Inside C#

Tom Archer

Microsoft®

IT Press

Inside C#

Kirjoittanut	Tom Archer
Kääntäjä	Jussi Arola
Kansi	Frank Chaumont
Kustantaja	Edita Oyj IT Press PL 760 00043 EDITA Sähköpostiosoite palvelu@itpress.fi Internet www.itpress.fi
Painopaikka	Edita Oyj, Helsinki 2001

Alkuperäisen teoksen nimi on *Inside C#*

ISBN 951-826-455-4

Kirjan luvut

Sisällysluettelo

Esipuhe

Olen ollut koko urani ajan Microsoftilla parantamassa ohjelmoijien mahdollisuuksia, yleensä keskittyen ohjelmoinnin tuottavuuden nostoon. Työni on käsittänyt suuren määrän eri tuotteita ja tekniikoita, mutta en koskaan ole ollut näin innostunut työstäni kuin nyt. Se mahdollisuuksien laajuus, jonka Microsoft .NET tarjoaa, on uskomaton. Tarjoamme uuden ohjelmointikielen rikkoen perinteisen ohjelmoijien jaon erillisiin mutta erilaisiin kielimaailmoihin ja antaen Web-sivustojen osaltaan auttavan vastaamaan käyttäjien tarpeisiin. Jokainen näistä olisi jo yksinään mielenkiintoinen, mutta niiden yhdistelmä on todella mukaansatempaava.

Katsotaan .NETin peruspalikoita ja niihin liittyviä tekniikoita:

- **C#, uusi kieli** C# on ensimmäinen komponenttisuuntautunut kieli C ja C++ -kieliperheessä. Se on yksinkertainen, nykyaikainen, olioperusteinen ja tyyppiturvallinen ohjelmointikieli, joka periytyy C:stä ja C++:sta. C# yhdistää Microsoft Visual Basicin tuottavuuden ja C++:n tehon.
- **Common Language Runtime** Tehokas Common Language Runtime (CLR) sisältää suoritusmoottorin, roskienkeruun, täsmäkääntäjän ja laajan luokkakirjaston (.NET Framework). CLR suunniteltiin alusta asti tukemaan useita ohjelmintikieliä.
- **Common Language Specification** Common Language Specification (CLS) kuvaa kielen toiminnallisuuden yleisen tason. CLS:n suhteellisen tiukka vaatimustaso mahdollistaa CLS-sopivien kielten joukon kehittämisen. Sen kukin jäsen nauttii kahdesta edusta: täydestä .NET Frameworkin toiminnallisuudesta ja yhteistyöstä muiden CLS-sopivien kielten kanssa. Esimerkiksi Visual Basic voi periä luokan C#:sta ja korvata sen virtuaalisen metodin.
- **Laaja joukko CLR:ään perustuvia kieliä** Microsoftin toteuttamia CLR-sopivia kieliä ovat Visual Basic, Visual C++ Managed Extensions -laajennuksin, Visual C# ja JScript. Kolmannet osapuolet ovat tuottamassa monia muita kieliä. Kieliä on niin monta, että en voi edes luetella niitä tässä!
- **Web services -palvelut** Nykyään World Wide Web koostuu pääosin yksittäisistä sivustoista. Vaikka käyttäjä voi vierailla useilla sivustoilla suorittaakseen määrätyn tehtävän, kuten esimerkiksi tehdä ryhmän

matkajärjestelyt, nämä sivustot eivät yleensä toimi yhteistyössä. Webin seuraava sukupolvi perustuu Web-sivustojen yhteistyöverkostoon. Syy on yksinkertainen: yhteistoiminnassa Web-sivustot täyttävät paremmin käyttäjien vaatimukset. Microsoftin Web services -palvelut edistävät Web-sivustojen yhteistyötä mahdollistamalla tietojen siirron XML-perusteisilla protokollilla, jotka ovat sekä kieliriippumattomia että alustariippumattomia. Monet tärkeät Web-palvelut tulevat perustumaan C#:lle ja Windows-käyttöjärjestelmässä pyörivälle CLR:lle mutta arkkitehtuuri on aidosti avoin.

- **Visual Studio.NET** Visual Studio.NET sitoo nämä kaikki palat yhteen ja tekee helpoksi julkaista erilaisia komponentteja, sovelluksia ja palveluja, jotka on tehty eri ohjelmointikielillä.

Nyt kun olen maininnut muutamista tärkeistä C#-kieleen liittyvistä tekniikoista, katsotaan tarkemmin itse C#-kieltä. Ohjelmoijat ovat investoineet paljon valitsemaansa kieleen ja siten on uuden kielen velvollisuus todistaa arvonsa yhdistämällä hyväksi koettuja ratkaisuja, lisäparannuksia ja syvällisiä keksintöjä.

Hyväksi koetut ratkaisut

Hippocrates sanoi, "Tee kahdesta asiasta tapa: auta tai älä ainakaan aiheuta haittaa." "Älä aihauta haittaa"-osa näytteli merkittävää osaa C#:n suunnittelussa. Jos C:n tai C++:n ominaisuus ratkaisi ongelman hyvin, emme muuttaneet sitä. Oleellisimmailta osiltaan C# lainaan C:stä ja C++:sta ydinalueilla, kuten lausekkeissa, käskyissä ja koknaisrakenteessa. Koska normaalista ohjelmasta suuri osa koostuu näistä osista, C ja C++-ohjelmoijat tuntevat heti olonsa kotoisaksi C#:n kanssa.

Lisäparannuksia

Paljon erilaisia parannuksia on tehty, liian paljon, jotta niitä voisi luetella esipuheessa. Mutta seuraavassa mainitsen muutamia, jotka ovat aiheuttaneet yleisiä ja aikaa vieviä ongelmia C ja C++ -ohjelmissa:

- Muuttujat pitää alustaa ennen käyttöä, joten alustamattoman muuttujan käytöstä aiheutuneet virheet jäävät pois.
- *if* ja *while* -käskyt tarvitsevat Boolean-arvon, joten ohjelmoija, joka vahingossa käyttää sijoitusoperaattoria (=) vertailuoperaattorin (==) sijasta, löytää virheen jo käännöksessä.
- "Läpijuoksu" *switch*-käskyssä on estetty, joten ohjelmoija, joka epähuomiossa unohtaa *break*-käskyn, löytää virheen jo käännöksessä.

Syvällisiä keksintöjä

Syvällisempiä keksintöjä on löydettävissä C#:n tyyppijärjestelmästä. Se sisältää seuraavia etuja:

- C#:n tyyppijärjestelmä suorittaa automaattisen muistinhallinnan ja vapauttaa siten ohjelmoijan aikaavievältä ja virheherkältä muistin hallinnalta. Toisin kuin useimmat tyyppijärjestelmät, C#:n tyyppijärjestelmä mahdollista myös osoitintyyppien ja objektiosoitteiden suoran käsittelyn. (Nämä toimenpiteet sallitaan vain määrätyssä turvaympäristössä.)
- C#:n tyyppijärjestelmä on yhtenäinen, sillä kaikki ovat objekteja. Paketointi (boxing) ja purkaminen (unboxing) -tekniikoiden avulla C# yhdistää arvotyyppin ja viittaustyypin erot ja salliin kaiken tiedon käsittelyn objektien tapaan.
- Ominaisuudet, metodit ja tapahtumat ovat C#:n perusjuttuja. Monet kielet jättävät pois luontaisen tuen ominaisuuksille ja tapahtumille ja aiheuttavat tarpeettoman epäyhteensopivuuden kielen ja siihen liittyvän kehyksen välille. Jos esimerkiksi kehys tukee ominaisuuksia ja kieli ei, ominaisuuden lisäämnen on kömpelöä (esimerkiksi *o.SetValue(o.GetValue( ) +1)*). Jos myös kieli tukee ominaisuuksia, toiminto on yksinkertainen (*o.Value*++).
- C# tukee attribuutteja, joiden avulla määritellään komponentteja ja liitetään selventävää informaatiota niihin. Mahdollisuus määritellä uuden tyyppistä informaatiota on aina ollut tehokas työkalu kielen suunnittelijoille. Nyt kaikilla C#-hjelmoijilla on tämä mahdollisuus.

Inside C#

Inside C#-kirjassa Tom Archer luo pohjan esittelemällä .NETin ja CLR:n, selvittää C#:n perusteet, ja sukeltaa sitten vaativampiin C#-kielen rakenteisiin. Hänen syvällinen kokemuksensa C++:sta, J++:sta ja Microsoft Windowsista, sekä ohjelmoijana että kirjoittajana, antaa mahdollisuuden selvittää C# tavalla, joka on sekä miellyttävä että informatiivinen.

Lukijat, toivon, että nautitte ensimmäisen C#-ohjelmanne kirjoittamisesta ja toivon myös, että se on ensimmäinen niistä monista, joita tulevina vuosina tulette kirjoittamaan.

Scott Wiltamuth
C#-suunnitteluryhmän jäsen
Microsoft Corporation

Johdanto

Miksi kirjoitin tämän kirjan

Olen tehnyt ohjelmia 20 vuotta (tunnen oloni vanhemmaksi aina kun ajattelen sitä!) ja olen tullut tilanteeseen, jossa ohjelmointi on alkanut tuntua vanhan toistolta. Älä käsitä minua väärin: jos olisin monimiljonääri eikä minun tarvitsisi tehdä työtä, luultavasti silti jatkaisin ohjelmien kirjoittamista, koska todella nautin siitä paljon. Olen kuitenkin ruvennut ajattelemaan, "Kaikki on tehty!" Sitten tulivat Microsoft .NET ja C# ja avautui kokonaan uusi mahdollisuuksien maailma. Olen puhnut muutamien kaverien kanssa, joille on myös tullut tämä jonkinasteinen uudelleenherääminen .NETin julkaisun myötä. Meillä on uusi, jännittävä tekniikka, joka lopulta ratkaisee asiat, joiden parissa olemme painineet vuosia (esimerkiksi monikieliset kehitysympäristöt, suurten järjestelmien jakelu- ja versiointiongelmat ja niin edelleen.) Kirjoitin tämän kirjan, koska koodin kirjoittaminen on taas jännttävää. Koska jälleen nousen aamuisin ylös ajatellen uutta ja mielenkiintoista asiaa, jonka tulen oppimaan. Toivon, että samalla kun opit tämän uuden kielen, jaat innostukseni.

Jokainen, joka kirjoittaa kirjaa C#:sta tämän kirjoittamisen aikaan, pitää opetella itse kieli samaan aikaan. Jos olet tehnyt sovellusta ja samaan akaan opetellut käytettävää SDK:ta tai kieltä (kukapa ei olisi?) tietää, että se on hankala tilanne. Yritä sitten kuvitella, että kymmenet tuhannet ihmiset tulevat arvostelemaan sen, kun se on valmis! Suurin ongelma on se, että puoliväissä projektia, kun olet selvittänyt, mitä sinun tulee tehdä, palat halusta suunnitella ja tehdä koko asia uudelleen! Kirjan valmistumisaikataulun takia se on tietenkin mahdotonta toteuttaa. Uskon kuitenkin, että tämä kirja on hyvä apuväline C#:n opiskeluun. Koska opettelin itse samalla kuin kirjoitin tätä, kirjaan jäi ristiriitaisuuksia ja asioita, joita olisin voinut tehdä paljon paremmin. Mutta jos saan mahdollisuuden tehdä kirjasta toisen painoksen, voin luvata, että sinä ja minä tulemme hyötymään omasta oppimisestani, jonka olen tätä kirjaa tehdessäni hankkinut.

Lopksi haluan sanoa, että kaikki tätä kirjaa koskeva palaute on tervetullutta. En ole yksi noista "Olen niin hyvä, koska kirjoitan kirjoja" -kavereista. Olen aivan tavallinen kaveri, joka on ollut onnekas saadessaan mahdollisuuden tämän kirjan kirjoittamiseen. Olen aina valmis oppimaan muilta ja itse asiassa pidän siitä kovasti. Saat minuun yhteyden osoitteessa *http://www.TheCodeChannel.com*.

Kenen tulisi lukea tämä kirja

Tämä kirja on henkilöille, jotka suunnittelevat C# ja .NET-sovelluskehityksen aloittamista. Kuten mainitsin, tämä on yksi mielenkiintoinen alusta ja tulee olemaan tulevaisuudessa niin kauan kuin hajautettua Microsoft Windows -ohjelmistokehitystä tehdään. Tämä kirja olettaa, että sinulla on taustaa jostakin C-perheen kielestä: C:stä, C++:sta tai Javasta. Ainoa muu tarvittava vaatimus on mielestäni päätös oppia ja hankkia uusia ulottuvuuksia sovellusten kirjoittamiseen. Ja koska pidät tätä kirjaa kädessäsi, uskon, että sinä luultavasti olet sen päätöksen tehnyt!

Kirjan rakenne

Tämä kirja on järjestetty huolellisesti neljään loogisesti peräkkäiseen osaan, joista kukuin koostuu joukosta lukuja. Kukin luku käsittelee määrätyn alueen C# tai .NET-ohjelmoinnista.

Kirja alkaa osalla I, "Pohjan luominen," joka on tarkoitettu aloitteleville C#-ohjelmoijille ja niille, joille .NET ei ole ennestään tuttu. Osan luvut tarjoavat johdannon .NETiin ja näyttävät, miten luot ja testaat ensimmäisen C#-sovelluksesi.

Osassa II, "C#-luokkien perusteet," esittelen C#-luokkien määrittelyn ja käytön perusteet. Osan luvut antavat sinulle vankan tietopohjan siihen, miten C# tukee jäseniä (luetellut tyypit, ominaisuudet, taulukot a niin edelleen) ja miten määrittelet ja käytät niitä C#-sovelluksessa.

Vaikka olet kirjoittanut koodia useissa luvuissa määritellessäsi luokan jäseniä, osassa III, "Koodin kirjoittaminen," alat nähdä erilaisia tapoja eri tehtävien tekemiseen, kuten ohjauskäskyt, virheiden käsittely (poikkeusten avulla) ja tapahtumaksäittelijöiden kirjoittaminen delegaattien avulla.

Kirja päättyy osaan IV, "Vaativampi C#." Tyypillisenä nörttinä nautin eniten tämän osan kirjoittamisesta. Se sisältää luvut monisäikeisestä ohjelmoinnista, reflection-menetelmästä, työskentelystä hallitsemattoman koodin kanssa (mukaanlukien COM-yhteistoiminta) ja versioinnista.

Kirjan mukana tuleva cd

Tämän kirjan mukana tulee cd. Jos Windowsissasi on Autorun-toiminto päällä, saat aloitusruudun eteesi, kun asetat cd:n cd-rom-asemaan. Aloitusruudulta voit valita asennusvalinnat. Voit käynnistää aloitusruudun myös cd-juurihakemistosta käskyllä StartCD. StartCD-ohjelma sisältää linkin cd:llä olevaan eKirjaan, asennusohjelman kirjan esimerkkitiedostoille ja linkin MSDN:ään, josta voit ladata Microsoft .NET Framework SDK:n viimeisimmän version, jota tarvitset kääntääksesi ja ajaaksesi esimerkkiohjelmat.

Kirjan esimerkkiohjelmat sjaitsevat Code-kansiossa. Voit selata esimerkkejä cd:ltä tai voit asentaa ne koneellesi käyttämällä StartCD:n asennusohjelmaa.

> Huomaa Jos et pysty selaamaan Samples-kansion tiedostoja, sinulla saattaa olla vanhempi cd-ohjain, joka ei tue pitkiä tiedostonimiä. Tällöin sinun pitää asentaa esimerkkitiedostot kiintolevyllesi asennusohjelmalla, jotta voisit tutkia niitä.

Järjestelmävaatimukset

Jos haluat saada tästä kirjasta mahdollisimman paljon irti, suosittelen, että käyt esimerkkisovellukset läpi samalla kuin luet tekstiä. Siksi sinun pitää asentaa viimeisin .NET Framework SDK. Tämän kirjoitushetkellä se sisältää .NETin ajonaikaisen ympäristön ja C#-kääntäjän. Olen lisäksi tietoisesti välttänyt Visual Studio.NET-tuotteiden käyttöä ja keskittynyt itse kieleen ja ajonaikaiseen ympäristöön, jotta en rajoittaisi sinua määrättyyn kehitysympäristöön. Siksi kirjan kaikki esimerkit käännetään ja suoritetaan komentorivikäskyillä.

Kiitokset

Ensimmäiseksi haluan kiittää avusta toimittajaani, Devon Musgravea. En teeskentele olevani "kirjoittaja." Olen ohjelmoija, joha haluaa auttaa muita ja kirjojen kirjoittaminen on yksi tapa tehdä se. Ilman Devonin taivuttelua ja muotoilua se mitä sanoin eroaisi huomattavasti siitä, mitä halusin sanoa ja kirjan teksti ei olisi lähimainkaan niin luettavaa kuin se nyt on. Kiitokset, Devon!

Haluan kiittää myös Brian Johnsonia, kirjan teknistä toimittajaa. Brian oli korvaamaton, kun yritimme täsmätä kirjan tekstiä ja esimerkkiohjelmia. Hän oli myös suureksi avuksi selvitellessämme kääntäjään tehtyjä viime hetken muutoksia, kun kirja oli menossa painoon. Kirjoitin tämän kirjan käyttäen C#:n beta-versiota. Kun kääntäjä aikanaan julkaistaan, muutama kirjan esimerkeistä ei ehkä toimi. Yksikään niistä ei ole kuitenkaan Brianin vika, sillä hän oli äärimmäisen huolellinen testatessaan jokaisen esimerkkiohjelman.

Myös kaksi muuta Microsoft Pressin henkilökunnasta ansaitsee kiitokset: Anne Hamilton ja Danielle Bird. He molemmat olivat vastuussa tämän kirjaprojektin käynnistymisestä ja he antoivat minulle mahdollisuuden tämän kirjan kirjoittamiseen. Heidän uskonsa minuun auttoi muutamien vaikeiden hetkien ohi ja arvostan paljon heidän tukeaa. Kiitokset!

Haluan myös kiittää seuraavia Microsoftin työntekijöitä, jotka suhtautuivat kärsivällisesti kysymyksiini, kun opiskelin C#-kieltä ja .NET BLC:tä: Joe Nalewabau (joka vastasi muutamiin aikaisiin C#-kysymyksiini), Brian Harry (joka tarjosi tietoja deterministisestä lopetuksesta) ja Steven Pratschner (joka auttoi yleiseen koostevarastoon ja koosteen versiontiin liittyvissä asioissa). Kiitokset myös Scott Wiltamuthille hänen erinomaisesta esipuheestaan.

Lopuksi haluan kiittää Aravind Coreraa, joka auttoi valtavasti luvussa, joka käsittelee hallitsematonta koodia. Hänen erinomaisen COM-kappaleensa ansiosta luku sisältää jotain erikoista. Hienoa työtä, Aravind. Toivon, että voimme tehdä yhteistyötä toisenkin kerran!