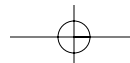
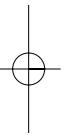
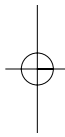
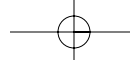




.NET iznutra



- 1. POGLAVLJE:** UVOD U .NET
- 2. POGLAVLJE:** ARIHITEKTURA .NETA-A
- 3. POGLAVLJE:** PROGRAMSKI JEZICI
- 4. POGLAVLJE:** RAZVOJNA OKOLINA



1. POGLAVLJE

Uvod u .NET

U ovom poglavlju:

- Osnovni koncept i snaga .NET-a
- Budućnost .NET-a

Računala su nesumnjivo promijenila naš život, i njihovo poznavanje nužno je za obavljanje bilo kakvog posla. Internet i računala omogućavaju ono što smo dosad mogli samo sanjati – od korištenja bankarskih usluga i potpunog upravljanja svojim novcem do rezervacije kino-ulaznica i kupovine svakojakih proizvoda.

Dok je sav napredak na području računarstva iznesen na leđima računalnih zanesenjaka i tehnološki orijentiranih ljudi, popularizaciju svega ipak možemo zahvaliti isključivo svakodnevnoj upotrebljivosti dostupne tehnologije. Ona se sve više okreće *običnim* ljudima kojima računala nisu nužna za obavljanje svakodnevnog posla. I tu leži ključ popularizacije tehnologije: u trenutku kad više nije potrebno čekati satima u redu u banci, izlaziti s posla da bi se obišlo dućane ili dolaziti mnogo ranije u kino da bi se uhvatila dobra mjesta, svatko želi znati koristiti računala.

No opet, biti programer u takvom svijetu sve je teže. Aplikacije se nalaze na svakom koraku i međusobno se veoma razlikuju – od aplikacija za mobitele i aplikacija za bankomate, preko knjigovodstvenih aplikacija i web-dućana do standardnih prozorskih aplikacija za pisanje teksta i slanje *e-mailova*. Želite li se baviti razvojem aplikacija, morate biti upoznati s izuzetno velikim brojem tehnologija i tehnika programiranja, kojih je svakim danom sve više, a često već postojeća znanja nisu upotrebljiva u novim okruženjima.

I. DIO: .NET IZNUTRA

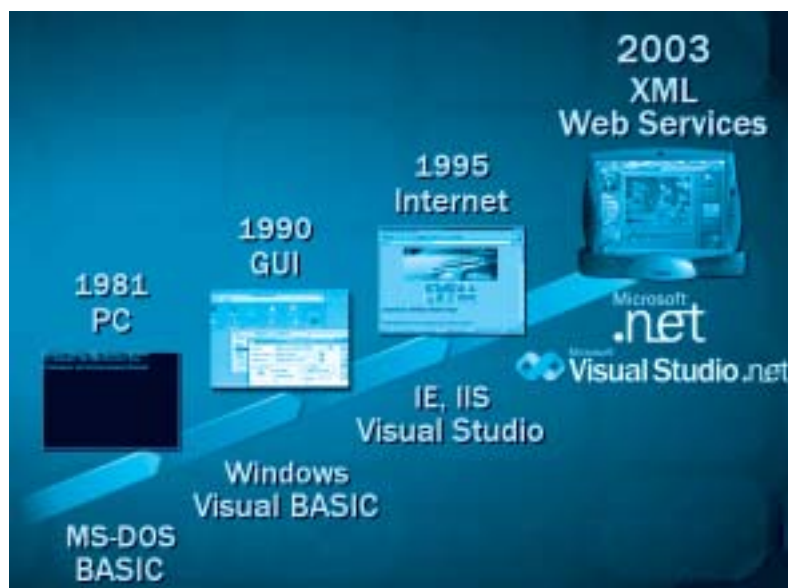
To je pak, izuzetno važno – tvrtkama je vrlo teško u stopu pratiti nove tehnologije zbog ograničenog znanja. Skupo je stalno mijenjati fokus i proučavati različite tehnologije, pa mnogi idu linijom manjeg otpora i slijepo se drže onog što znaju i što im dobro ide. Tako nije rijedak slučaj da se i danas razvijaju tekstualne aplikacije u okruženju DOS-a za upravljanje dućanima i knjigovodstvom.

Pri razvoju novih tehnologija mnogo se polaže na iskoristivost dostupnog znanja. Ukoliko je programerima zahvaljujući trenutnom poznavanju programskih jezika i tehnologija vrlo lako usvojiti nove tehnike programiranja, svi su zadovoljni – i programer koji osjeća da napreduje – i tvrtka koja dobiva kvalitetnije ljude i može obavljati modernije projekte, i – na kraju – sam proizvođač tehnologije, zbog prihvaćenosti njegovih modela.

Koncept .NET-a

Počnimo s objašnjenjem .NET-a. Radi se o Microsoftovoj platformi za razvoj aplikacija koja predstavlja ujedinjenje različitih tehnologija i načina programiranja. Tijekom povijesti su se programeri orijentirani Microsoftu susretali s gomilom tehnologija, kao što su MS-DOS, Windows, OLE, COM, COM+, ActiveX, MFC, ASP...

Slika 1-1:
Razvoj programiranja
tijekom godina



.NET, dakle, predstavlja ujedinjenje različitih koncepata. Želite li raditi web-aplikacije? Nema problema, programirajte u .NET-u. Želite li raditi obične *desktop*-aplikacije? Nema problema, programirajte u .NET-u. Želite li raditi aplikacije za mobilne uređaje, kao što su dlanovnici ili SmartPhone uređaji? Nema problema, programirajte u .NET-u.

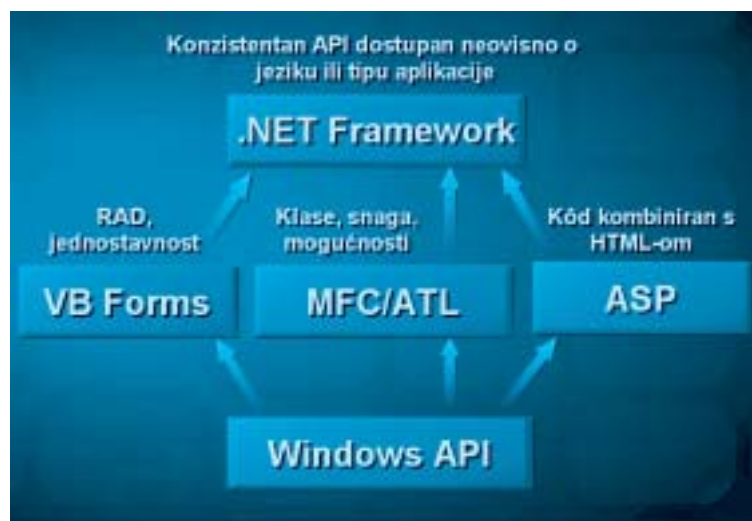
1. POGLAVLJE: UVOD U .NET

Za razvoj aplikacija za mobilne uređaje postoji posebna verzija .NET-a, nazvana .NET Compact Framework. Radi se verziji .NET-a iz koje su izbačene neke bazne klase koje nisu potrebne na mobilnim uređajima, no ona je inače u potpunosti funkcionalna i omogućava razvoj aplikacija za mobilne uređaje na jednak način kao i za Windowsse.



Više nema potrebe učiti različite tehnologije – ako znate programirati web-aplikacije u .NET-u, tada znate raditi i prozorske aplikacije za Windowsse. Tome je uvelike zaslužna unificirana razvojna okolina Visual Studio .NET koja sve to omogućava.

Visual Studio .NET predstavlja razvojnu okolinu za svaki tip .NET aplikacija, no i više od toga – mnogi Microsoftovi proizvodi koriste Visual Studio .NET kao razvojnu okolinu. Primjerice, želite li stvarati izvještaje koristeći Reporting Services dodatak za SQL Server 2000, koristit ćete Visual Studio .NET. Isto tako, želite li raditi s Commerce Serverom ili BizTalk Serverom, Visual Studio .NET postat će vam najbliži suradnik.



Slika 1-2:
.NET predstavlja
unifikaciju različitih
programskih modela.

Shvatili ste poantu, no to nije sve. .NET u sebi sadržava niz tehnologija, koje će redom biti objašnjene u ovoj knjizi, a sve one predstavljaju rezultat višegodišnjeg razvoja i unaprjeđenja postojećih tehnologija.

I. DIO: .NET IZNUTRA

Višejezičnost

Vratimo li se na primjer s početka teksta, .NET je učinio ogroman korak u smjeru olakšavanja prvih programerskih koraka na novoj platformi. Microsoft je, naime, standardizirao osnovne dijelove .NET Frameworka (postali su dio ISO i ECMA standarda), što pak omogućava nezavisnim proizvođačima da nadograđuju mogućnosti, kao što su novi jezici kojima se može programirati u .NET-u.

Microsoft je napravio podršku za nekoliko jezika u .NET-u – VB.NET, C++ .NET, J#, C# i JScript.NET – a zahvaljujući akademskoj zajednici i pojedinim industrijama, postoji više od 15 jezika u kojima se može programirati. Stoga, ukoliko ste iskusan programer u COBOL-u, Perlu, Fortranu, Eiffelu ili Smalltalku, bez straha – i vi možete lako početi programirati u .NET-u. Svima drugima, programerima s iskustvom na nekoj od Microsoftovih tehnologija, .NET će predstavljati samo korak naprijed.

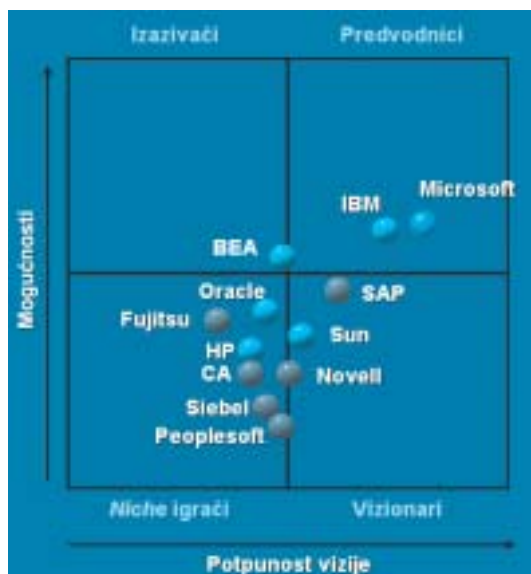
Stoga za .NET kažemo da je višejezičan, jer omogućava programiranje u velikom broju jezika, a zahvaljujući javno dostupnim standardima i vi možete napraviti podršku za neki svoj jezik. Višejezičnost podrazumijeva istu razvojnu okolinu (Visual Studio .NET) za bilo koji od programskih jezika, isti debugger za pronalaženje problema i njihovo otklanjanje iz kôda aplikacija.

Web-servisi

.NET donosi novost u programiranju na području web-servisa. Odmah razjasnimo – radi se o prihvaćenom standardu za komunikaciju između aplikacija pomoću SOAP poruka prenošenih Internetom (više o svemu saznat ćete u kasnijim poglavljima knjige). No snagu web-servisa predstavlja mogućnost integracije različitih sustava i aplikacija na standardizirani način za komunikaciju.

Slika 1-3:

Slika prikazuje razmještaj platformi za razvoj web-servisa prema Gartnerovom izvješću iz 2003. godine. Važno je uočiti da se samo Microsoft i IBM nalaze u magičnom kvadrantu odnosno među vodećim proizvođačima (primijetite i da Microsoft vodi). Znakovita je i anketa CIO Magazina iz 2003, prema kojoj od 369 CIO osoba čak 46,5% preferira Microsoftovu platformu .NET za razvoj web-servisa (drugi je IBM-ov WebSphere s 19%).



1. POGLAVLJE: UVOD U .NET

I ostali proizvođači imaju razvojne alate i okruženja za razvoj web-servisa, no izrađivanje web-servisa uz pomoć .NET-a slovi za najjednostavniji i najmoćniji način. Iskoristivost web-servisa velika je i veoma raširena – koriste se za dohvaćanje različitih informacija, a mogu se koristiti u različitim aplikacijama (primjerice, Microsoft Office 2003 može koristiti web-servise iz svog *Research and Reference* prozora).



Slika 1-4:
Web-servisi mogu uvelike olakšati razvoj aplikacija i njihovo povezivanje s drugim uslugama.

Uporaba je jednostavna: u svojoj aplikaciji koju izrađujete za Visual Studio .NET-u jednostavno dodate referencu na neki web-servis, upišete njegovu adresu i možete ga pozivati iz svoje aplikacije kao da se radi o bilo kakvom drugom servisu koji se ne nalazi negdje na Internetu. Njihova izrada je također jednostavna – razvojno okruženje će najveći dio posla obaviti za vas, a vama ostaje samo izraditi funkcionalnost web-servisa odnosno metode koje će moći izvršavati. No strpite se još malo, sve vas to čeka kasnije u knjizi.

Snaga .NET-a

Danas se još uvelike koriste DLL-ovi i druge COM komponente koje sadržavaju neku funkcionalnost. S njima se radi drugačije i često predstavljaju problem – zahvaljujući .NET-u, programski modeli su se unificirali i sa svime se može komunicirati standardnom objektno orijentiranom funkcionalnošću.

Također, zahvaljujući CLR-u, središnjem dijelu .NET-a, više nema potrebe za mučenjem s dosad obaveznim elementima u Win32 i COM programiranju, kao što su GUID-ovi, HRESULT i sučelja IUnknown. Ti elementi nisu skriveni od programera, već jednostavno više ne postoje – CLR ih u potpunosti zamjenjuje.

I. DIO: .NET IZNUTRA

Iako bi neupućeni promatrač na .NET mogao gledati kao na još jednu u nizu Microsoftovih tehnologija koje zahtijevaju isključivu posvećenost Microsoftovim proizvodima, uslugama i alatima, to nije točno. Naime, zbog već spomenute standardizacije CLR-a, .NET može biti implementiran na bilo kojoj platformi.

.NET je po nekim dijelovima sličan Javi – programi se prvo prevode u MSIL, a tek na ciljnom računalu pri prvom pokretanju se generira izvršni kôd. Iz toga slijedi logičan zaključak da se .NET aplikacije mogu izvršavati na bilo kojoj platformi na kojoj postoji verzija CLR-a kompatibilna sa standardom ECMA i bazne klase. Dakle, .NET može biti bez problema implementiran na IA64, Alpha, PowerPC računalima ili na bilo kojem operacijskom sustavu, kao što su MacOS, Linux ili Unix. Zapravo, u vrijeme pisanja ove knjige završava se projekt koji omogućava izvršavanje .NET aplikacija na Linuxu.



Kôd pisan u .NET-u često se naziva *managed* kôdom odnosno *upravljanim kôdom*, jer se izvršava u kontroliranom okruženju. Naravno, još uvijek možete pisati tzv. *unmanaged* kôd, no kako se .NET bude razvijao u narednim verzijama, to više neće biti moguće.

.NET se uvelike bazira i na XML-u, standardnom formatu za razmjenu podataka. Napredujući kroz ovu knjigu, uvidjet ćete da mnogi dijelovi .NET-a koriste baš XML za svoju internu uporabu, a ne treba ni spominjati web-servise, koji su u potpunosti temeljeni na XML-u.

S nabranjem snaga .NET-a mogli bismo nastaviti unedogled, stoga stavimo. Pravu snagu .NET-a uvidjet ćete čitajući naredna poglavlja ove knjige.

Budućnost .NET-a

Sve što ćete naučiti uz pomoć ove knjige i svih dostupnih resursa o .NET-u relativno je nevažno, ako .NET ne predstavlja važnu tehnologiju koju ćete moći još dugo vremena upotrebljavati. Naime, bilo bi potpuno besmisleno da utrošite godinu dana na svladavanje .NET-a i postanete vrhunski stručnjak pa doznate da Microsoft odustaje od daljnjeg korištenja .NET-a, da ga baca u sjenu i predstavlja neku novu tehnologiju.

Zahvaljujući svojoj snazi i poziciji na tržištu, Microsoft mora ponuditi plan razvitka svojih tehnologija i dugoročnu obavezu, a njegova je poruka jasna – .NET će bit s nama još jako dugo. Trenutno je aktualna verzija 1.1 .NET-a, a sljedeća (2.0) nosi kodno ime Whidbey. No Microsoft ima mnogo dugoročniji plan – sljedeća verzija klijentskog operacijskog sustava, kodnog imena Longhorn, čiji se izlazak planira u 2006. godini, u potpunosti će se bazirati na *managed* kodu odnosno na .NET-u.

1. POGLAVLJE: UVOD U .NET



Slika 1-5:
Razvoj koncepata programiranja vodi prema uslugama.

Zar uopće trebate bolju potvrdu Microsoftove predanosti .NET-u? Sama činjenica da će cijeli operacijski sustav biti izrađen oko .NET-a dovoljno govori o njegovoj uporabljivosti u stvarnom svijetu. Da biste se pripremili za vrijeme koje nam dolazi, pišite *managed* kôd i koristite .NET za stvaranje aplikacija.

Hrabro zakoračite u svijet .NET-a! Izabrali ste pravu programersku platformu, koja će vam omogućiti da lako razvijete sve svoje mogućnosti i stvarate aplikacije po vlastitoj želji. Web-aplikacije, web-servisi, *desktop* aplikacije, aplikacije za mobilne uređaje – izbor je samo vaš.

