

Introducere

Această carte vă arată cum să scrieți programe care rulează sub Microsoft Windows. Există mai multe căi prin care puteți scrie asemenea programe. În această carte utilizez noul limbaj de programare orientat spre obiecte C# (pronunțat C sharp – în traducere literală C diez) și o bibliotecă modernă de clase numită *Windows Forms*. Biblioteca de clase Windows Forms face parte din Microsoft .NET (punct net) Framework anunțată în vara anului 2000 și lansată pe piață aproximativ un an și jumătate mai târziu.

Microsoft .NET Framework este o colecție cuprinzătoare de clase, care furnizează programatorilor aproape tot ce le trebuie pentru a scrie aplicații pentru Internet, Web și Windows. .NET se adresează cu predilecție domeniului programării pentru Web. Această carte prezintă *cealaltă* parte a mediului .NET. Utilizați Windows Forms pentru a scrie aplicații tradiționale, autonome, pentru Windows (ceea ce în prezent se numește uneori aplicație *client*) sau programe de interfață pentru aplicații distribuite.

Windows Forms asigură aproape tot ce vă trebuie pentru a scrie aplicații Windows complet echipate. Marea omisiune o reprezintă suportul pentru multimedia. Nu există nici măcar o funcție Windows Forms care să emită un semnal sonor prin difuzorul calculatorului! Am fost tentat să scriu singur clase multimedia, dar m-am oprit gândindu-mă (logic, presupun) că următoarea versiune de Windows Forms va include suport multimedia flexibil, performant și ușor de utilizat.

Clasele definite în .NET Framework sunt neutre față de limbaj. Microsoft a lansat noi versiuni de C++ și Visual Basic, care pot utiliza aceste clase, precum și noul limbaj de programare C#. Alți producători de limbaje își adaptează limbajele pentru a putea utiliza clasele .NET. Aceste noi compilatoare (fie opționale, fie prestabilite) convertesc codul sursă într-un limbaj intermediar într-un fișier .exe. În timpul execuției, este compilată varianta scrisă în limbajul intermediar într-un cod-mașină corespunzător microprocesorului. Astfel, .NET Framework este potențial independent de platformă.

În această carte am ales să folosesc C# deoarece C# și .NET au fost create, într-un mod cât se poate de concret, unul pentru altul. Datorită aspectului neutralității față de limbaj al mediului .NET Framework puteți utiliza această carte pentru a învăța să scrieți aplicații Windows Forms folosind alte limbaje .NET.

Programarea pentru Windows: o prezentare de ansamblu

Microsoft a lansat pe piață prima versiune de Windows în toamna anului 1985. De atunci, Windows a fost actualizat și îmbunătățit treptat, cele mai marcante progrese fiind vizibile în Windows NT (1993) și Windows 95 (1995), când Windows a trecut de la arhitectura pe 16 biți la cea pe 32 de biți.

Când Windows a fost lansat pentru prima dată, exista cu adevărat o singură posibilitate de a scrie aplicații pentru Windows, iar aceasta presupunea folosirea limbajului de programare C pentru a accede la interfața de programare a aplicațiilor Windows (API). Deși era

posibil să accedeți la Windows API și utilizând Microsoft Pascal, această abordare era utilizată arareori.

Pe parcursul anilor, multe alte limbaje au fost adaptate pentru a realiza programe pentru Windows. Printre acestea se numără Visual Basic și C++. Atât C++, cât și C# sunt limbaje orientate spre obiecte care recunosc majoritatea tipurilor, operatorilor, expresiilor și instrucțiunilor din limbajul de programare C. Din acest motiv, C++ și C# (precum și Java) sunt numite uneori limbaje *bazate pe C*, sau limbaje care aparțin *familiei C*.

După introducerea mediului .NET, Microsoft pune la dispoziție trei abordări pentru scrierea aplicațiilor Windows, utilizând un limbaj bazat pe C:

Cum să scrieți o aplicație pentru Windows utilizând un limbaj bazat pe C (abordare axată pe evoluția Microsoft)

Anul lansării	Limbaj	Interfață
1985	C	Interfața de programare a aplicațiilor Windows (API)
1992	C++	Biblioteca Microsoft Foundation Class (MFC)
2001	C# sau C++	Windows Forms (face parte din .NET Framework)

Nu este sarcina mea să vă spun ce limbaj sau ce interfață ar trebui să utilizați pentru a scrie aplicații sub Windows. Aceasta este o decizie pe care o puteți lua doar dumneavoastră, în funcție de programul pe care trebuie să-l scrieți și de resursele care vă stau la dispoziție.

Dacă doriți să învățați mai multe despre Windows API, mulți au considerat că lucrarea mea intitulată *Windows Programming* (ediția a cincea, Microsoft Press, 1998) este valoroasă.

Niciodată nu mi-a plăcut MFC. Încă de la apariția sa pe piață, mi s-a părut că este slab concepută și că nu prea este orientată spre obiecte. În consecință, nu am scris niciodată nimic despre MFC. Acesta este, însă, doar un punct de vedere personal. Mulți alți programatori au folosit cu succes MFC, iar în prezent este una dintre abordările care se bucură de cea mai mare popularitate ale programării pentru Windows. Puteți învăța multe despre MFC din cartea *Programming Windows with MFC* (ediția a doua, Microsoft Press, 1999) de Jeff Prosise. Pentru programatorul de aplicații Windows, care are mai multă experiență, aș dori să recomand *Programming Applications for Microsoft Windows* (Microsoft Press, 1999) de Jeffrey Richter.

După părerea mea, Windows Forms este mult mai bine proiectat decât MFC și mult mai apropiat de ceea ce consider eu a fi o interfață ideală pentru Windows. În ultimele 14 luni, în care am lucrat la această carte, Windows Forms a devenit abordarea mea preferată a programării pentru Windows.

Din punctul de vedere al programării, ambele interfețe, MFC și Windows Forms, funcționează prin apeluri la Windows API. Din punctul de vedere al arhitecturii, se poate spune că acestea se află deasupra lui API. Aceste interfețe de nivel mai înalt sunt concepute pentru a simplifica programarea sub Windows. De obicei puteți efectua diverse activități în MFC sau Windows Forms utilizând mai puține instrucțiuni decât atunci când utilizați API.

Deși interfețele de nivel înalt, cum ar fi MFC sau Windows Forms, adesea cresc productivitatea programatorului, orice interfață care utilizează o altă interfață este evident mai puțin flexibilă decât interfața pe care se bazează. Puteți face multe lucruri utilizând Windows API, pe care nu le puteți face utilizând Windows Forms.

Din fericire, cu ceva muncă în plus, puteți efectua apeluri la Windows API dintr-un program Windows Forms. Doar ocazional am întâlnit în această carte o omisiune în .NET Framework atât de serioasă încât a trebuit să utilizez această facilitate. Filosofia mea a fost în general aceea de a respecta izolarea pe care Windows Forms o asigură față de mecanismele interne al sistemului de operare Windows propriu-zis!

Cerințele pentru utilizator

Pentru a utiliza această carte cât mai profitabil, trebuie să puteți compila și rula programe C#. Pentru a compila programele, aveți nevoie de un compilator de C#. Pentru a rula aceste programe, aveți nevoie de .NET Runtime (numit Common Language Runtime, sau CLR) care este o colecție de biblioteci cu legare dinamică.

Ambele aceste elemente sunt incluse în Microsoft Visual C#, un mediu de dezvoltare integrat modern. Ca o alternativă, puteți cumpăra pachetul cuprinzător și mai costisitor Microsoft Visual Studio .NET care vă va permite, de asemenea, să programați în C++ și Visual Basic, nu numai în C#.

Dacă preferați o abordare mai greoaie, puteți descărca pachetul de dezvoltare a aplicațiilor .NET Framework distribuit gratuit (SDK). Varianta care poate fi descărcată include un compilator de C# cu linie de comandă și .NET Runtime. Mai întâi vizitați adresa <http://msdn.microsoft.com/downloads>. La stânga, selectați Software Development Kits (Pachete de dezvoltare a aplicațiilor), după care căutați .NET Framework. (Nu uitați că acest site Web, la fel ca toate site-urile Web menționate în această carte, ar putea să se modifice, să se mute sau să dispară complet în orice moment.)

Am scris această carte pornind de la premisa că știți cel puțin să programați în C. Cunoașterea limbajelor C++ și Java este utilă, dar nu indispensabilă. Deoarece C# este un limbaj nou, primul capitol al acestei cărți face o prezentare rapidă a limbajului C# și a conceptelor esențiale ale programării orientate spre obiecte. În restul acestei cărți, îmi rezerv adesea timp să prezint diverse concepte C# pe măsură ce le întâlnesc.

Această carte nu reprezintă însă un manual cuprinzător al limbajului C#. Dacă doriți să aveți mai multe cunoștințe și o îndemânare mai mare în lucrul cu acest limbaj, sunt disponibile alte cărți de C# și multe altele vor deveni cu siguranță disponibile pe măsură ce popularitatea limbajului crește. Cartea *Inside C#* (Microsoft Press, 2001) de Tom Archer oferă informații privind scrierea programelor C# și privind „dedesubturile” acestui limbaj. *Microsoft Visual C# Step by Step* (Microsoft Press, 2001) de John Sharp și Jon Jagger are o abordare mai „școlărească”.

Uneori în această carte fac referire la Windows API. Așa cum am spus mai înainte, consultați cartea mea intitulată *Programming Windows* pentru a învăța mai multe despre API.

Cerințe de sistem

Așa cum am menționat în subcapitolul precedent, pentru a utiliza eficient această carte, trebuie să puteți compila și rula programe în C#. Cerințele de sistem sunt următoarele:

- Microsoft .NET Framework SDK (cel puțin); Microsoft Visual C# sau Microsoft Visual Studio .NET (de preferință)
- Microsoft Windows NT 4.0, Windows 2000 sau Windows XP

Pentru a rula programele dumneavoastră la alte calculatoare, trebuie să aveți .NET Runtime (numit și .NET Framework *pachet redistribuibil*) instalat la respectivele calculatoare. Acel pachet însoțește .NET Framework SDK, Visual C# și Visual Studio .NET. pachetul redistribuibil poate fi instalat sub versiunile de Windows menționate anterior, precum și sub Windows 98 și Windows Millenium Edition (Me).

Cum este organizată această carte

La lansarea produsului Windows 1.0, întreaga interfață API a fost implementată în trei biblioteci cu legare dinamică, numite KERNEL, USER și GDI. Deși bibliotecile cu legare dinamică asociate cu Windows au ajuns să fie mult mai voluminoase, este în continuare util să împărțiți apelurile la funcții Windows (sau la clase ale cadrului de lucru) în aceste trei categorii: apelurile la nucleu sunt cele implementate în arhitectura sistemului de operare și privesc de obicei lucrul cu taskuri, gestionarea memoriei și operațiile de intrare/ieșire pentru fișiere. Termenul *user* se referă la interfața cu utilizatorul. Acestea sunt funcții pentru crearea ferestrelor, utilizarea meniurilor și a casetelor de dialog și afișarea elementelor de control, cum ar fi butoane și bare de derulare. GDI este Graphics Device Interface, acea parte din Windows care răspunde de afișarea ieșirii grafice (inclusiv a textului) pe ecran și la imprimantă.

Această carte începe cu patru capitole introductive. Începând de la capitolul 5 (care vă arată cum să desenați linii drepte și curbe) și până la capitolul 24 (despre memoria Clipboard din Windows), capitolele prezintă alternativ subiecte de grafică (capitolele cu numere impare) și subiecte legate de interfața cu utilizatorul (capitolele cu numere pare).

În mod normal, o carte ca aceasta nu va pierde prea mult timp cu prezentarea unor subiecte care nu sunt legate de Windows, cum ar fi operațiile de intrare și ieșire pentru fișiere, aritmetica în virgulă mobilă și manipularea șirurilor de caractere. Pe de altă parte, pentru că .NET Framework și C# sunt atât de noi, m-am aflat în situația să-mi doresc să am un îndrumar coerent care să-mi prezinte aceste clase. Atunci, am scris singur acest îndrumar. Acesta cuprinde trei anexe cu fișiere, operații matematice și șiruri. Puteți consulta aceste anexe în orice moment ulterior citirii capitolului 1.

Am încercat să ordonez aceste capitole – și subiectele din capitole – astfel încât fiecare subiect să se bazeze pe subiectele precedente, făcând un număr minim de trimeri la capitolele următoare. Am scris cartea astfel încât să o puteți citi de la un capăt la altul, la fel cum aș citi versiunea neabreviată a lucrării *The Stand* sau *The Decline and Fall of the Roman Empire*¹.

¹ Autor: Edward Gibbon. (n.t.)

Desigur, este bine dacă o carte ca aceasta servește atât ca material de referință, cât și ca povestire. Din acest motiv, multe dintre metodele, proprietățile și enumerările importante utilizate în programarea folosind Windows Forms sunt prezentate în tabele în cadrul capitolelor în care sunt discutate. Pe de altă parte, nici măcar o carte de asemenea dimensiuni nu poate spera să trateze *tot* ce este legat de Windows Forms. Aceasta nu reprezintă un substitut pentru documentația oficială a claselor.

Programele Windows Forms necesită o prezentare mai completă și, prin urmare, această carte include mai multe exemple de cod sub forma unor programe complete. Puteți copia și lipi fără restricții fragmente de cod din aceste exemple în propriile dumneavoastră programe. (În acest scop sunt create aceste programe.) Nu distribuiți însă codul sau programele ca atare. În acest scop, există cartea.

Compilerul de C# are o caracteristică teribilă, care vă permite să scrieți comentarii cu etichete XML. Totuși, am decis să nu folosesc această caracteristică. Programele din această carte au tendința să includă puține comentarii deoarece codul este descris în textul care înconjoară programele.

Așa cum probabil știți, Visual C# vă permite să proiectați interactiv aspectul aplicațiilor. Plasați diverse elemente de control (butoane, bare de derulare și altele) pe suprafața ferestrei dumneavoastră, iar Visual C# generează codul acestora. Deși asemenea tehnici sunt foarte utile în proiectarea rapidă a casetelor de dialog și a tipurilor de aplicații de interfață, în carte am ignorat această caracteristică din Visual C#.

În această carte nu vom lăsa Visual C# să genereze codul în locul nostru. În această carte vom învăța să scriem singuri programe.

Asistență

Au fost depuse toate eforturile pentru a asigura corectitudinea informațiilor din această carte. Microsoft Press furnizează erate ale cărților sale în World Wide Web, la adresa:

<http://www.microsoft.com/mspress/support>

Pentru a consulta direct Microsoft Press Knowledge Base și a formula o întrebare sau semnala o problemă cu care vă confrunțați, vizitați:

<http://www.microsoft.com/mspress/support>

Dacă aveți comentarii, întrebări sau idei legate de această carte vă rugăm să le trimiteți către Microsoft Press, pe oricare dintre următoarele căi:

Prin poștă:

Microsoft Press

ATTN: *Programming Microsoft Windows with C#* Editor

One Microsoft Way

Redmond, WA 980526399

E-mail:

MSPINPUT@MICROSOFT.COM

Asistență pentru produs nu se oferă la adresa de mai sus. Pentru informații de asistență privind C#, Visual Studio sau .NET Framework, vizitați site-ul Web Microsoft Product Support la adresa

<http://support.microsoft.com>

Mulțumiri deosebite

Scrisul este o activitate solitară dar, din fericire, întotdeauna există mai mulți oameni care fac munca să fie mai ușoară.

Doresc să mulțumesc agentului meu, Claudette Moore de la Moore Literary Agency pentru că a obținut acest proiect și s-a ocupat de toate aspectele juridice încâlcite.

Ca de obicei, a fost o deosebită plăcere să lucrez cu echipa de la Microsoft Press și încă o dată m-au ajutat să evit greșeli importante. Fără redactorul meu de proiect, Sally Stickney, și redactorul meu de specialitate, Jean Ross, această carte ar fi fost împânzită de text neinteligibil și cod cu erori.

Doresc să-l menționez și pe Johannes Brahms care a asigurat acompaniamentul muzical în timp ce lucram și pe Anthony Trollope pentru literatura de evadare citită seara.

Întâlnirile mele de duminică, marți și joi cu prietenii au continuat să mă ajute și să mă susțină moral în moduri uneori evidente, alteori subtile, dar întotdeauna inestimabile.

Mai mult decât oricât, doresc să-i mulțumesc logodnicei mele, Deirdre, pentru că mi-a oferit un mediu extrem de diferit (nu .NET) în care să trăiesc, să muncesc și să iubesc.

*Charles Petzold
New York
noiembrie 2001*