



# Deo

## **Visual Studio okruženje**

- Čas 1      "Turneja" kroz C# programiranje
- Čas 2      Navigacija C#-om
- Čas 3      Razumevanje objekata i kolekcija
- Čas 4      Razumevanje događaja





# Čas /.

## "Turneja" kroz C# programiranje

**UČENJE NOVOG PROGRAMSKOG JEZIKA MOŽE BITI ZASTRAŠUJUĆE.** Ukoliko nikada ranije niste programirali, akt kucanja naizgled tajnog teksta da bi se proizvele uređene i moćne aplikacije verovatno izgleda kao crna umetnost i možda se pitate kako ćete ikada naučiti sve što je potrebno da znate. Odgovor je, naravno: korak po korak. Prvi korak u učenju jezika je isti kao i kod bilo koje druge aktivnosti - stvaranje samopouzdanja. Programiranje je delom umetnost, a delom nauka. Iako može izgledati kao magija, to je više iluzija; kada naučite kako "stvari" rade, uklanja se mnogo mističizma, oslobadajući Vas da se usredsredite na mehaniku neophodnu da bi se proizveo željeni rezultat.

U ovom času kompletiraćete brzu "turneju" koja Vas vodi korak po korak kroz kreiranje kompletnog, iako malog C# programa. Do sada nisam video "Hello World" program koji je i najmanje koristan (obično ne radi ništa više osim što ispisuje "hello world" na ekranu). Umesto toga, kreiraćete program za pregled slika koji omogućava da pregledate Windows bitmap slike i ikone na svom računaru. Naučićete kako da omogućite korisniku da traži fajl i kako da prikazuje izabrani fajl slike na ekranu, što su veštine koje će biti pogodne u kasnijim aplikacijama koje ćete kreirati. Kreiranje velikih komercijalnih solucija postiže se nizom malih koraka. Nakon što završite kreiranje ovog malog projekta, steći ćete ukupan "osećaj" za razvojni proces.



## Čas 1.

Iz ovog časa posebno ističemo sledeće:

- izrada jednostavne, a, ipak, funkcionalne C# aplikacije
- omogućavanje korisniku da pretražuje hard disk
- prikazivanje slike iz fajla sa diska
- upoznavanje dela programerskog žargona
- učenje o Visual Studio - C# IDE-u.

Nadam se da ćete do kraja ovog časa shvatiti koliko je zabavno programirati upotrebom C#-a.

## Pokretanje C#-a

Morate se navići na neke termine pre nego što počnete da kreirate programe u C#-u:

- **Distributivna komponenta** To je finalna, kompajlirana verzija projekta. Komponente mogu biti distribuirane drugim ljudima i drugim računarima i ne zahtevaju C# da bi se izvršavale (iako je .NET "kostur" potreban, o čemu ćete naučiti u narednim časovima). Distributivne komponente se nazivaju i programi. Na Času 22, "Razvoj soluciјe," naučićete kako da distribuirate drugim kompjuterima program za pregledanje slika koji počinjete da kreirate.
- **Projekat** To je kolekcija fajlova koja može biti kompajlirana da bi bila kreirana distributivna komponenta (program). Postoji mnogo tipova projekata i kompleksne aplikacije mogu sadržati više projekata, kao što su Windows Application (Windows aplikacija) projekat i DLL projekti podrške.
- **Solucija** To je kolekcija projekata i fajlova koji čine aplikaciju, ili komponentu.

### NOVI TERMIN

C# je, takođe, kompletno razvojno okruženje; svaki alat koji Vam je potreban da biste kreirali C# projekte je dostupan iz Visual Studia. Visual Studio - C# okruženje je nazvano IDE, što predstavlja skraćenicu od Integrated Development Environment (integrisano razvojno okruženje); ono je dizajnerski "kostur" u kome gradite aplikacije. Da biste radili sa C# projektima, morate prvo pokrenuti Visual Studio IDE.

Sada pokrenite C# IDE, tako što ćete izabrati Microsoft Visual Studio .NET iz Microsoft Visual Studio .NET fascikle iz Start menija.



## Kreiranje novog projekta

Kada prvi put pokrenete Visual Studio .NET, prikazaće vam se Visual Studio Start Page (početna strana) kartica iz IDE-a. Upotreboom ove strane možete otvoriti ranije kreirane projekte, ili kreirati nove (videti sliku 1.1). Za ovu brzu "turneju" kreirajte novu Windows aplikaciju, pa kliknite New Project (novi projekt) da bi bio prikazan okvir za dijalog New Project, koji je predstavljen na slici 1.2.

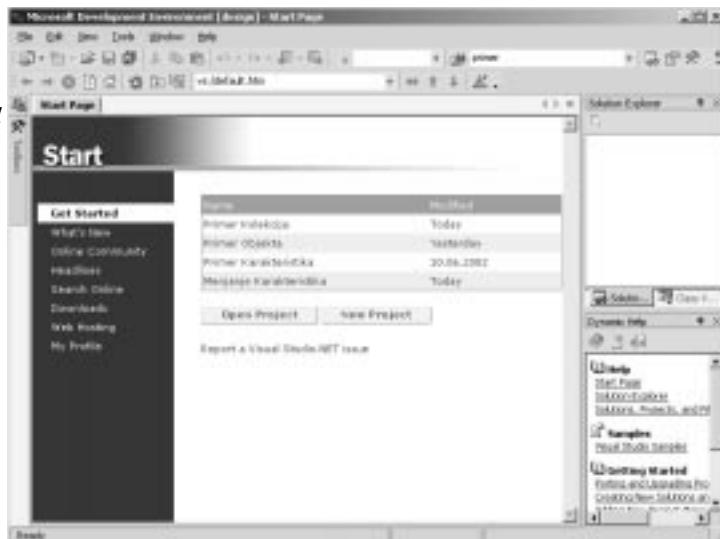


### Napomena

Ukoliko ne vidite Visual Studio Start stranu, verovatno ste promenili osnovno podešavanje. Na Času 2, "Navigacija C#-om", videćete kako da vratite to podešavanje na početnu vrednost. Za sada, znajte da možete kreirati novi projekat iz File menija, osim što upotrebljavate tehnike opisane na ovom času.

Slika 1.1

Možete otvoriti postojeće, ili kreirati nove projekte iz Visual Studio Start strane.

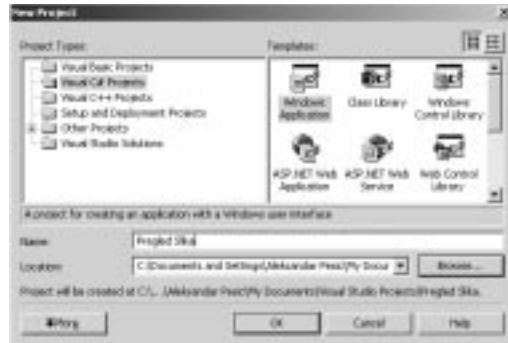




## Čas 1.

### Slika 1.2

Okvir za dijalog New Project omogućava da kreirate mnoge tipove .NET projekata.



Možete kreirati mnoge tipove projekata sa C#-om, kao i sa drugim podržanim jezicima .NET platforme. Okvir za dijalog New Project se koristi za specificiranje tipa C# projekta koji želite da kreirate. Ukoliko Visual C# Projects fascikla nije izabrana, kliknite je da biste prikazali C# tipove projekata, pa proverite da li je izabrana ikona Windows Application (ukoliko nije, kliknite je da biste je izabrali). Na dnu okvira za dijalog New Project je polje za unos teksta Name (ime), u kome zadajete ime projekta koji kreirate; u polju za unost teksta Location možete uneti lokaciju na koju ćete snimiti fajlove projekta.



### Savet

Trebalo bi da uvek postavite ove vrednosti na nešto smisleno pre nego što kreirate projekat, ili ćete imati više posla kasnije ukoliko poželite da premestite projekat, ili da mu promenite ime.

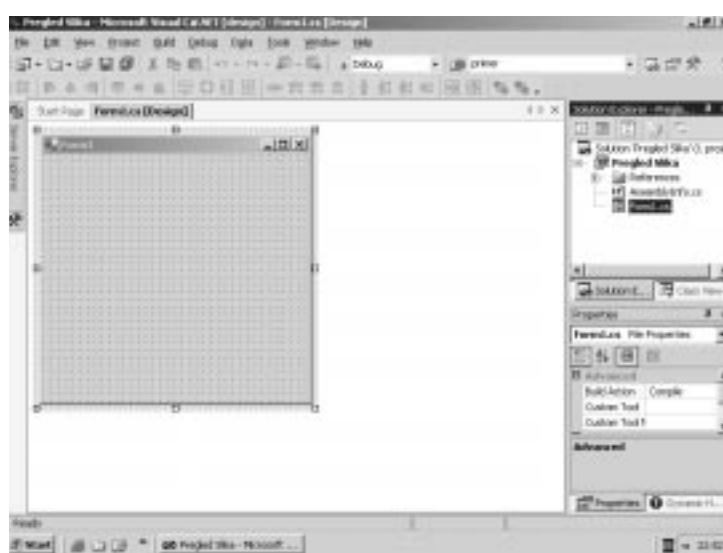
Ukucajte **Picture Viewer** u polje za unos teksta Name da biste imenovali svoj projekat. Nije potrebno da u ovom trenutku menjate lokaciju gde će fajlovi projekta biti snimljeni, pa nastavite dalje i kreirajte novi Windows Application projekat, tako što ćete kliknuti OK. C# kreira novi projekat, upotpunjeno jednom formom (u dizajn prozoru) da biste počeli da izrađujete interfejs za Vašu aplikaciju (videti sliku 1.3).

## “Turneja” kroz C# programiranje



**Slika 1.3**

Nove Windows aplikacije počinju praznom formom; "zabava" je upravo počela!



Vaše C# okruženje može izgledati drugačije od ovog koje je pokazano na slikama na ovom času, u zavisnosti od edicije C# koju koristite (ili od toga da li ste se već "igrali" C#-om) i drugih činilaca, kao što je rezolucija monitora. Svi elementi predstavljeni na ovom času, međutim, postoje u svim edicijama C#-a. (ukoliko Vaš IDE nema prikazan prozor sa slike, upotrebite View meni da biste ga prikazali).



### Napomena

Da biste kreirali program koji može biti izvršavan na drugom računaru, počnite sa kreiranjem projekta, pa kompajlirajte projekat u komponentu, kao što je, na primer, izvršiva (program koji korisnik može izvršiti), ili DLL (komponenta koja može biti korišćena u drugim programima i komponentama). Proces kompajliranja je detaljnije obrađen na Času 22, "Razvoj solucije". Važno je napomenuti u ovom trenutku: kada čujete da neko govori o kreiranju, ili pisanju programa, upravo kao što Vi sada kreirate program za pregled slika, on govori o kompletiranju svih koraka do kompajliranja projekta u distributivni fajl, uključujući i kompajliranje.



Čas 1.

## Razumevanje C# okruženja

Kada prvi put pokrenete C#, primetićete da IDE sadrži mnogo prozora, kao što je prozor karakteristika (Properties) sa desne strane, koji se koristi za pregled i postavljanje karakteristika objekata. Osim ovih prozora, IDE sadrži mnogo kartica, kao što je kartica okvira sa alatom na levoj (Toolbox) ivici IDE-a (videti sliku 1.3). Klik na karticu prikazuje odgovarajući prozor. Probajte sada sledeće: kliknite karticu okvira sa alatom da biste prikazali prozor okvira sa alatom. Osim toga, možete postaviti kurSOR miša iznad kartice na nekoliko sekundi da biste prikazali prozor. Da biste sakrili prozor, jednostavno, pomerite miša van njega. Da biste kompletno zatvorili prozor, kliknite dugme za zatvaranje (X) u naslovnoj liniji prozora.

Možete podesiti poziciju i veličinu svakog od ovih prozora, koje možete, čak, sakrivati i prikazivati prema želji. Naučićeće kako da posebno prilagodite Vaše dizajnersko okruženje na Času 2, "Navigacija C#-om".



### Savet

*Nemojte duplo kliknuti bilo šta u C# dizajnerskom okruženju, osim ukoliko Vam se da instrukcija da uradite tako. Dupli klik na većini objekata izaziva potpuno drugačiji ishod od jednog klika. Ukoliko greškom duplo kliknete objekat sa forme, prikazuje se prozor koda. Na vrhu prozora koda je set kartica: jedna je za dizajniranje forme i jedna za kod. Kliknite karticu za dizajniranje forme da biste sakrili prozor koda i vratili se u formu.*

Prozor karakteristika sa desne strane dizajnerskog okruženja je možda najvažniji prozor; on je jedan od onih koje koristite najčešće. Ukoliko je prikaz Vašeg kompjutera postavljen na 640x480, možete verovatno videti samo nekoliko karakteristika u jednom trenutku. To čini teškim pregled i postavljanje karakteristika dok kreirate projekte. Preporučujem da ne pokušavate razvoj sa Visual Studiom na rezoluciji ispod 800x600; ja preferiram 1024x768, jer ona nudi obilje radnog prostora. Da biste promenili Vaše podešavanje prikazivanja, desnim tasterom miša kliknite Desktop i izaberite Properties.

## Promena karakteristika objekata

Skoro sve sa čime radite u C#-u je objekat. Forme su, na primer, objekti, kao i sve stavke koje možete postaviti na formu da biste izradili interfejs, kao što su polja liste i dugmad. Postoji mnogo tipova objekata (o njima će više reći biti na Času 3, "Razumevanje objekata i kolekcija"). Objekti su, nadalje, klasifikovani po tipovima. Na primer, forma je objekat forme, dok su stavke koje možete postaviti na formu nazvane objekti kontrola, ili kontrole. Neki objekti nemaju fizičku pojavu, već egzistiraju samo u kodu. Naučićeće o ovoj vrsti objekata na kasnijim časovima.



### NOVI TERMIN

Svaki objekat, bez obzira da li ima fizičku pojavu, ima poseban set atributa poznatih kao karakteristike (*properties*). Kao što Vi imate određene karakteristike, kao što su, na primer, boja kose i visina, imaju ih i C# objekti, kao što su Height (visina) i BackColor (boja pozadine). Karakteristike definišu karakteristike objekta. Kada kreirate novi objekat, prvo što treba da uradite je da postavite njegove karakteristike, tako da se on pojavi i ponaša na način na koji želite. Da biste prikazali njegove karakteristike, kliknite objekat u njegovom dizajneru. Kliknite sada formu da biste se osigurali da su njene karakteristike prikazane u prozoru karakteristika.

## Imenovanje objekata

Karakteristika koju prvo treba da postavite za bilo koji novi objekat je Name (ime). Pritisnite F4 da biste prikazali prozor karakteristika (ukoliko već nije vidljiv) i obratite pažnju na ime dato Vašoj osnovnoj formi (prva karakteristika navedena u prozoru karakteristika) - Form1. Kada prvi put kreirate objekat, C# mu daje jedinstveno generičko ime, zasnovano na tipu objekta. Iako su ova imena funkcionalna, nisu previše opisna. Na primer, C# je imenovao Vašu formu Form1, ali je čest slučaj mnoštvo formi u projektu i bilo bi izuzetno teško upravljati komplikovanim projektom ukoliko se forme razlikuju samo po broju (Form2, Form3 i tako dalje).



### Napomena

*U stvarnosti, ono što kreirate je klasa forme, ili šablon, koji će biti korišćeni za kreiranje i prikazivanje formi u vreme izvršenja. U svrhu ove brze "turneje", ja, jednostavno, govorim o njoj kao o formi (više informacija naći ćete na Času 5, "Izrada formi - deo I").*

Da biste bolje kontrolisali svoje forme, treba da svakoj date opisno ime. C# Vam daje šansu da imenujete nove forme kada se kreiraju. Pošto je C# kreirao ovu osnovnu formu umesto Vas, niste imali šansu da je imenujete, tako da morate promeniti i ime fajla i ime forme. Promenite ime forme tako što ćete kliknuti Name karakteristiku i promeniti tekst od Form1 u **fclsViewer**. Obratite pažnju da ovo nije promenilo ime fajla forme koji je prikazan u Solution Explorer prozoru. Sada promenite ime fajla tako što ćete desnim tasterom miša kliknuti Form1.cs u Solution Explorer prozoru, izabrati Rename (promeni ime) iz kontekstnog menija i promeniti tekst iz Form1.cs u **fclsViewer.cs**. U narednim primerima neću Vas primoravati da promenite ime fajla svaki put, jer ćete imati dovoljno koraka koje treba izvršiti. Međutim, preporučujem da uvek promenite imena fajlova u nešto simisleno u Vašim "pravim" projektima.



### Čas 1.



#### Napomena

Ja ovde koristim `fcls` prefiks da bih označio da je fajl klasa forme. Postoje različiti tipovi klase, tako da upotreba prefiksa pomaže da razlikujete klase u kodu. C# ne zahteva da koristite prefikse objekata, ali ja preporučujem da uradite tako. Na Času 12, "Upotreba konstanti, tipova podataka, promenljivih i nizova", naučićete prednosti upotrebe konvencije imenovanja i standardnih prefiksa za mnoge .NET objekte.

## Postavljanje Text karakteristike forme

Obratite pažnju da tekst koji se pojavljuje u naslovnoj liniji forme "kaže" Form1. Ovo je zbog toga što C# postavlja tekst naslovne linije forme u ime forme kada je prvi put kreirana, ali ga ne menja kada Vi promenite ime forme. Tekst u naslovnoj liniji je određen vrednošću Text karakteristike forme. Kliknite formu još jednom, tako da se njene karakteristike pojave u prozoru karakteristika. Koristite klizač u prozoru karakteristika da biste locirali Text karakteristiku u prozoru karakteristika, pa promenite tekst u **Pregled slike**.

## Davanje ikone formi

Svako ko je koristio Windows poznaje ikone, sličice korišćene za predstavljanje programa. One se moraju često pojaviti u Start meniju pored imena njihovih odgovarajućih programa. U C#-u nemate kontrolu samo nad ikonama Vašeg programskog fajla, već možete, ukoliko želite, dati svakoj formi u Vašem programu jedinstvenu ikonu.



#### Napomena

Instrukcije koje slede podrazumevaju da imate pristup izvornim fajlovima za primere u ovoj knjizi. Oni su dostupni na [www.samspublishing.com/detail\\_sams.cfm? item=0672320800](http://www.samspublishing.com/detail_sams.cfm?item=0672320800). Ne morate koristiti ikonu koju sam ja obezbedio za ovaj primer, već bilo koju, po Vašem izboru. Ukoliko nemate dostupnu ikonu, možete "preskočiti" ovaj odeljak, bez uticaja na ishod pristema.

Da biste dodelili ikonu formi, pratite naredne korake:

1. U prozoru karakteristika kliknite Icon karakteristiku da biste je izabrali.
2. Kada kliknete Icon karakteristiku, malo dugme sa tri tačke se pojavljuje desno od karakteristike. Kliknite ga.
3. Da biste locirali HourOne.ico fajl, ili neki drugi ico fajl po Vašem izboru, koristite okvir za dijalog Open koji se pojavljuje. Kada ste pronašli ikonu, duplo je kliknite, ili je kliknite jednom da biste je izabrali, pa kliknite Open.



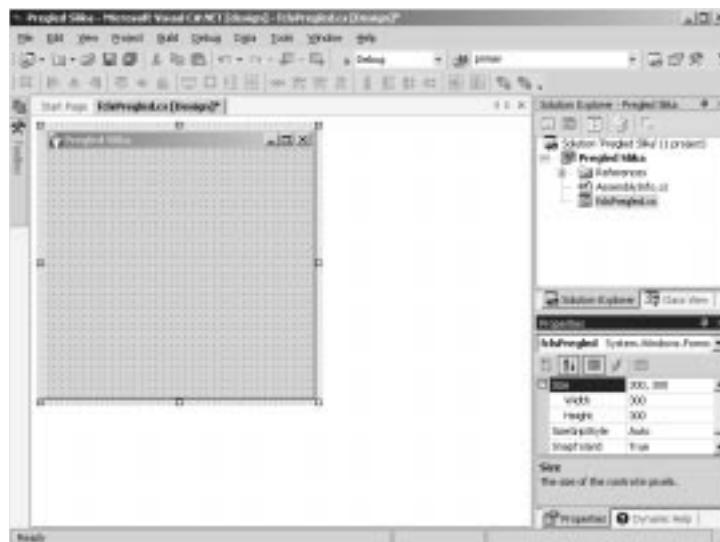
Nakon što ste izabrali ikonu, ona se pojavljuje u Icon karakteristici zajedno sa rečju (Icon). Mala verzija ikone se, takođe, pojavljuje u gornjem levom uglu forme. Bilo kada da je ova forma minimizirana, to je ikona koja će biti prikazana u Windows taskbaru. (napomena: ovo ne menja ikonu projekta kao celine; na Času 22 naučićete kako da dodelite ikonu Vašem distributivnom fajlu).

## Promena veličine forme

Zatim ćete promeniti Width (širina) i Height (visina) karakteristike forme. Height i Width vrednosti su prikazane skupa pod Size karakteristikom; Width se pojavljuje sa leve strane zareza, a Height sa desne. Možete promeniti Width i Height, tako što ćete promeniti odgovarajući broj u Size karakteristici. Obe vrednosti predstavljaju broj piksela dimenzije. Da biste prikazali i podesili ove dve karakteristike odvojeno, kliknite mali znak plus (+) pored karakteristike Size (videti sliku 1.4).

Slika 1.4

Neke karakteristike mogu biti proširene da bi prikazale specifičnije karakteristike.

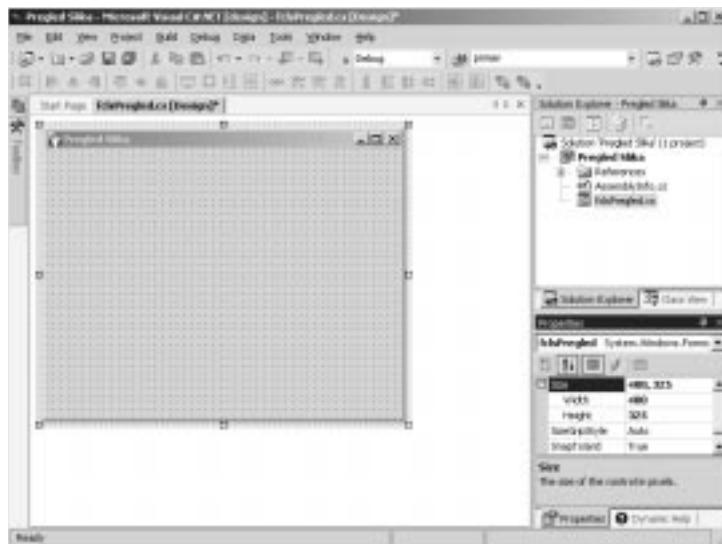


Promenite Width karakteristiku u **400** i Height u **325**. Da biste potvrdili promenu karakteristike, pritisnite Tab, ili kliknite drugu karakteristiku, ili drugi prozor. Ekran bi sada trebalo da izgleda kao onaj na slici 1.5.



## Čas 1.

**Slika 1.5**  
Promena u prozoru karakteristika je odražena čim se promena potvrdi.



Kada prvi put kreirate ovaj projekat, C# snima kopiju izvornih fajlova u njihovom početnom stanju. Promene koje ste do sada izvršili postoje samo u memoriji; ukoliko hoćete da isključite kompjuter u ovom trenutku (nemojte to da radite), izgubićete sav rad do ovog trenutka. Trebalo bi da steknete naviku čestog snimanja rada. Snimite sada projekat, tako što ćete izabrati Save All (snimi sve) iz File menija, ili tako što ćete kliknuti Save All dugme na paleti sa alatom (ima na sebi sliku naslaganih disketa).

## Dodavanje kontrola formi

### NOVI TERMIN

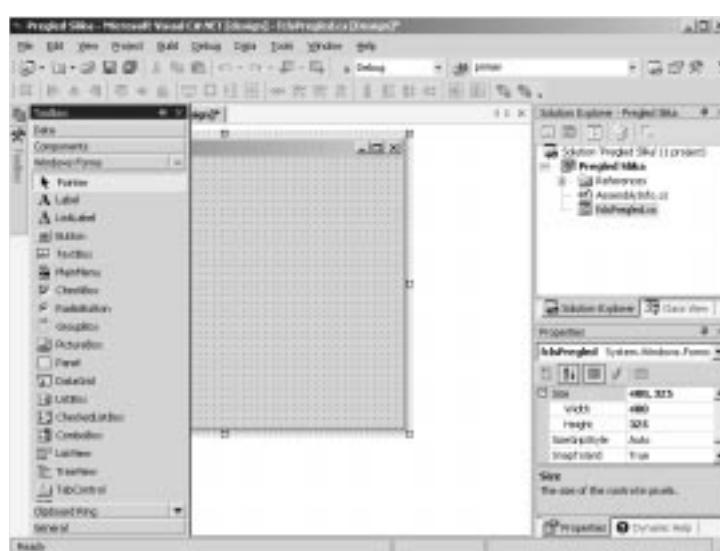
Sada, kada su karakteristike Vaše forme postavljene, potrebno je da dodate objekte formi da bi proizveli korisnički interfejs. Objekti koji mogu biti smešteni na forme nazivaju se *kontrole*. Neke kontrole imaju vidljivi interfejs sa kojim korisnik može interagovati, dok su ostale uvek nevidljive za korisnika. Koristićete kontrole oba tipa u ovom primeru. Sa leve strane ekrana je kartica okvira sa alatom (Toolbox). Kliknite sada karticu okvira sa alatom da biste prikazali prozor okvira sa alatom (videti sliku 1.6). Okvir sa alatom sadrži sve kontrole dostupne u projektu, kao što su oznake i polja za unos teksta.

## “Turneja” kroz C# programiranje



**Slika 1.6**

Okvir sa alatom se koristi da bi bile izabrane kontrole za izradu korisničkog interfejsa.



### Savet

Možete dodati kontrolu formi na tri načina (detaljno su objašnjeni na Času 5). U ovom času koristite tehniku duplog klika na alatu u okviru sa alatom.

Okvir sa alatom se zatvara sam čim ste dodali kontrolu formi i kurzor miša nije više iznad okvira sa alatom. Da biste učinili da okvir sa alatom ostane vidljiv, kliknite sličicu pribadače, lociranu u naslovnoj liniji okvira sa alatom.



### Napomena

Za više informacija o posebnom prilagođavanju dizajnerskog okruženja pogledajte Čas 2, "Navigacija C#-om".

Interfejs Vašeg programa za pregled slika će se sastojati od narednih kontrola:

- dve Button (dugme) kontrole
- PictureBox (okvir slike) kontrole
- OpenFileDialog kontrole



Čas 1.

## Dizajniranje interfejsa

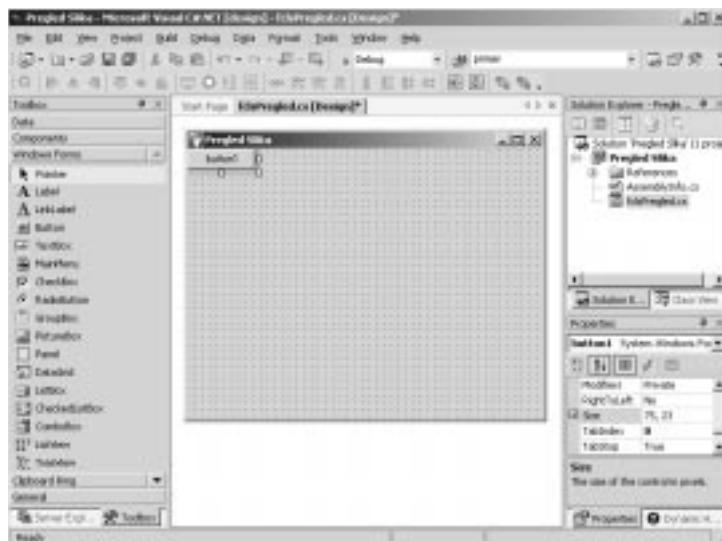
Uopšteno govoreći, najbolje je dizajnirati korisnički interfejs forme, pa dodati kod u pozadini interfejsa koji čini formu funkcionalnom. Korisnički interfejs za Vaš program za pregled slika će se sastojati od dugmeta za izbor slike, dugmeta za zatvaranje i okvira u kome će biti prikazana slika.

### Dodavanje vidljive kontrole forme

Počnite tako što ćete dodati Button kontrolu formi. Uradite to tako što ćete duplo kliknuti stavku dugmeta u okviru sa alatom. C#, zatim, kreira novo dugme i postavlja ga u gornji levi ugao forme (videti sliku 1.7).

Slika 1.7

Kada duplo kliknete kontrolu u okviru sa alatima, ona je dodata u gornji levi ugao forme.



Upotrebom prozora karakteristika postavite karakteristike dugmeta na sledeći način (obratite pažnju da ćete možda poželeti da promenite listu karakteristika u abecedni redosled, ukoliko već nije u njemu, da biste olakšali pronalaženje ovih karakteristika po imenu):

## “Turneja” kroz C# programiranje



Karakteristika	Vrednost
Name	btnSelectPicture
Text	Izabri sliku
Location	301, 10 (napomena: 301 je x koordinata, a 10 je y koordinata)
Size	85, 23

Sada ćete kreirati dugme koje korisnik može koristiti da bi zatvorio program za pregled slika. Umesto da dodajete novo dugme formi, kreiraćete kopiju dugmeta koje ste već definisali. Da biste to uradili, desnim tasterom miša kliknite dugme na formi i izaberite Copy iz menija prečice. Zatim, desnim tasterom miša kliknite bilo gde na formi i izaberite Paste iz menija prečice forme. Novo dugme se pojavljuje preko dugmeta koje ste kopirali; ono je izabrano po osnovnom podešavanju. Promenite karakteristike novog dugmeta na sledeći način:

Karakteristika	Vrednost
Name	btnQuit
Text	kraj
Location	301, 40

Poslednja kontrola koju je potrebno da dodate formi je PictureBox. Ova kontrola ima mnogo mogućnosti, ali je njena osnovna namena prikazivanje slika, što je upravo ono za šta ćete je Vi koristiti u ovom primeru. Dodajte novu PictureBox kontrolu formi i postavite njene karakteristike kao što sledi:

Karakteristika	Vrednost
Name	picShowPicture
BorderStyle	FixedSingle
Location	8, 8
Size	282, 275

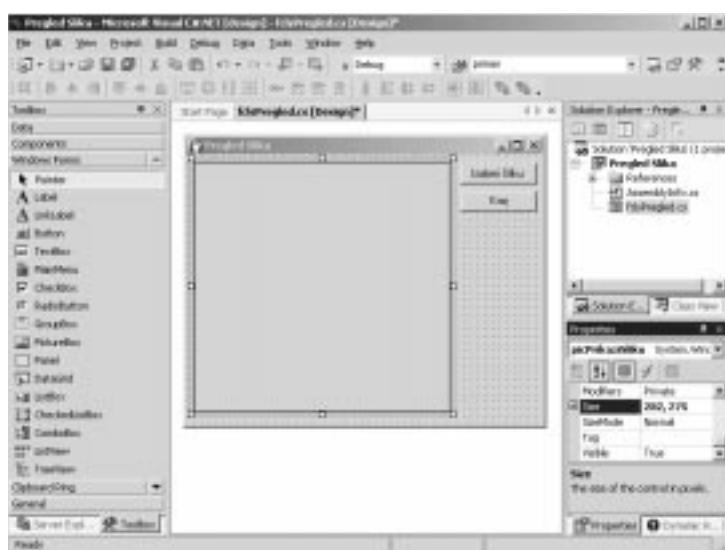
Nakon što ste izvršili ove promene karakteristika, Vaša forma će izgledati poput one na slici 1.8. Kliknite Save All dugme na paleti sa alatom da biste snimili svoj rad.



## Čas 1.

**Slika 1.8**

*Interfejs aplikacije ne mora biti kompleksan da bi bio koristan.*



## Dodavanje nevidljive kontrole formi

### NOVI TERMIN

Sve kontrole koje ste do sada koristili nalaze se na formi i imaju fizičku pojavu. Međutim, nemaju sve fizičku pojavu. Ovakve kontrole, o kojima govorimo kao o *kontrolama nevidljivim za vreme izvršavanja*, nisu dizajnirane za interakciju sa korisnikom, već da bi dale programeru funkcionalnost izvan osnovnih mogućnosti C#-a.

Da biste omogućili korisniku da izabere sliku koju će prikazati, potrebno je da mu pružite mogućnost da locira fajl na svom hard disku. Verovatno ste primetili ranije da, kada god izaberete da otvorite fajl iz bilo koje Windows aplikacije, okvir za dijalog koji se prikaze je skoro uvek isti. Nema nikakvog smisla prisiljavati svakog razvojnog programera da piše kod neophodan za izvođenje standardnih operacija sa fajlovima. Umesto toga, Microsoft je izložio tu funkciju kroz kontrolu koju možete koristiti u Vašem projektu. Ova kontrola se zove OpenFileDialog; uštедеće Vam mnogo sati koje biste, u suprotnom, utrošili pri pokušajima da duplirate zajedničke funkcije.



### Napomena

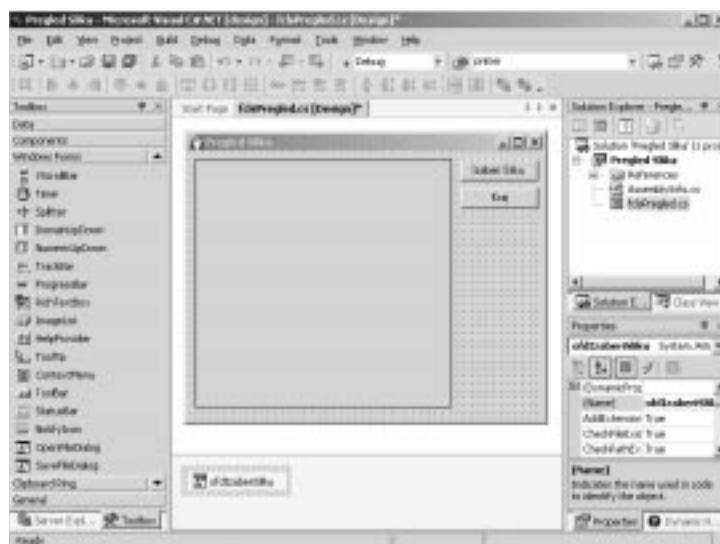
Ostale kontrole, osim OpenFileDialog kontrolе, daju Vam fajl funkcionalnost. Na primer, SaveFileDialog kontrola obezbeđuje mogućnosti korisniku da snimi fajl.



Skrolujte okvir sa alatom dok ne budete mogli da vidite OpenFileDialog kontrolu, a zatim je duplo kliknite da biste je dodali formi. Možda ćete morati da skrolujete okvir sa alatom, što ćete uraditi tako što ćete kliknuti strelicu nagore da biste se kretali ka vrhu, ili strelicu nadole da biste se kretali prema dnu. Obratite pažnju da kontrola nije postavljena na formu, već se pojavljuje u posebnoj oblasti ispod forme (videti sliku 1.9). Ovo se dešava zato što OpenFileDialog kontrola nema interfejs za prikazivanje korisniku. Ona ima interfejs, okvir za dijalog koji možete prikazati kada je potrebno, ali nema ništa za prikazivanje direktno na formi.

Slika 1.9

Kontrole koje nemaju interfejs se pojavljuju ispod dizajnera forme.



Izaberite OpenFileDialog kontrolu da biste promenili njene karakteristike na sledeći način:

Karakteristika	Vrednost
Name	ofdSelectPicture
Filter	Windows Bitmaps   *.BMP   JPEG Files   *.JPG
Title	izaberi sliku

Filter karakteristika određuje filtriranje kontrole. Tekst koji se pojavljuje pre simbola vertikalne crte (|) je opisni tekst tipa fajla, dok je tekst nakon simbola vertikalne crte obrazac koji treba koristiti za filtriranje fajlova. Možete zadati više od jednog tipa filtra. Tekst unet u Title karakteristiku se pojavljuje u naslovnoj liniji okvira za dijalog Open File



## Čas 1.

# Kodiranje interfejsa

Grafički interfejs za Vaš program za pregled slika je sada završen, pa kliknite pribadač u naslovnoj liniji okvira sa alatom da biste ga zatvorili. Sada treba da napišete kod za program da bi bio sposoban da izvršava akcije. C# je događajima vođeni jezik, što znači da se kod izvršava kao odgovor na događaje. Ovi događaji mogu poticati od korisnika, kao što je korisnikov klik dugmeta, ili od samog Windowsa (za kompletno objašnjenje događaja videti Čas 4, "Razumevanje događaja"). Trenutno, Vaša aplikacija izgleda lepo, ali ne radi ništa. Korisnik može kliknati dugme Izaberi sliku, na primer, dok krave ne progovore, ali se ništa neće desiti, jer niste rekli programu šta da radi kada korisnik klikne dugme.

Sada ćete napisati kod za izvršavanje dva zadatka. Prvo ćete napisati kod koji omogućava korisniku da pretražuje svoj hard disk da bi locirao i izabrao fajl slike, pa prikazuje fajl u okviru za sliku (ovo zvuči teže nego što, u stvari, jeste). Drugo, dodaćete kod Kraj dugmetu koji zatvara program kada korisnik klikne dugme.

## Omogućavanje korisniku da traži fajl

Prvi komadić koda koji ćete napisati će omogućiti korisniku da pretražuje svoj hard disk, izabere fajl i prikaže izabranu sliku u PictureBox kontroli. Ovaj kod će se izvršavati kada korisnik klikne dugme Izaberi sliku; zbog toga je on dodat Click događaju tog dugmeta (naučićeće sve o događajima na kasnijim časovima). Kada duplo kliknete kontrolu na formi u dizajn pogledu, kreira se osnovni događaj za tu kontrolu i prikazuje se u prozoru koda. Osnovni događaj za Button kontrolu je Click događaj, što ima smisla, jer je klik dugmeta njegova najčešća svrha. Duplo kliknite dugme Izaberi sliku da biste pristupili njegovom Click događaju u prozoru koda (videti sliku 1.10).

### NOVI TERMIN

Kada pristupite događaju, C# kreira *manipulator događaja* (event handler), što je, u suštini, šablon procedure u koju dodajete kod koji se izvršava kada je događaj nastupio. Kursor je već smešten u kod procedure, pa treba samo da dodate kod. Takođe ćete primetiti da su otvorena i zatvorena vitičasta zagrada postavljene za Vašu novu proceduru događaja. Vitičaste zgrade u ovom slučaju definišu početak i kraj Vaše procedure. Uskoro ćete videti da C# zahteva mnoge otvorene i zatvorene vitičaste zgrade (šč). Do vremena kada završite proučavanje ove knjige "do ludila" ćete kliknati van dok pišete kod da biste prisilili aplikaciju da radi tačno ono što želite da radi. Za sada, samo unesite kod kako sam ga ja ovde predstavio.

## “Turneja” kroz C# programiranje



**Slika 1.10**

Napisáete sav kod u prozoru kao što je ovaj.

```
using System;
using System.Windows.Forms;

namespace Turneja
{
    static class Program
    {
        /// <summary>
        /// The main entry point for the application.
        /// </summary>
        [STAThread]
        static void Main()
        {
            Application.EnableVisualStyles();
            Application.Run(new Form1());
        }
    }
}
```

Vrlo je važno da steknete naviku komentarisanja svog koda, tako da je prvi red koji ćete uneti komentar. Započinjanje izraza znacima // označava izraz kao komentar; kompjuter neće raditi ništa sa izrazom, tako da možete uneti bilo koji tekst koji želite nakon duple kose crte. Otkucajte naredni izraz istovetno kako je ovde prikazan i pritisnite Enter taster na kraju reda.



### Napomena

Za više informacija o kreiranju dobrih komentara videti Čas 16, "Debagovanje koda".

```
// Pokazi okvir za dijalog open file.
```

Sledeći izraz koji ćete uneti okida metod OpenFileDialog kontrole koju ste dodali formi. Naučićete sve o metodima na Času 3, "Razumevanje objekata i kolekcija". Od sada razmišljajte o metodu kao o mehanizmu za nametanje da kontrola preduzme akciju. ShowDialog metod "govori" kontroli da prikaže njen okvir za dijalog Open i omogući korisniku da izabere fajl. ShowDialog metod vraća vrednost koja ukazuje na njegov uspeh, ili neuspeh, a koja se, zatim, upoređuje sa unapred definisanim rezultatom (DialogResult.OK). Ne brinite previše o onome šta se ovde dešava, jer ćete naučiti detalje u kasnijim časovima. U stvari, ShowDialog metod je pozvan da bi omogućio korisniku da traži fajl i da se, ukoliko korisnik izabere fajl, izvrši još koda. Naravno, OpenFileDialog kontrola postoji mnogo više u upotrebi nego što sam ja predstavio u ovom osnovnom primeru, ali ovaj jednostavni izraz završava posao. Unesite naredni if izraz praćen otvorenom vitičastom zagradom:



## Čas I.

```
if (ofdSelectPicture.ShowDialog() == DialogResult.OK)
{
```



### Napomena

Otvorene i zatvorene vitičaste zgrade su neophodne za ovaj if izraz, jer označavaju da će više redova biti deo if konstrukcije.

Vreme je za još jedan komentar. Unesite ovaj izraz i zapamtite da pritisnete Enter na kraju svakog reda koda.

```
// Ucitavanje slike u okvir slike
```



### Savet

Ne brinite o uvlačenju koda tako što ćete pritiskati Tab taster, ili koristiti razmake. C# .NET automatski uvlači kod umesto Vas.

Sada ćete uneti sledeći red koda. Ovaj izraz, koji se pojavljuje u okviru if konstrukcije, predstavlja red koda koji zapravo prikazuje sliku u okviru za sliku. (više detalja o slikama naći ćete na Času 10, "Crtanje i prikazivanje").

Unesite naredni red:

```
picShowPicture.Image = Image.FromFile(ofdSelectPicture.FileName);
```

Osim prikazivanja izabrane slike, Vaš program će prikazati putanju i ime fajla slike u naslovnoj liniji. Kada ste prvi put kreirali formu, promenili ste Text karakteristiku forme upotrebotom prozora karakteristika. Da bi bili kreirane dinamičke aplikacije, karakteristike treba konstantno podešavati za vreme izvršavanja; to je postignuto upotrebotom koda:

```
// Prikazivanje imena fajla u naslovu forme.
this.Text = String.Concat("Pregled Slika
(" + ofdSelectPicture.FileName + ")");
```



### Napomena

U C# je veličina slova važna. Morate uneti sav kod, koristeći mala i velika slova, baš kao što je pokazano u tekstu.



## Provera tačke unosa programa

Svi C# programi moraju sadržati tačku unosa. Main() metod je tačka unosa. U ovom primeru potrebno je da promenite referencu iz Form1 u **fclsViewer** (ovo je neophodno, jer je ranije promenjeno ime forme). Ovaj izraz će uključiti konstruktor forme. C++ programeri će biti "bliski" sa Main() metod tačkom unosa, ali bi trebalo da obrate pažnju na prvo veliko slovo u Main. Govorimo nešto više o tački unosa programa kasnije u knjizi.

Radi ažuriranja tačke unosa u ovom primeru, pritisnite CTRL+F da biste otvorili Find prozor, unesite **Form1** i kliknite Find Next. Zatvorite Find prozor i zamenite tekst Form1 sa **fclsViewer**. Ažurirani izraz bi sada trebalo da bude ovakav:

```
Application.Run(new fclsViewer());
```

Nakon što ste uneli sav kod, Vaš editor bi trebalo da izgleda poput onog prikazanog na slici 1.11.

**Slika 1.11**

Uverite se da Vaš kod potpuno odgovara kodu pokazanom ovde.

The screenshot shows the Microsoft Visual Studio IDE. The title bar says "Microsoft Visual Studio .NET [Design] EditProject.cs". The code editor window displays the following C# code:

```
namespace Turneja
{
    public class Program
    {
        static void Main()
        {
            Application.Run(new fclsViewer());
        }
    }
}
```

Below the code editor, the Task List shows one item: "Description: Click here to add a new task".



**Čas 1.**

## Okončavanje programa upotrebom koda

Poslednji delić koda koji ćete napisati će okončati aplikaciju kada korisnik klikne Kraj dugme. Da biste uradili ovo, potrebno je da pristupite Click manipulatoru događaja za btnQuit dugme. Na vrhu prozora koda su dve kartice. Aktuelna kartica ima tekst fclsViewer.cs. Pored nje je kartica koja sadrži tekst fclsViewer.cs [Design]. Kliknite sada nju da biste se prebacili iz pogleda koda u dizajner forme. Ukoliko primite poruku o grešci kada kliknete karticu, kod koji ste uneli nije tačan i potrebno je da ga editujete da biste ga učinili istim kao što sam ga ja prikazao. Nakon što je dizajner formi prikazan, duplo kliknite Kraj dugme da biste pristupili njegovom Click dogadaju.

Unesite naredni kod u Click manipulator događaja Kraj dugmeta:

```
this.Close();
```



### Napomena

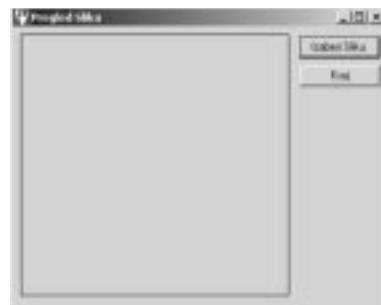
*Close izraz zatvara aktuelnu formu. Kada je poslednja učitana forma u programu zatvorena, aplikacija se kompletno zatvara. Kada budete pravili robustnije aplikacije, verovatno ćete željeti da izvršite sve vrste rutina za čišćenje pre nego što okončate Vašu aplikaciju, ali za ovaj primer zatvaranje forme je sve što treba da uradite.*

## Izvršavanje projekta

Vaša aplikacija je sada kompletan. Kliknite dugme Save All (liči na slog disketa) na paleti sa alatom, pa pokrenite Vaš program, tako što ćete pritisnuti F5. Možete pokrenuti program i tako što ćete kliknuti dugme sa palete sa alatom koje liči na desno usmereni trougao i na Play dugme na video-rekorderu (ovo dugme se može naći i u Debug meniju, a zove se Start). Međutim, učenje prečica sa tastature će učiniti Vaš proces programiranja bržim. Kada pokrenete program, C# interfejs se menja i čini se da forma koju ste dizajnirali pluta iznad dizajnerskog okruženja (videti sliku 1.12).

### Slika 1.12

*Kada je u Run modu (izvršni mod), Vaš program se izvršava na isti način kao što će se izvršavati kod krajnjeg korisnika.*



## “Turneja” kroz C# programiranje



Sada izvršavate Vaš program kao da je samostalna aplikacija koja se izvršava na računaru drugog korisnika; ono što vidite je tačno ono što će neko drugi videti ukoliko pokrene program (bez C# dizajnerskog okruženja u pozadini, naravno). Kliknite dugme Izaberite sliku da biste prikazali okvir za dijalog Izaberite sliku (videti sliku 1.13). Koristite okvir za dijalog da biste locirali fajl slike. Kada nađete fajl, duplo ga kliknite, ili ga kliknite jednom da biste ga izabrali, pa kliknite Open. Izabrana slika je, zatim, prikazana u PictureBox kontroli, kao što je pokazano na slici 1.14.

**Slika 1.13**  
*OpenFileDialog  
kontrola vodi računa  
o svim detaljima  
traženja fajlova.  
Ludo, zar ne?*



**Slika 1.14**  
*C# čini lakim  
prikazivanje slika sa  
vrlo malo rada.*



## Rezime

Kada ste završili igranjem sa programom, kliknite Kraj dugme i snimite Vaš projekat, tako što ćete kliknuti Save All sa C# palete sa alatom.

Upravo ste kreirali istinski C# program. Koristili ste okvir sa alatom da biste izradili interfejs pomoću koga korisnici mogu interagovati sa Vašim programom i napisali ste kod u strateškim manipulatorima događaja da biste svom programu dali sposobnost da radi nešto. To su osnove razvoja aplikacija u C#-u. Čak i najkomplikovaniji programi su izrađeni upotrebom ovog osnovnog pristupa; izradili ste interfejs i dodali kod da



## Čas 1.

biste postigli da aplikacija radi nešto. Naravno, pisanje koda da bi ona radila nešto tačno na način na koji želite da to bude urađeno je mesto gde se proces može iskomplikovati, ali Vi ste "na svom putu".

Ukoliko pažljivije pogledate organizaciju časova u ovoj knjizi, videćete da sam počeo od toga da naučite C# okruženje. Zatim sam prešao na izgradnju interfejsa, pa sam Vas učio svemu o pisanju koda. Ova organizacija je promišljena. Možda ćete biti malo zabrinuti zbog "uletanja" u pisanje ozbiljnog koda, ali pisanje koda je samo deo jednačine. Dok budete napredovali kroz časove, izgradićete solidne osnove razvojnih sposobnosti.

Uskoro nećete obraćati pažnju na čoveka "iza zavese" - Vi ćete biti taj čovek!

## Pitanja i odgovori

- P Da li mogu pokazati bitmapu fajlova tipa različitog od BMP i JPG?**
- O** Da. PictureBox podržava prikazivanje slika sa ekstenzijama BMP, JPG, ICO, EMF, WMF i GIF. PictureBox može, čak, i snimati slike u fajl.
- P Da li je moguće prikazati slike u drugim kontrolama?**
- O** PictureBox je kontrola za upotrebu kada samo prikazujete slike. Međutim, mnoge druge kontrole omogućavaju da prikazujete slike kao deo kontrole. Na primer, možete prikazati sliku na Button kontroli, tako što ćete postaviti Image karakteristiku dugmeta na validnu sliku.

## Radionica

Radionica je dizajnirana da bi pomogla da predvidite moguća pitanja, razmotrite šta ste naučili i da bi Vas navela da razmišljate kako da primenite Vaše znanje u praksi. Odgovori za kviz su u dodatku A, "Odgovori na kviz pitanja i vežbe".

## Kviz

1. Koji tip C# projekta kreira standardni Windows program?
2. Koji prozor se koristi za promenu atributa (lokacije, veličine i tako dalje) forme, ili kontrole?
3. Kako pristupate osnovnom dogadaju (kodu) kontrole?
4. Koju karakteristiku kontrole PictureBox postavljate da bi prikazala sliku?
5. Koji je osnovni događaj za Button kontrolu?



## **Vežba**

1. Promenite Vaš program za pregled slika da korisnik može, takođe, locirati i izabrati GIF fajlove (nagoveštaj: promenite Filter karakteristiku OpenFileDialog kontrole).
2. Promenite formu u Vašem projektu programa za pregled slika da dugmad budu jedno pored drugog u donjem desnom uglu forme, umesto da budu vertikalno poravnata u gornjem desnom uglu.

blanko