.4, 10–11

Igyekeztünk elérni, hogy az egyes fejezetek a lehetőségekhez mérten és a téma jellegétől függően önálló egészet alkossanak, esetleges rövidebb ismétlések árán is. Mivel jelen munkát nem kézikönyvnek szántuk, hanem elejétől a végéig elolvasható tankönyvnek, a felépítés kialakítása során nagy hangsúlyt vittünk figyelembe a didaktikai szempontokat. A C++ nyelvben minden mindennel összefügg, így már az első fejezetekben előfordul, hogy egy adott jelenség magyarázatát csak egy későbbi fejezet megismerése után lehet megérteni. A könyv összeállítása során, ahol csak lehet, elkerültük ezeket a helyzeteket. Ahol nem sikerült, ott lábjegyzetbeli vagy egyéb utalással mutatunk rá az összefüggésekre. Másrészről vannak olyan osztálykönyvtárbeli elemek – jellemzően a kiírást, illetve beolvasást megvalósító objektumok –, melyek használata egyszerű, ahhoz viszont, hogy teljes mélységükben tárgyalhassuk őket, gyakorlatilag az egész C++ nyelvre szükségünk van. Ezek tárgyalását két részre osztottuk: egy bevezető fejezet ismerteti a használatukat, a későbbi fejezetek pedig részletesen bemutatják a működésüket.

Az itt leírt ismeretek tananyagként való számonkérése a téma jellege miatt különösen nehézkes. Ezt megkönnyítendő a számon kérhető fogalmakra és a fontosabb mondanivalóra vastag betűkkel és útmutatókkal hívtuk fel a figyelmet, valamint a könyv végén „kislexikon” jelleggel kigyűjtöttük a fontosabb fogalmakat és magyarázatukat. A haladóbb fejezetek címeit dőlt betűtípus és a végén * jelzi. A folyó szövegben gyakran hivatkozunk a programkódban szereplő változókra, konstansokra, makrókra stb. Ezeket az elkülöníthetőség végett dőlt betűkkel szedtük. A programrészleteket szürke háttér jelzi. A programrészletekben található megjegyzések magyar nyelvűek, az általuk kiírt szöveg nyelve angol, kivéve a magyar nyelvű fejlesztést és a szövegfeloldozást ismertető fejezetekben.

Az egyes új fogalmak mellett további tájékozódás megkönnyítésére zárójelben közöljük az angol megfelelőiket. Az angol neveknél vastag betűvel jelöltük meg a fogalomhoz kapcsolódó gyakrabban használt betűszót, illetve rövidítést.

Jelen munkában a szerzők többéves ipari és oktatási tapasztalatai is megjelennek, amelyek számos projekt, internetes oldal, kurzusanyag eredményei – ezek forrásai ma már felidézhetetlenek. A köszönetnyilvánításban igyekeztünk ezt némileg pótolni. Vannak olyan források (jellemzően a fordítók dokumentációi), amelyeket lépten-nyomon használtunk, ezekre nem hivatkozunk minden egyes alkalommal.

Elsőként szeretnénk megköszönni Adamis Gusztáv különösen gondos lektori munkáját és értékes tanácsait. Köszönetünket szeretnénk kifejezni Péteri Szilárdnak és Kovács Tibornak a borító és a CD-melléklet külső megjelenésének elkészítéséért, valamint az Automatizálási és Alkalmazott Informatikai Tanszék Alkalmazott Informatika Csoportjának, hallgatóinknak és a kiadó munkatársainak. Továbbá köszönettel tartozunk Iváncsy Renátának, Kovács Ferencnek, Lengyel Lászlónak, Vajk Istvánnak és Vajk Tamásnak a kézirat átolvasásáért és az értékes megjegyzésekért.

Végül pedig bízunk abban, hogy ez a könyv sokak számára lesz megbízható segítség tanulmányaik és munkájuk során, és reményeink szerint mindez számos áttekinthető, igényesen megírt, jól használható és hatékony C++ nyelvű szoftver elkészítésében jelenik meg.

Budapest, 2007. augusztus

A szerzők