

James Foxall

Samostalno naučite

Visual Basic[®] 2015

u **24**
Lekcije

 kompjuter
biblioteka

SAMS

James Foxall

Samostalno naučite

Visual Basic[®] 2015

u **24**
Lekcije

 kompjuter
biblioteka

SAMS

Izdavač:



**kompjuter
biblioteka**

Obalskih radnika 15, Beograd

Tel: 011/2520272

e-mail: kombib@gmail.com

internet: www.kombib.rs

Urednik: Mihailo J. Šolajić

Za izdavača, direktor:

Mihailo J. Šolajić

Autor: James Foxall

Prevod: Biljana Tešić i
Slobodan Šećerovski

Lektura: Zlatan Pešić

Slog : Zvonko Aleksić

Znak Kompjuter biblioteke:
Miloš Milosavljević

Štampa: „Pekograf“, Zemun

Tiraž: 500

Godina izdanja: 2017.

Broj knjige: 488

Izdanje: Prvo izdanje

ISBN: 978-86-7310-511-6

**Visual Basic® 2015 in 24 Hours,
Sams Teach Yourself**

Copyright © 2016 by Pearson Education, Inc.

James Foxall

ISBN-13: 978-0-672-33745-1

All right reserved. No part of this book may be reproduced or transmitted in any form or by means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission from the Publisher.

Autorizovani prevod sa engleskog jezika edicije u izdanju „Pearson Education, Inc“, Copyright © 2016.

Sva prava zadržana. Nije dozvoljeno da nijedan deo ove knjige bude reprodukovano ili snimljeno na bilo koji način ili bilo kojim sredstvom, elektronskim ili mehaničkim, uključujući fotokopiranje, snimanje ili drugi sistem presnimavanja informacija, bez dozvole izdavača.

Zaštitni znaci

Kompjuter Biblioteka i „Pearson Education, Inc“ su pokušali da u ovoj knjizi razgraniče sve zaštitne oznake od opisnih termina, prateći stil isticanja oznaka velikim slovima.

Autor i izdavač su učinili velike napore u pripremi ove knjige, čiji je sadržaj zasnovan na poslednjem (dostupnom) izdanju softvera. Delovi rukopisa su možda zasnovani na predizdanju softvera dobijenog od strane proizvođača. Autor i izdavač ne daju nikakve garancije u pogledu kompletnosti ili tačnosti navoda iz ove knjige, niti prihvataju ikakvu odgovornost za performanse ili gubitke, odnosno oštećenja nastala kao direktna ili indirektna posledica korišćenja informacija iz ove knjige.

CIP - Каталогизација у публикацији
Народна библиотека Србије, Београд,
се добија на захтев



Kratak sadržaj

POGLAVLJE 1

Bez zadržke: Programiranje u Visual Basicu 2015..... 1

POGLAVLJE 2

Snalaženje u Visual Basicu 201531

POGLAVLJE 3

Objekti i kolekcije63

POGLAVLJE 4

Događaji87

POGLAVLJE 5

Osnove izrade formulara 107

POGLAVLJE 6

Napredne tehnike izrade formulara..... 131

POGLAVLJE 7

Korišćenje tradicionalnih kontrola163

POGLAVLJE 8

Korišćenje naprednih kontrola193

POGLAVLJE 9

Postavljanje menija i traka sa alatkama na obrascima215

POGLAVLJE 10

Programiranje i pozivanje procedura239

POGLAVLJE 11**Korišćenje konstanti, tipova podataka, promenljivih i nizova.....259****POGLAVLJE 12****Obavljanje aritmetičkih operacija, rad sa stringovima
i podešavanje datuma i vremen291****POGLAVLJE 13****Donošenje odluka korišćenjem Visual Basic koda.....313****POGLAVLJE 14****Korišćenje ciklusa radi postizanja veće efikasnosti329****POGLAVLJE 15****Otklanjanje grešaka iz programskog koda343****POGLAVLJE 16****Dizajniranje objekata pomoću klasa.....371****POGLAVLJE 17****Interakcija sa korisnicima391****POGLAVLJE 18****Računarska grafika413****POGLAVLJE 19****Upravljanje datotekama435****POGLAVLJE 20****Rad sa bazom Registry i tekstualnim datotekama.....457****POGLAVLJE 21****Rad sa bazama podataka483****POGLAVLJE 22****Štampanje505****POGLAVLJE 23****Slanje elektronske pošte529****POGLAVLJE 24****Instaliranje aplikacija545****DODATAK A****Pogled sa distance559****INDEKS567**

Sadržaj

POGLAVLJE 1

Bez zadržke: Programiranje u Visual Basicu 2015..... 1

Pokretanje Visual Basicu 2015	2
Započinjanje novog projekta.....	3
Upoznavanje korisničkog okruženja Visual Basicu 2015.....	6
Menjanje karakteristika objekata.....	7
Imenovanje objekata.....	8
Zadavanje svojstva Text formulara.....	10
Zapisivanje projekta na hard disk	10
Zadavanje ikone za formular	11
Menjanje veličine formulara	11
Postavljanje kontrola na formular	13
Dizajniranje korisničkog interfejsa	15
Postavljanje vidljivih kontrola na formular.....	15
Postavljanje nevidljivih kontrola na formular.....	18
Pisanje programskog koda za korisnički interfejs.....	20
Omogućite korisniku da pronađe datoteku.....	20
Prekidanje izvršavanja programa pomoću programskog koda.....	23
Pokretanje i izvršavanje projekta	24
Rezime	27
Pitanja i odgovori	28
Radionica	28
Test.....	28
Odgovori.....	28
Vežbanja	29

POGLAVLJE 2**Snalaženje u Visual Basicu 201531**

Početna stranica u Visual Basicu 2015	32
Pravljenje novih projekata.....	32
Otvaranje postojećeg projekta	34
Snalaženje u korisničkom okruženju Visual Basica i njegovo podešavanje	34
Rad sa prozorima za dizajniranje objekata	35
Prikazivanje i sakrivanje prozora za dizajniranje objekata	36
Pokretni prozori za dizajniranje objekata	36
Fiksiranje prozora za dizajniranje objekata.....	37
Automatsko skrivanje prozora za dizajniranje.....	39
Korišćenje paleta alatki (Toolbar)	40
Prikazivanje i skrivanje paleta alatki.....	40
Postavljanje kontrola na formular pomoću paleta alatki (Toolbox).....	41
Zadavanje svojstava objekata korišćenjem prozora Properties.....	43
Selektovanje objekta i prikazivanje njegovih svojstava	44
Prikazivanje svojstava i promena vrednosti svojstava	44
Svojstva pomoću kojih se zadaje boja	46
Prikazivanje opisa svojstava	49
Upravljanje projektima	50
Upravljanje datotekama koje sačinjavaju projekt korišćenjem prozora Solution Explorer.....	50
Upravljanje rešenjima.....	52
Komponente projekta	53
Zadavanje vrednosti svojstvima projekta	54
Dodavanje i uklanjanja datoteka projekta.....	55
Kratko uputstvo za pisanje programskog koda.....	57
Dodeljivanje vrednosti promenljivama	58
Upotreba procedura radi pisanja funkcionalnih jedinica programskog koda	58
Dobijanje pomoći.....	59
Rezime	60
Pitanja i odgovori	60
Radionica	61
Test.....	61
Odgovori	61
Vežbanja	61

POGLAVLJE 3**Objekti i kolekcije63**

Objekti	64
Svojstva objekata	64
Čitanje i zadavanje vrednosti svojstvima	65
Objekti i njihova svojstva	67
Metode objekata.....	72
Pozivanje metoda	73
Dinamizam metoda.....	73
Izrada jednostavnog objekta.....	73
Kreiranje interfejsa za projekat	74

Objektni programski kod	74
Testiranje primera projekta u kojem se koriste objekti	78
Kolekcije	79
Upotrebljavanje Object Browsersa	82
Rezime	84
Pitanja i odgovori	84
Radionica	85
Test	85
Odgovori	85
Vežbanja	85

POGLAVLJE 4

Događaji 87

Izrada programa vođenih događajima	87
Iniciranje događaja	88
Događaji koje inicira korisnik	88
Događaji koje inicira objekat	89
Događaji koje inicira operativni sistem	90
Izbegavanje rekurzivnih događaja	90
Pristupanje događajima objekta	91
Parametri događaja	95
Izrada primera vođenog događajima	97
Pravljenje korisničkog interfejsa	97
Pisanje programskog koda za događaje	98
Imena događaja	103
Rezime	103
Pitanja i odgovori	104
Radionica	104
Test	104
Odgovori	104
Vežbanja	105

POGLAVLJE 5

Osnove izrade formulara 107

Menjanje imena formulara	108
Promena izgleda formulara	109
Prikazivanje teksta u naslovnoj liniji formulara	110
Menjanje boje pozadine formulara	111
Postavljanje slike kao pozadine formulara	113
Pridruživanje ikone formularu	116
Pridruživanje kontrolnih dugmadi Minimize, Maximize i Control Box	117
Promena izgleda i ponašanja bordure formulara	119
Zadavanje minimalne i maksimalne veličine formulara	121
Prikazivanje i skrivanje formulara	122
Prikazivanje formulara	122
Modaliteti formulara	123
Prikazivanje formulara u stanju Normal, Maximized i Minimized	124

Zadavanje inicijalne pozicije formulara	126
Sprečavanje prikazivanja formulara na paleti aktivnih poslova	127
Izbacivanje formulara iz memorije	127
Rezime	128
Pitanja i odgovori	128
Radionica	129
Test	129
Odgovori	129
Vežbanja	130

POGLAVLJE 6

Napredne tehnike izrade formulara..... 131

Upotrebljavanje kontrola.....	131
Postavljanje kontrola na formular.....	132
Postavljanje kontrola pomoću dva klika u prozoru Toolbox.....	132
Postavljanje kontrola prevlačenjem na formular	132
Postavljanje kontrola iscrtavanjem na formularu.....	132
Rukovanje kontrolama.....	133
Korišćenje mreže (veličina i priljubljivanje)	133
Korišćenje linija za priljubljivanje kontrola	136
Označavanje grupe kontrola.....	138
Poravnavanje kontrola	140
Zadavanje iste veličine za grupu kontrola	142
Zadavanje jednakog razmaka za grupu kontrola	142
Zadavanje vrednosti svojstava za grupu označenih kontrola	142
Automatsko fiksiranje i automatska promena veličine kontrola	143
Zadavanje „redosleda“ kontrola	149
Slaganje kontrola	151
Izrada nemodalnih prozora.....	151
Izrada transparentnih formulara	151
Izrada formulara sa klizačima (scrollbar)	152
Izrada MDI formulara.....	154
Zadavanje formulara koji se prikazuje prilikom pokretanja programa.....	158
Rezime	159
Pitanja i odgovori	160
Radionica	160
Test	160
Odgovori	160
Vežbanja	161

POGLAVLJE 7

Korišćenje tradicionalnih kontrola 163

Prikazivanje teksta pomoću kontrole Label.....	163
Omogućite korisnicima da tekst unesu pomoću kontrole Text Box.....	164
Poravnavanje teksta.....	166
Tekstualno polje u kojem se sadržaj prikazuje u više redova	167
Prikazivanje klizača.....	169

Ograničavanje broja znakova u tekstualnom polju	170
Polja za unošenje lozinki	171
Događaji tekstualnog polja koji se obično koriste	172
Izrada komandnih dugmadi	172
Komandna dugmad Accept i Cancel	173
Prikazivanje opcija da/ne pomoću polja za potvrdu	175
Izrada kontejnera i grupisanih polja	176
Panoi i grupisana polja	176
Kontrola tipa Radio Button	178
Prikazivanje spiska pomoću kontrole tipa List Box	180
Upravljanje elementima tokom dizajniranja	181
Upravljanje elementima tokom izvršavanja programa	182
Dodavanje elemenata u spisak	182
Uklanjanje elemenata iz spiska	183
Uklanjanje svih elemenata iz spiska	185
Dobijanje informacija o elementu koji je odabran u spisku	185
Sortiranje spiska elemenata	187
Izrada spiskova pomoću kontrole Combo Box	188
Rezime	190
Pitanja i odgovori	191
Radionica	191
Test	191
Odgovori	192
Vežbanja	192

POGLAVLJE 8

Korišćenje naprednih kontrola 193

Kreiranje tajmera	193
Kreiranje okvira za dijalog sa karticama	197
Skladištenje slika u kontroli Image List	200
Pravljenje poboljšanih lista pomoću kontrole List View	202
Kreiranje kolona	203
Dodavanje stavki liste	204
Upravljanje kontrolom List View pomoću koda	206
Dodavanje stavki liste pomoću koda	206
Određivanje izabrane stavke u kodu	206
Uklanjanje stavki liste pomoću koda	207
Uklanjanje svih stavki liste	207
Kreiranje hijerarhijskih lista pomoću kontrole Tree View	207
Dodavanje čvorova kontroli Tree View	208
Uklanjanje čvorova	211
Brisanje svih čvorova	211
Rezime	211
Pitanja i odgovori	212
Radionica	212
Test	212
Odgovori	213
Vežbanja	213

POGLAVLJE 9**Postavljanje menija i traka sa alatkama na obrascima215**

Izrada menija	215
Izrada menija najvišeg nivoa	216
Kreiranje stavki menija za meni najvišeg nivoa	219
Kreiranje i brisanje stavki menija	220
Kreiranje potvrđenih stavki menija	220
Pogramiranje menija	223
Implementiranje kontekstnog menija	225
Dodeljivanje tastatskih prečica stavkama menija	227
Korišćenje kontrole Toolbar	229
Programiranje traka sa alatkama	233
Kreiranje padajućih menija za dugmad na traci sa alatkama	234
Kreiranje statusne linije (Status Bar)	235
Rezime	237
Pitanja i odgovori	237
Radionica	238
Kviz	238
Odgovori	238
Vežbanja	238

POGLAVLJE 10**Programiranje i pozivanje procedura239**

Modul sa programskim kodom u Visual Basicu	239
Procedure	242
Deklasiranje procedura koje ne daju vrednost	243
Deklarisanje procedura koje daju vrednosti	247
Pozivanje procedura	248
Prosleđivanje parametara	252
Izlaženje iz procedura	254
Izbegavanje beskonačne rekurzije	255
Rezime	256
Pitanja i odgovori	257
Radionica	257
Kviz	257
Odgovori	257
Vežbanja	258

POGLAVLJE 11**Korišćenje konstanti, tipova podataka, promenljivih i nizova259**

Tipovi podataka	260
Određivanje tipova podataka	260
Konverzija iz jednog tipa podataka u drugi	262
Definisanje i korišćenje konstanti	263
Deklarisanje i referenciranje promenljivih	266
Deklasiranje promenljivih	266
Prosleđivanje literalnih vrednosti promenljivama	268

Korišćenje promenljivih u izrazima	268
Prisilno deklarisanje i zadavanje tipa podataka	269
EksPLICITNO deklarisanje promenljive	269
Precizno zadavanje.....	270
Rad sa nizovima.....	273
Dimenzionisanje nizova	273
Referenciranje promenljive niza.....	273
Kreiranje višedimenzionalnih nizova	274
Određivanje opsega primene	276
Upoznavanje opsega primene bloka.....	277
Upoznavanje (lokalnog) opsega primene na nivou procedure	278
Upoznavanje opsega primene na nivou modula.....	278
Korišćenje globalnog opsega primene (opsega primene imenskog prostora)	279
Neusaglašenosti naziva opsega primene.....	280
Deklarisanje promenljivih statičkog opsega primene	281
Korišćenje promenljivih u programu za prikazivanje slika	282
Kreiranje promenljivih za opcije.....	282
Menjanje imena promenljivih.....	286
Rezime	287
Pitanja i odgovori	288
Radionica	288
Kviz.....	288
Odgovori	289
Vežbanja	289

POGLAVLJE 12

Obavljanje aritmetičkih operacija, rad sa stringovima

i podešavanje datuma i vremen291

Obavljanje osnovnih aritmetičkih operacija u Visual Basicu	292
Sabiranje	292
Oduzimanje i negacija.....	292
Množenje	293
Deljenje.....	293
Stepenovanje	293
Modularna aritmetika.....	293
Određivanje redosleda prioriteta operatora.....	294
Poređenje po jednakosti.....	295
Upoznavanje Bulove logike	296
Korišćenje operatora And	297
Korišćenje operatora Not	297
Korišćenje operatora Or.....	298
Korišćenje operatora Xor.....	298
Rad sa stringovima.....	298
Nadovezivanje stringa teksta	299
Korišćenje osnovnih funkcija stringa.....	300
Određivanje broja znakova pomoću funkcije Len().....	300
Preuzimanje teksta sa leve strane stringa pomoću funkcije Microsoft.VisualBasic.Left()	300

Preuzimanje teksta sa desne strane stringa pomoću funkcije Microsoft.VisualBasic.Right()	301
Preuzimanje teksta iz stringa pomoću funkcije Microsoft.VisualBasic.Mid()	301
Utvrdite da li jedan string sadrži drugi pomoću funkcije Instr().....	302
Uklanjanje početnih i pratećih razmaka u stringu	303
Zamena teksta u stringu	304
Podešavanje vremena i datuma.....	304
Upoznavanje tipa podataka Date	304
Dodavanje ili oduzimanje datuma i vremena	305
Određivanje intervala između dva datuma i vremena.....	307
Preuzimanje delova datuma	307
Formatiranje datuma i vremena.....	308
Preuzimanje trenutnog sistemskog datuma i vremena	309
Utvrdite da li je vrednost datum	309
Rezime	309
Pitanja i odgovori	310
Radionica	310
Kviz	310
Odgovori	311
Vežbanja	311

POGLAVLJE 13

Donošenje odluka korišćenjem Visual Basic koda.....313

Postavljanje uslova pomoću komande If...Then.....	313
Izvršavanje koda kada expression ima vrednost False	316
Ugnežđavanje konstrukcije If...Then	318
Izračunavanje više vrednosti izraza pomoću iskaza Select Case.....	319
Izračunavanje više mogućih vrednosti u iskazu Case.....	320
Pravljenje primera Select Case	321
Kreativna upotreba iskaza Select Case.....	323
Prelazak na programski red unutar procedure pomoću komande GoTo	324
Rezime	326
Pitanja i odgovori	327
Radionica	327
Kviz	327
Odgovori	328
Vežbanja	328

POGLAVLJE 14

Korišćenje ciklusa radi postizanja veće efikasnosti329

Ponavljanje dela programskog koda korišćenjem petlje For...Next.....	329
Pokretanje petlje pomoću iskaza For	330
Zatvaranje petlje pomoću iskaza Next.....	330
Ranije izlaženje iz petlje.....	332
Ponavljanje dela programskog koda pre iskaza Next.....	332
Kreiranje primera petlje For...Next.....	332
Neograničeno ponavljanje dela programskog koda korišćenjem petlje Do...Loop.....	336

Pravljenje petlje Do...Loop.....	336
Završetak petlje Do...Loop	336
Pravljenje primera petlje Do...Loop.....	338
Rezime	341
Pitanja i odgovori	341
Radionica	342
Kviz	342
Odgovori	342
Vežbe.....	342

POGLAVLJE 15

Otklanjanje grešaka iz programskog koda 343

Pisanje komentara u okviru koda	344
Identifikovanje dve osnovne vrste grešaka.....	346
Korišćenje Visual Basic alatki za otklanjanje grešaka.....	349
Korišćenje tačaka prekida	349
Korišćenje prozora Immediate	351
Korišćenje naprednih funkcija tačke prekida.....	357
Zaustavljanje samo kada je uslov zadovoljen	358
Zaustavljanje samo kada se tačka prekida dostigne zadati broj puta.....	359
Prikazivanje poruka u prozoru Output kada se dostigne zadato mesto u programskom kodu	360
Obrada grešaka korišćenjem programskog bloka Try...Catch...Finally.....	360
Obrada greške.....	363
Obrada predviđene greške	364
Rezime	368
Pitanja i odgovori	368
Radionica	368
Kviz	368
Odgovori	369
Vežbanja	369

POGLAVLJE 16

Dizajniranje objekata pomoću klasa..... 371

Klase	372
Enkapsuliranje podataka i koda pomoću klasa.....	372
Upoređivanje klasa sa standardnim modulima	373
Kreiranje interfejsa objekta	374
Izlaganje atributa objekta kao svojstva.....	376
Kreiranje svojstava koja se mogu samo čitati korišćenjem konstrukcije Get	378
Kreiranje svojstava koja se mogu samo čitati korišćenjem konstrukcije Set.....	379
Kreiranje svojstava read-only i write-only	380
Izlaganje funkcija kao metoda.....	381
Instanciranje objekata pomoću klasa.....	381
Povezivanje reference objekta sa promenljivom	382
Kasno povezivanje promenljive objekta.....	382
Rano povezivanje promenljive objekta.....	384

Kreiranje novog objekta prilikom dimenzionisanja promenljive	386
Oslobađanje referenci objekata	387
Trajanje objekta	387
Rezime	388
Pitanja i odgovori	388
Radionica	388
Kviz	388
Odgovori	389
Vežbanja	389

POGLAVLJE 17

Interakcija sa korisnicima391

Prikazivanje poruka pomoću funkcije MessageBox.Show().....	391
Određivanje dugmadi i ikone	393
Određivanje dugmeta koje će se kliknuti.....	396
Kreiranje dobrih poruka.....	397
Kreiranje sopstvenih okvira za dijalog	398
Dobijanje informacija od korisnika pomoću funkcije InputBox()	401
Rad sa tastaturom.....	404
Korišćenje standardnih događaja miša	406
Rezime	409
Pitanja i odgovori	410
Radionica	410
Kviz	410
Odgovori	411
Vežbanja	411

POGLAVLJE 18

Računarska grafika413

Objekat Graphics	413
Kreiranje objekta Graphics za obrazac ili kontrolu	414
Kreiranje objekta Graphics za novu bitmapu	415
Korišćenje objekta Pen	416
Korišćenje sistemskih boja	417
Korišćenje objekta Rectangle.....	421
Crtanje raznih oblika.....	422
Crtanje linija	422
Crtanje pravougaonika	423
Crtanje krugova i elipsa	423
Brisanje površine za crtanje.....	423
Isctavanje teksta	423
Prikazivanje grafike na obrascu	425
Izrada primera u kojima se koristi grafika	425
Rezime	432
Pitanja i odgovori	432
Radionica	432

Kviz	432
Odgovori	433
Vežbanja	433

POGLAVLJE 19

Upravljanje datotekama435

Korišćenje kontrola OpenFileDialog i SaveFileDialog	435
Korišćenje kontrole OpenFileDialog	436
Kreiranje filtera datoteke	439
Prikazivanje okvira za dijalog Open File	440
Korišćenje kontrole SaveFileDialog	441
Upravljanje datotekama pomoću objekta File	443
Utvrđivanje da li datoteka postoji	443
Kopiranje datoteke	444
Premeštanje datoteke	445
Menjanje naziva datoteke	446
Brisanje datoteke	446
Preuzimanje svojstava datoteke	447
Dobijanje informacija o datumu i vremenu datoteke	448
Dobijanje atributa datoteke	449
Pisanje koda za preuzimanje svojstava datoteke	450
Upravljanje direktorijumima pomoću objekta Directory	452
Rezime	453
Pitanja i odgovori	454
Radionica	454
Kviz	454
Odgovori	454
Vežbanja	455

POGLAVLJE 20

Rad sa bazom Registry i tekstualnim datotekama457

Rad sa bazom Registry	457
Struktura baze Windows Registry	458
Pristupanje bazi Registry pomoću objekta My.Computer.Registry	460
Kreiranje ključeva baze Registry	460
Brisanje ključeva baze Registry	461
Dobijanje i podešavanje vrednosti ključa	462
Modifikovanje projekta Picture Viewer radi korišćenja baze Registry	464
Prikazivanje opcija iz baze Registry	464
Skladištenje opcija u bazu Registry	465
Korišćenje opcija koje su uskladištene u bazu Registry	465
Testiranje i otklanjanje grešaka projekta Picture Viewer	467
Čitanje i zapisivanje tekstualnih datoteka	470
Zapisivanje tekstualne datoteke	470
Čitanje tekstualne datoteke	472
Modifikovanje projekta Picture Viewer radi korišćenja tekstualne datoteke	474
Kreiranje datoteke evidencije programa Picture Viewer	474

Prikazivanje datoteke evidencije programa Picture Viewer	477
Testiranje evidencije programa Picture Viewer	479
Rezime	480
Pitanja i odgovori	481
Radionica	481
Kviz	481
Odgovori	482
Vežbanja	482

POGLAVLJE 21

Rad sa bazama podataka483

Uvod u ADO.NET	484
Povezivanje sa bazom podataka	485
Zatvaranje veze sa izvorom podataka	491
Upravljanje podacima.....	491
Objekti DataTable	491
Kreiranje objekta DataAdapter	492
Kreiranje i popunjavanje objekata DataTable	493
Referenciranje polja u objektu DataRow.....	494
Kretanje kroz zapise u bazi podataka	496
Uređivanje zapisa podataka.....	498
Kreiranje novog zapisa	499
Brisanje zapisa.....	500
Pokretanje primera baze podataka	502
Rezime	502
Pitanja i odgovori	502
Radionica	503
Kviz	503
Odgovori	503
Vežbanja	503

POGLAVLJE 22

Štampanje505

Pripremanje projekta Picture Viewer	506
Dodavanje dugmadi Print i Print Preview.....	506
Dodavanje kontrola PrintDocument, PrintPreviewDialog i PageSetupDialog.....	508
Pregled za štampu i štampanje dokumenta.....	509
Štampanje dokumenta	510
Kreiranje procedure PrintImage	512
Štampanje trenutno prikazane slike.....	514
Pregled dokumenta	517
Menjanje podešavanja stranice i štampača	519
Menjanje razmere slike da bi odgovarala veličini stranice	522
Rezime	527
Pitanja i odgovori	528

Radionica	528
Kviz	528
Odgovori	528
Vežbanja	528

POGLAVLJE 23

Slanje elektronske pošte529

Klase koje se koriste za slanje elektronske pošte	530
Slanje elektronske pošte iz aplikacije Picture Viewer	530
Dodavanje dugmeta Send Email na traci sa alatkama	530
Kreiranje okvira za dijalog Send Email	532
Pisanje koda za slanje elektronske pošte	537
Testiranje koda za elektronsku poštu	541
Rezime	543
Pitanja i odgovori	544
Radionica	544
Kviz	544
Odgovori	544
Vežbanja	544

POGLAVLJE 24

Instaliranje aplikacija545

Tehnologija ClickOnce	545
Korišćenje čarobnjaka Publish Wizard za pravljenje ClickOnce programa.....	547
Testiranje ClickOnce instalacionog programa za aplikaciju Picture Viewer.....	552
Deinstaliranje aplikacije koju ste distribuirali.....	553
Podешavanje naprednih opcija za pravljenje ClickOnce programa.....	556
Rezime	557
Pitanja i odgovori	557
Radionica	557
Kviz	557
Odgovori	558
Vežbanja	558

POGLAVLJE A

Pogled sa distance559

Okruženje .NET	560
Komponenta Common Language Runtime	560
Microsoftov među-jezik.....	561
Imenski prostori	562
Korišćenje standarada Common Type System	563
Automatsko upravljanje podacima (garbage collection)	564
Pročitajte još.....	564
Rezime	565

INDEKS567



Uvod

Visual Basic 2015 je Majkrosoftova najnovija inkarnacija izuzetno popularnog programskog jezika Visual Basic i značajno se razlikuje od prethodnih verzija. Visual Basic je moćniji nego ikada ranije, a njegove mogućnosti i funkcionalnost se mogu porediti sa programskim jezicima „višeg nivoa“ kao što je C++. Jedna od posledica novootkrivene moći Visual Basica je njegova povećana složenost. Davno su prošli dani kada ste mogli da pokrenete Visual Basic i čitate ugrađene primere i objašnjenja i tako samostalno naučite da pravite funkcionalne programe.

Kome je knjiga namenjena i kako je strukturirana

Ova knjiga je namenjena svima koji nemaju nikakvo, ili gotovo nikakvo, iskustvo u programiranju, kao i ljudima kojima će Visual Basic biti „drugi“ programski jezik. Knjiga je strukturirana i napisana sa jednim ciljem: da što pre budete osposobljeni da programirate. Iskristio sam svoje veliko iskustvo u pravljenju komercijalnih aplikacija pomoću Visual Basica, te se nadam da će vam knjiga koju sam napisao pomoći da naučite sve što vam je potrebno. Autori, veoma često, prave grešku i fokusiraju se na tehnologiju umesto da čitaoc nauče praktičnim veštinama koje odmah mogu da primene u njihovim projektima.

Knjigu sam podelio u pet delova, a svaki deo se odnosi na po jedan aspekt pravljenja aplikacija pomoću Visual Basica. Delovi su poredani tako da odgovaraju koracima koje treba preduzeti prilikom pravljenja programa pomoću Visual Basica. Savetujem vam da ih pročitate onako kako sam ih poredao:

Deo I, „Korisničko okruženje u Visual Basicu 2015“, se odnosi na snalaženje u okruženju Visual Basica, kao i na načine kako da upotrebite brojne alatke. Osim toga, u ovom delu knjige ćete naučiti neke od bitnih koncepata programiranja kao što su objekti, kolekcije i događaji.

Deo II, „Pravljenje korisničkog interfejsa“, se odnosi na pravljenje privlačnih i funkcionalnih korisničkih interfejsa. U ovom delu knjige ćete se upoznati sa formularima i kontrolama – elementima korisničkog interfejsa, kao što su tekstualna polja i polja sa spiskovima.

Deo III, „Pokrenimo sled događaja – programiranje“, se odnosi na gradivne elemete koji se koriste u programiranju pomoću Visual Basica – a takvih je zaista mnogo. Saznaćete kako da napravite module i procedure, kako se zapisuju podaci, prave programski ciklusi i kako se donose odluke pomoću programskog koda. Nakon što ovladate osnovnim programerskim veštinama, preći ću na objektno-orijentisano programiranje i objasniću kako se otklanjaju greške u aplikacijama.

Deo IV, „Rad sa podacima“, se odnosi na rad sa grafikom, tekstualnim datotekama i na programiranje baza podataka, kao i na načine na koje možete automatizovati aplikacije kao što su Word ili Excel. Osim toga, u ovom delu knjige ćete naučiti kako da upravljate sistemom datoteka i bazom podataka Registry.

Do V, „Instaliranje gotovih programa i još ponešto“, se odnosi na ugrađivanje u aplikaciju mogućnosti slanja elektronske pošte i na distribuiranje aplikacije krajnjim korisnicima. U Dodatku A, „Pogled sa distance“, ćete se upoznati sa Majkrosoftovim .NET okruženjem iz manje tehničkog ugla.

Mnogi koji su pročitali prethodna izdanja su mi uputili sugestije kako da ovu knjigu učinim boljom. Najveći broj njih je tražio da u knjigu uvrstim primere tako da se u svakom poglavlju nadograđuju primeri iz prethodnih poglavlja. U ovoj knjizi sam pokušao da, što je moguće više, udovoljim tim zahtevima. Umesto da učite o konceptima kroz izolovane primere, napravićete program Picture Viewer. Prvo ćete napraviti osnovni deo aplikacije. U program ćete, kako budete čitali poglavlja, ugrađivati menije i palete alatki, okvir za dijalog za podešavanje opcija, izmeniti program tako da koristi Windowsovu bazu Registry i tekstualnu datoteku, pa ćete, čak, napraviti program pomoću kojeg se aplikacija Picture Viewer instalira na računare korisnika. Nadam se da će vam ovakav pristup, pristup u kontekstu pravljenja upotrebljivog programa, omogućiti da bolje savladate programiranje.

Konvencije koje se koriste u knjizi

U knjizi se koristi nekoliko grafičkih elemenata i konvencija koje treba da vam pomognu da lakše uočite i klasifikujete informacije:

Izdvojeni pasusi „**Uzged budi rečeno**“ sadrže korisne informacije koje odmah možete pročitati ili se vrtatiti na njih, a da pritom ne izgubite nit izlaganja u knjizi.

Izdvojeni pasusi „**Da li ste znali?**“ sadrže važne informacije koje će vaše programiranje pomoću Visual Basica učiniti efikasnijim.

Izdvojeni pasusi „**Obrati pažnju!**“ usmeravaju pažnju na probleme ili neželjene efekte koji se mogu desiti u nekim situacijama.

Nov termin je prikazan kurzivnim tekstom da bi bio bolje uočljiv.

Osim toga, u knjizi se koriste različiti oblici slova da biste bolje razlikovali programski kod od pisanog teksta. Programski kod je prikazan posebnom vrstom slova, monospaced. Reči ili znakovi koji predstavljaju reči i znakovi koje ćete vi uneti u programski kod – su prikazani kurzivom posebnog oblika slova. Kada se traži da unesete tekst, takav tekst je prikazan masnim slovima.

Opcije menija su razdvojene zarezima. Na primer, ako treba da otvorite meni File i odaberete opciju New Project, u tekstu je to prikazano kao „File, New Project“.

Neki programski redovi su previše dugački pa ne mogu biti prikazani u jednom redu knjige. Takvi programski redovi su u knjizi prikazani korišćenjem znaka za nastavak u sledećem redu (_), što znači da se u sledećem redu nalazi nastavak programsko reda.

Od sad pa nadalje!

Uzbudljivo je učiti programiranje. Iskreno se nadam da ćete, kada budete pročitali knjigu, biti sposobni da, koristeći mnoge alatke Visual Basica, pravite, otklanjate greške i distribuirate pristojne Visual Basic programe. Iako nećete postati ekspert, bićete iznenađeni koliko toga ste naučili čitajući knjigu. Takođe, nadam se da će vam ova knjiga pomoći da utvrdite kojim putem da krenete i postanete ekspert za Visual Basic.

Obožavam da programiram u Visual Basicu, a ponekad ne mogu da verujem da me plaćaju da to radim. Nadam se da će se i vama Visual Basic dopadati koliko se meni dopada!

LEKCIJA

1

Bez zadržke: Programiranje u Visual Basicu 2015

U ovoj lekciji ćete naučiti:

- ▶ Kako da napravite jednostavnu (ali ipak funkcionalnu) Visual Basic aplikaciju
- ▶ Kako da korisniku omogućite da pogleda sadržaj hard diska
- ▶ Kako da prikazete sliku koja je zapisana u datoteci na hard disku
- ▶ Upoznaćete neke termine koji se koriste u programiranju
- ▶ Upoznaćete se sa razvojnim okruženjem Visual Basic 2015

Učenje novog programskog jezika može biti zastrašujuće iskustvo. Ukoliko nikada ranije niste programirali, unošenje naizgled šifrovanog teksta radi pravljenja lepe za oko i funkcionalne aplikacije verovatno vam liči na crnu magiju, i verovatno se pitate kako ćete naučiti sve što je neophodno da znate. Odgovor na vaše strepnje je, naravno, korak po korak. Verujem da je prvi korak u ovladavanju programskog jezika sticanje samopuzdanja. Programiranje je delimično umetnost, a delimično nauka. Iako vam programiranje izgleda kao da je magija, sličnije je stvaranju iluzije. Nakon što saznate kako se stvari odvijaju, programiranje nije više mistična veština, pa se možete usredsrediti na stvari koje su neophodne za dobijanje željenog rezultata.

Velika, komercijalna rešenja se prave tako što se prvo napravi niz manjih delova. Kada budete pročitali ovu lekciju imaćete uvid u celokupan postupak pravljenja aplikacije i napravite prvi korak ka tome da postanete uspešan programer. U narednim lekcijama ćete nadograđivati primere iz ove lekcije. Kako budete čitali knjigu, pravićete robusnu aplikaciju, čiji prozor će moći da menja veličinu i koja će sadržati intuitivni interfejs u koji su ugrađeni meniji i palete alatki, pomoću koje možete upravljati Windowsovom bazom Registry, a sve to pomoću robusnog programskog koda koji će sadržati obradu grešaka. Međutim, da ne brzam previše.

U ovoj lekciji ćete se upoznati sa Visual Basicom i napravite kompletan, ali jednostavan, Visual Basic program. U većini knjiga za početnike čitalac prvo pravi program koji ispisuje rečenicu „Zdravo svete!“. Još uvek nisam video takav program, a da ima ikakvu edukativnu ulogu.

(Sve što samo ispisuje „Zdravo svete“ i niša više – gde je tu nekakva zabava!) Vi ćete, umesto toga, napraviti aplikaciju Picture Viewer pomoću koje se na računaru prikazuju slike. Tehnike koje ćete u ovoj lekciji naučiti primenićete u aplikacijama koje ćete praviti. Ipak, cilj ove lekcije je da uvidite koliko zadovoljstva može da pruži programiranje pomoću Visual Basica.

Pokretanje Visual Basica 2015

Pre nego što počnete da programirate u Visual Basicu 2015, treba da razumete sledeće termine:

- ▶ **Distributivna komponenta:** ovo je konačna, kompajlirana verzija projekta. Komponente mogu biti distribuirane drugim korisnicima i na druge računare, i za njih nije potrebno postojanje razvojnog okruženja Visual Basic 2015 (alati koje koristite za pravljenje .NET programa) za njihovo izvršavanje (mada je neophodno da postoji .NET izvršna biblioteka, što je detaljno objašnjeno u Lekciji 24, „Instaliranje aplikacija“). Distributivne komponente se, obično, nazivaju programi. U Lekciji 24 ćete naučiti kako se program Picture Viewer, koji ćete napraviti, distribuirati na druge računare.
- ▶ **Projekat:** kolekcija datoteka koje mogu biti kompajlirane da bi mogla da bude izražena distributivna komponenta (program). Postoje mnogi tipovi projekata, a složene aplikacije se mogu sastojati od više projekata, kao što su projekti Windows aplikacija i projekti biblioteka za dinamičko povezivanje (DLL).
- ▶ **Rešenje:** kolekcija projekata i datoteka koje sačinjavaju aplikaciju ili komponentu.

UZGRED BUDI REČENO

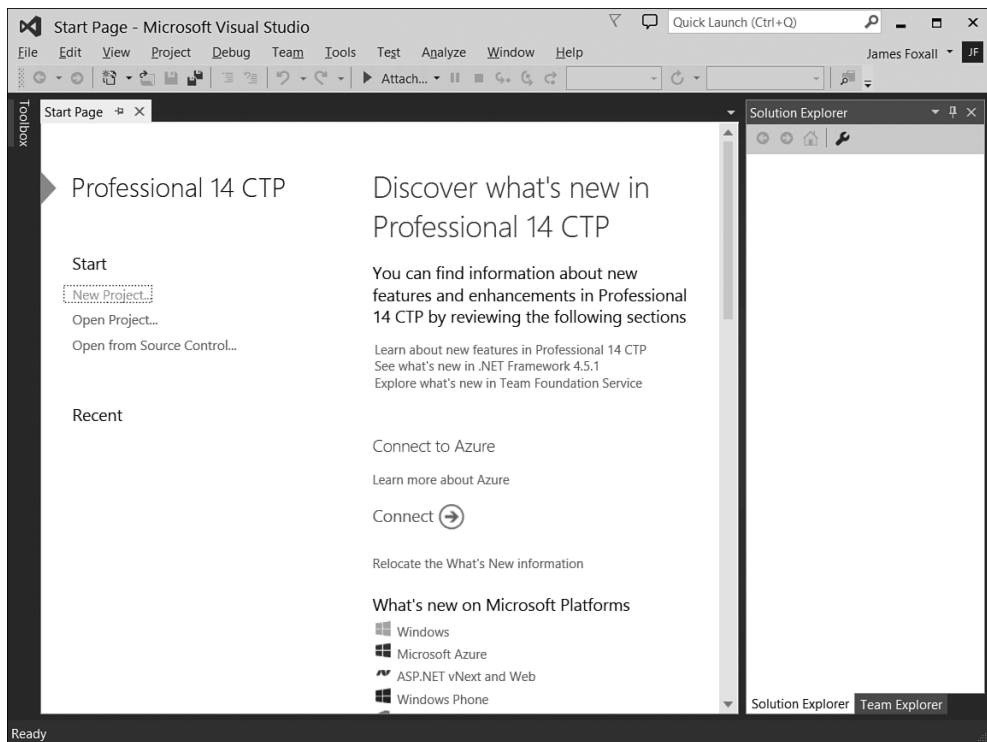
Visual Basic je, u prošlosti, bio autonoman programski jezik. To se sada promenilo. Visual Basic je sada deo većeg entiteta poznatijeg kao .NET okruženje. .NET okruženje obuhvata celokupnu .NET tehnologiju, uključujući vs .NET (paket alata za programiranje) i komponentu CLR (Common Language Runtime), koja predstavlja skup datoteka koje sačinjavaju jezgro svih .NET aplikacija. Sa ovim elementima ćete se detaljnije upoznati kasnije u ovoj knjizi. Za sada, treba da znate da je Visual Basic jedan od mnogobrojnih programskih jezika koji postoje u okruženju Visual Studio. Mnogi drugi programski jezici, kao što su C#, su takođe .NET programski jezici i koriste komponentu CLR i deo su okruženja Visual Studio

Visual Studio 2015 je kompletno okruženje za programiranje i skraćeno se označava sa IDE (integrated development environment). IDE je okruženje u kojem pravite aplikacije; svakoj alatki koja vam je potrebna za pravljenje Visual Basic programskog koda pristupate iz Visual Basic IDE-a. Ponavljam, u okruženju Visual Studio 2015 možete koristiti mnogobrojne programske jezike, a Visual Basic je najpopularniji od njih. Okruženje nije isto što i sam Visual Basic, ali programski jezik koji koristite u Visual Basicu 2015 jeste Visual Basic. Da biste radili sa Visual Basic projektima prvo morate da pokrenete okruženje Visual Basic 2015 IDE.

Pokrenite Visual Basic 2015, tako što ćete u meniju Start/Programs odabrati Microsoft Visual Basic 2015 Express Edition. Ukoliko koristite kompletnu komercijalnu verziju okruženja Visual Studio, prečica verovatno ima drugačije ime. Ako je tako, pronađite prečicu u meniju Start i pokrenite okruženje Visual Studio 2015 IDE.

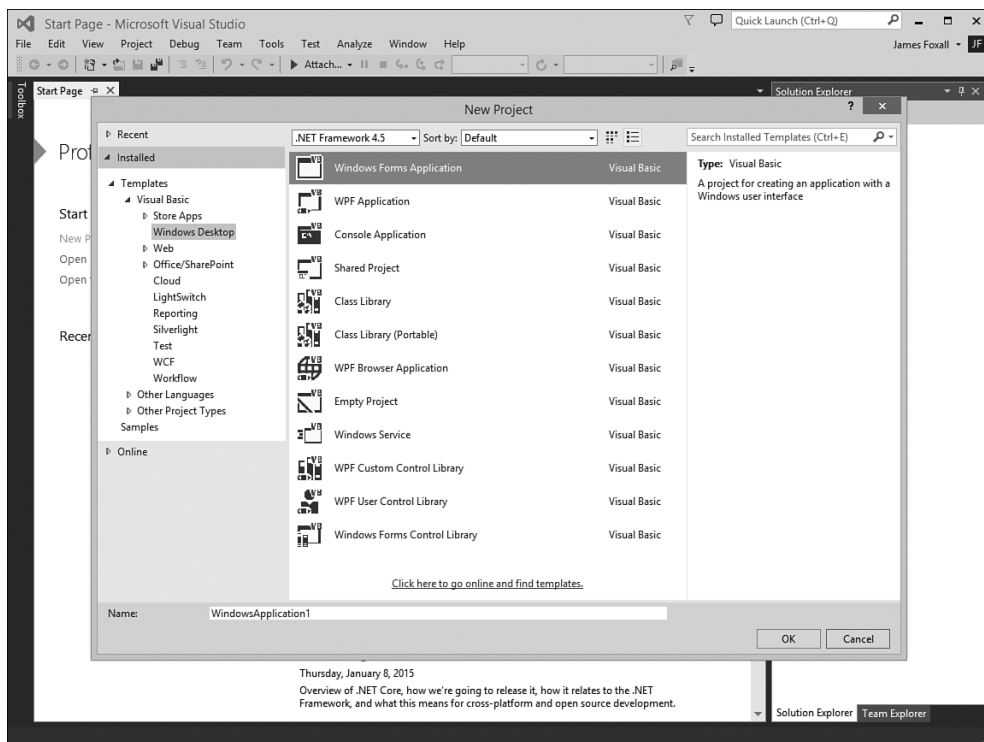
Započinjanje novog projekta

Kada prvi put pokrenete Visual Studio 2015 u IDE-u, vidite karticu Start Page (pogledajte sliku 1.1). Sa kartice Start Page možete pokrenuti prethodno napravljene projekte ili praviti nove projekte. U ovom kratkom pregledu okruženja napravićete novu Windows aplikaciju. Dakle, odaberite File, New Project da biste otvorili okvir za dijalog New Project, koji vidite na slici 1.2.



SLIKA 1.1

Sa kartice Start Page možete pokrenuti prethodno napravljene projekte ili praviti nove projekte.



SLIKA 1.2

Koristeći okvir za dijalog New Project možete napraviti razne tipove .NET projekata.

UZGRED BUDI REČENO

Kartica Start Page na vašem računaru se možda razlikuje od kartice koju vidite na slici 1.1, što zavisi od verzije okruženja Visual Studio koju koristite.

Okvir za dijalog New Project koristite kako biste zadali koji tip Visual Basic projekta pravite. (Koristeći Visual Basic možete da napravite razne vrste projekata, kao i koristeći druge programске jezike koji postoje u okruženju .NET.) Na slici 1.2 nisu prikazane sve opcije pošto ja, za sve primere u ovoj knjizi, koristim Express verziju Visual Basica. Ukoliko koristite kompletnu verziju okruženja Visual Studio, u okviru za dijalog će biti prikazano mnogo više opcija.

Da biste napravili nov projekat Windows Forms Application, treba da uradite sledeće:

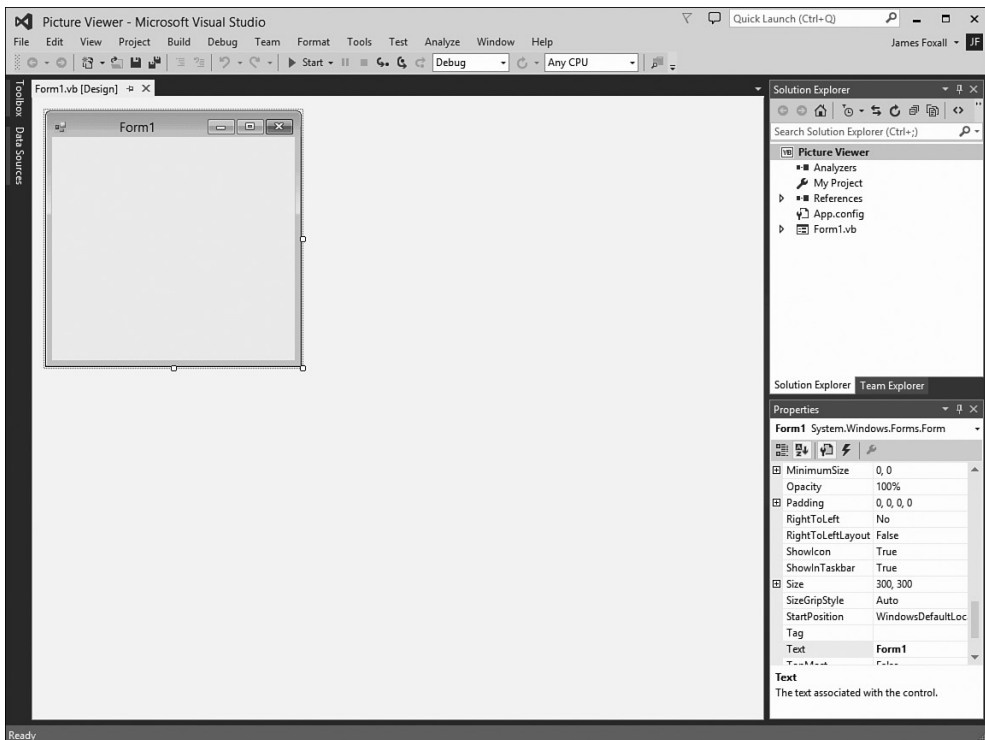
1. Među opcijama na levoj strani pronađite i kliknite Windows Desktop.
2. Kliknite element Windows Forms Application.

3. Na dnu okvira za dijalog New Project se nalazi tekstualno polje Name. Gle čuda, koristeći ovo polje zadajete ime projekta koji ćete praviti. U tekstualno polje Name unesite Picture Viewer.
4. Kliknite OK da biste napravili projekat.

DA LI STE ZNALI?

U tekstualno polje Name, pre nego što napravite projekat, uvek unesite smislen naziv. U suprotnom ćete, ukoliko budete hteli da premestite projekat ili mu promenite ime, imati mnogo više posla.

Kada Visual Basic pravi nov projekat Windows Forms Application, postavlja formular (prazan sivi prozor) kako biste mogli da počnete da pravite korisnički interfejs aplikacije (pogledajte sliku 1.3).



SLIKA 1.3

U novom projektu Windows Forms Application je prikazan prazan formular. Zabava tek počinje!

UZGRED BUDI REČENO

U okruženju Visual Studio 2015, termin formular se koristi za prozor koji se korisniku prikazuje prilikom dizajniranja.

Okruženje Visual Studio 2015, koje vi koristite, može se razlikovati od okruženja koje je prikazano na slikama u ovom poglavlju, što zavisi od verzije okruženja Visual Studio 2015 koje koristite, kao i od toga da li ste već koristili Visual Studio 2015 i drugih činilaca, na primer rezolucije vašeg monitora. Međutim, svi elementi koje obrađujem u ovom poglavlju postoje u svim verzijama okruženja Visual Studio 2015 koje podržavaju programiranje (nasuprot aplikacijama iz prodavnice). Ukoliko se prozor koji je prikazan na slici ne vidi u vašem IDE-u, upotrebite meni View da biste ga prikazali.

UZGRED BUDI REČENO

Da biste napravili program kojise može izvršavati na drugom računaru, najpre pravite projekat, a zatim projekat kompajlirate u komponentu, kao što je izvršna datoteka (program koji korisnik može izvršavati) ili DLL (komponentu koju će koristiti drugi programi ili komponente). Kompajliranje je postupak koji ću detaljnije objasniti u Lekciji 24, „Instaliranje aplikacija“. Zapamtite, ukoliko čujete da neko kaže da pravi program, kao što vi sada pravite program Picture Viewer, onda on misli na kompletiranje svih koraka do kompajliranja, i možda i na kompajliranje projekta radi dobijanja distributivne datoteke.

Upoznavanje korisničkog okruženja Visual Basicu 2015

Kada prvi put pokrenete Visual Studio 2015, primećujete da se IDE sastoji od brojnih prozora, kao što je prozor Properties, u donjem desnom uglu, koji se koristi za prikazivanje i zadavanje svojstava objekata. Pored tih prozora, IDE sadrži brojne kartice, kao što je uspravna paleta alata Toolbox i kartice Data Source na levoj strani IDE-a (pogledajte sliku 1.3). Probajte sledeće: kliknite karticu Toolbox da biste prikazali prozor Toolbox (kada kliknete karticu otvorićete odgovarajući prozor). Da biste sakrili prozor treba da kliknete neki drugi prozor. Da biste zatvorili prozor (nemojte to sada da uradite), kliknite Close (X) na naslovnoj liniji prozora.

Položaj i veličinu svakog od ovih prozora možete da prilagodite, a možete ih po potrebiti prikazati ili sakriti. U Lekciji 2, „Snalaženje u Visual Basicu 2015“, naučićete kako da prilagodite razvojno okruženje.

OBRATI PAŽNJU!

Nemojte, ukoliko drugačije nije rečeno, ništa dva puta da kliknete u razvojnom okruženju Visual Studio 2015. Ukoliko dva puta kliknete većinu objekata, uradićete nešto sasvim drugo nego kada jednom kliknete objekat. Ukoliko greškom dva puta kliknete objekat koji se nalazi na formularu (veoma brzo ću to objasniti), otvorićete prozor u koji se unosi programski kod. Na vrhu prozora u koji se unosi programski kod nalaze se dve kartice: jedna pomoću koje prelazite u prikaz za dizajniranje formulara i druga za prelazak u prozor za unošenje programskog koda. Kliknite karticu za prelazak na dizajniranje formulara da biste sakrili prozor za unošenje programskog koda.

Prozor Properties, koji se nalazi na desnoj strani razvojnog okruženja, je možda najvažniji prozor u IDE-u, i to je prozor koji ćete najčešće koristiti. Ukoliko ste za monitor zadali rezoluciju 1280x768, verovatno ćete videti samo nekoliko svojstava. Zbog toga nije lako zadavati vrednosti svojstvima dok pravite projekte. Sve slike koje su prikazane u ovoj knjizi su iz operativnog sistema Windows 8 u rezoluciji 1152x864 zbog ograničenja koja postoje u izdavaštvu, ali bi vi trebalo da koristite veću rezoluciju ukoliko je to moguće. Preporučujem vam da, kada budete pravili aplikacije pomoću Visual Basica, koristite rezoluciju 1280x768, ili veću rezoluciju, kako biste imali dovoljno radnog prostora. Da biste promenili rezoluciju treba da desnim tasterom miša kliknete radnu površinu operativnog sistema i u kontekstnom meniju odaberete Screen Resolution. Međutim, imajte na umu da korisnici vaših programa možda koriste nižu rezoluciju u odnosu na rezoluciju koju vi koristite prilikom programiranja.

Menjanje karakteristika objekata

Gotovo sve što koristite u Visual Basicu jeste objekat. Na primer, formulari su objekti, kao i svi elementi koje postavljate na formular da biste napravili korisnički interfejs, kao što su kontrole ListBox i Button. Postoje mnoge vrste objekata, pa se objekti klasifikuju prema tipu. Na primer, formular je objekat tipa Form, dok su elementi koje postavljate na formular objekti tipa Control, to jest kontrole. (U Lekciji 3, „Objekti i kolekcije“, ću detaljnije objasniti objekte.) Za neke objekte ne postoji fizička reprezentacija, već postoje samo u okviru programskog koda. Sa takvim objektima ćete se susresti u narednim poglavljima.

OBRATI PAŽNJU!

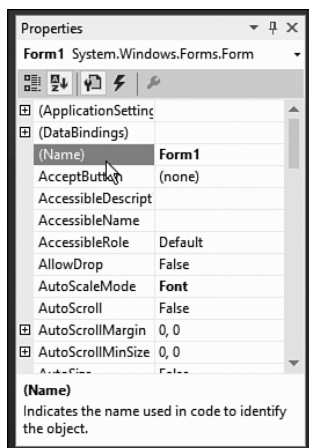
Videćete da ću često pominjati materijal koji ćete pročitati u narednom poglavljima. U izdavaštvu se to naziva referenciranje unapred. Neke ljude ovakve reference nerviraju. Ja koristim referenciranje unapred samo da biste shvatili da još uvek u potpunosti ne razumete temu koju obrađujem; ono što obrađujem će u narednim lekcijama biti detaljnije objašnjeno. Trudim se da referenciranje unapred što manje koristim, ali učenje programiranja, nažalost, nije linearan proces. Ponekad moram da se dotaknem teme za koju smatram da niste spremni da vas u potpunosti u nju uvedem. U takvim situacijama koristim referenciranje unapred da biste znali da je tema detaljnije obrađena u narednim lekcijama.

Svaki objekat ima poseban skup atributa koji se nazivaju svojstva (properties) (bez obzira da li objekat ima fizičku reprezentaciju). Karakteristike objekta su definisane njegovim svojstvima. Vi kao osoba imate izvesna svojstva, kao što su visina i boja kose. Objekti u Visual Basicu, takođe, imaju svojstva, kao što su Height (visina) i BackColor (boja pozadine). Kada pravite nov objekat, najpre morate da zadate njegova svojstva tako da objekat bude prikazan i da se ponaša onako kako ste zamislili. Da biste prikazali svojstva objekta, kliknite objekat u njegovom dizajneru (glavnoj radnoj površini u IDE-u).

Kliknite bilo gde unutar formulara (to je prozor naslovljen sa Form1) i videćete da su njegova svojstva prikazana u prozoru Properties. Kako ćete znati da su u prozoru Properties prikazana svojstva formulara? Tako što će na vrhu prozora Properties, u polju Name, biti prikazano ime formulara: Form1. Form1 je ime objekta, a njegov tip je System.Windows.Forms.Form.

Imenovanje objekata

Svojstvo koje, kada pravite nov objekat, uvek prvo treba da zadate jeste svojstvo Name. Pritisnite taster F4 da biste prikazali prozor Properties (ukoliko već nije prikazan) i potražite pri vrhu spiska svojstava svojstvo Name (pogledajte sliku 1.4). Ukoliko u spisku svojstava ne možete da pronađete svojstvo Name, onda je prozor Properties podešen tako da svojstva prikazuje po kategorijama, a ne prema imenu svojstava. Spisak možete urediti abecedno, ukoliko kliknete kontrolu Alphabetical koja se nalazi iznad spiska svojstava.



SLIKA 1.4

Kada u projekat dodajete nov objekat, prvo treba da zadate vrednost svojstvu Name.

UZGRED BUDI REČENO

Savetujem vam da svojstva u prozoru Properties budu poređana abecedno. Na taj način ćete lakše pronaći svojstvo koje pominjem u lekciji. Upamtite da se svojstvo Name uvek u spisku nalazi pri vrhu i da je prikazano kao (Name). Ukoliko se pitate zbog čega je ime svojstva prikazano unutar zagrada, onda znajte da jo to što se u spisku simboli prikazuju pre slova u abecednom uređenju spiska, razlog da se svojstvo Name uvek nalazi na početku spiska.

Kada projekat zapisujete na hard disk, takođe zadajete ime i mesto na koje će biti zapisan projekat i njegove datoteke. Kada unutar projekta napravite objekat, Visual Basic mu dodeljuje jedinstveno opšte ime na osnovu tipa objekta. Iako su tako dodeljena imena potpuno funkcionalna, ipak nisu dovoljno opisna za praktičnu primenu. Na primer, Visual Basic je formularu dodelio ime Form1, ali je uobičajeno da u projektu postoji stotine (pa čak i hiljade) formulara. To znači da bi takav projekat bilo izuzetno teško održavati, pošto se imena formulara razlikuju samo po broju (Form2, Form3 i tako dalje).

UZGRED BUDI REČENO

Ono što, zapravo, koristite jeste klasa Form, odnosno šablon za pravljenje formulara, na osnovu koje će formular biti napravljen i prikazan prilikom izvršavanja programa. U ovoj lekciji ću ovu klasu jednostavno zvati formular. Pogledajte Lekciju 5, „Osnove izrade formulara“, za više informacija na ovu temu.

Da biste bolje upravljali formularima, svakom od njih zadajte opisno ime. Visual Basic vam omogućava da imenujete formular čim ga dodate u projekat. Visual Basic je podrazumevani formular napravio prilikom pravljenja projekta, pa niste imali mogućnost da mu promenite ime. Važno je ne samo da imenujete formular, već i da imenujete njemu pripadajuću datoteku. Ime formulara i ime njemu pripadajuće datoteke ćete promeniti na sledeći način:

1. Kliknite svojstvo Name i umetok teksta Form1 unesite ViewerForm. Primetićete da na ovaj način niste promenili ime datoteke koja odgovara fomularu (pogledajte prozor Solution Explorer koji se nalazi iznad prozora Properties).
2. Desnim tasterom miša kliknite Form1.vb u prozoru Solution Explorer (prozor koji je prikazan iznad prozora Properties).
3. U kontekstnom meniju, koji će biti prikazan, odaberite Rename.
4. Umesto teksta Form1.vb unesite ViewerForm.vb

UZGRED BUDI REČENO

Sufiks Form koristim da bih istakao da datoteka sadrži objekat klase Form. Sufikse nije neophodno koristiti, mada smatram da su veoma korisni radi boljeg organizovanja elemenata projekta.

Vrednost svojstva Name formulara se automatski menja kada promenite ime njemu pripadajuće datoteke. U narednim primerima ću vas uputiti da promenite ime datoteke da biste tako automatski promenili vrednost svojstva Name. Hteo sam da vrednost svojstva promenite u prozoru Properties da biste videli kako se taj prozor koristi.

Zadavanje svojstva Text formulara

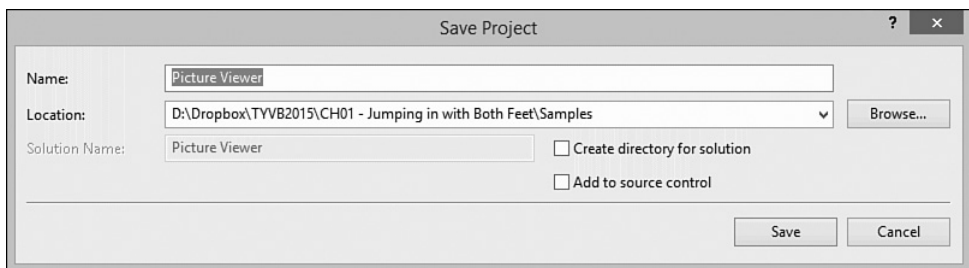
Primiteli ste da se u naslovnoj liniji formulara prikazuje tekst Form1. Visual Basic u naslovnoj liniji formulara prikazuje ime formulara (kada dodajete formular u projekat), ali taj tekst neće promeniti kada promenite ime formulara. Tekst koji se prikazuje u naslovnoj liniji jeste vrednost svojstva Text. Tekst koji se prikazuje u naslovnoj liniji ćete promeniti na sledeći način:

1. Ponovo kliknite formular kako bi u prozoru Properties bila prikazana njegova svojstva.
2. U prozoru Properties pronađite svojstvo Text. Ukoliko imate sreće, Visual Basic je već obeležio to svojstvo.
3. Promenite tekst u **Picture Viewer**. Pritisnite taster Enter ili taster Tab, ili kliknite ime nekog drugog svojstva kako biste „zadali“ promenu imena. Videćete da je tekst u naslovnoj liniji formulara promenjen.

Zapisivanje projekta na hard disk

Izмене koje ste do sada napravili postoje samo u memoriji računara. Kada biste sada isključili računar, sve izmene koje ste napravili do isključivanja računara bi bile nepovratno izgubljene. Naviknite se da često zapisujete izmene na hard disk.

Na paleti alatki kliknite Save All (komanda je predstavljena sličicom flopi diska) da biste na hard disk zapisali sve što ste do sada uradili. Visual Basic će prikazati okvir za dijalog Save Project koji je prikazan na slici 1.5. Primitićete da je svojstvo Name već popunjeno jer ste prilikom pravljenja projekta zadali ime. U polje Location unesite mesto gde treba zapisati projekat. Na zadanom mestu Visual Basic će napraviti poddirektorijum koristeći ime iz tekstualnog polja Name (u ovom slučaju je to Picture Viewer). Možete prihvatiti predloženu lokaciju, ili je promeniti po želji. Možete zadati da Visual Basic napravi direktorijum za rešenje, a ako to uradite, Visual Basic će napraviti rešenje u direktorijumu i napraviće poddirektorijum u koji će zapisati projekat i datoteke. Kada radite na velikim projektima, ovakva mogućnost je dobrodošla. To u našem primeru nije neophodan korak, te stoga nemojte potvrditi polje Create Directory for Solution, ukoliko je potvrđeno, pa kliknite Save kako biste projekat zapisali na hard disk.



SLIKA 1.5

Kada na hard disk zapisujete projekat, zadajte ime i lokaciju za projekat i pripadajuće datoteke.

Zadavanje ikone za formular

Svako ko je radio u operativnom sistemu Windows je upoznat sa konceptom ikona – malih sličica kojima su predstavljeni programi. Ikone se najčešće prikazuju u meniju Start pored imena programa. U Visual Basicu ne samo da možete da odaberete ikonu za program, već i svakom formularu možete pridružiti jedinstvenu ikonu, ukoliko to želite.

UZGRED BUDI REČENO

Pretpostavio sam da imate pristup datotekama za primere iz ove knjige i na osnovu te pretpostavke sam osmislio naredne korake. Datoteke za primere u ovoj knjizi ćete pronaći na adresi <http://www.sampublishing.com>. Datoteke, takođe, možete preuzeti iz odeljka Download na mom veb sajtu koji se nalazi na adresi <http://www.jamesfoxall.com>. Kada budete raspakovali primere, za svaku lekciju će biti napravljen poseban direktorijum, a u okviru svakog od njih će biti napravljeni poddirektorijumi koji će sadržati primere projekata. Ikonu koju ćete upotrebiti u ovom primeru ćete pronaći u direktorijumu Hour01\Samples.

Naravno, ne morate upotrebiti ikonu koju sam ja pripremio za ovaj primer; upotrebite ikonu koju god poželite. Ukoliko nemate datoteku koja sadrži ikonu, preskočite ovaj odeljak bez bojazni da će se to odraziti na konačan rezultat.

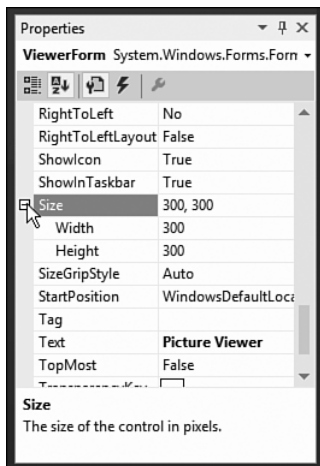
Da biste formularu pridružili ikonu, treba da uradite sledeće:

1. U prozoru Properties kliknite svojstvo `Icon`.
2. Kada budete kliknuli svojstvo `Icon`, pojaviće se kontrola, desno od svojstva, na kojoj su prikazane tri tačke. Kliknite tu kontrolu.
3. Koristeći okvir za dijalog Open pronađite datoteku `Picture Viewer.ico`, ili neku drugu datoteku koja sadrži ikonu koju želite da upotrebite. Kada pronađete ikonu, dva puta kliknite ime datoteke, ili kliknite ime datoteke i potom kliknite Open.

U svojstvu `Icon` se, pošto odaberete ikonu, prikazuje ta ikona, kao i reč `Icon`. Mala verzija ikone će, takođe, biti prikazana u gornjem levom uglu formulara. Svaki put kada formular bude bio minimizovan, na paleti aktivnih poslova Windowsa će biti prikazana odabrana ikona.

Menjanje veličine formulara

Sada ćete promeniti vrednosti svojstava `Height` i `Width`. Svojstva `Height` i `Width`, i njihove vrednosti, se nalaze unutar svojstva `Size`. Širina (`Width`) je vrednost levo od zareza, a visina (`Height`) je vrednost desno od zareza. Vrednosti svojstava `Height` i `Width` možete promeniti promenom odgovarajuće vrednosti u svojstvu `Size`. Obe vrednosti se odnose na veličinu u pikselima. (Dakle, formular za koji svojstvo `Size` ima vrednost 200,350 je 200 piksela širok i 350 piksela visok.) Da biste prikazali i podelili svojstva `Height` i `Width` odvojeno, kliknite znak plus koji je prikazan pored svojstva `Size`. (Pošto kliknete znak plus, umesto njega će biti prikazan znak minus, kao na slici 1.6.)

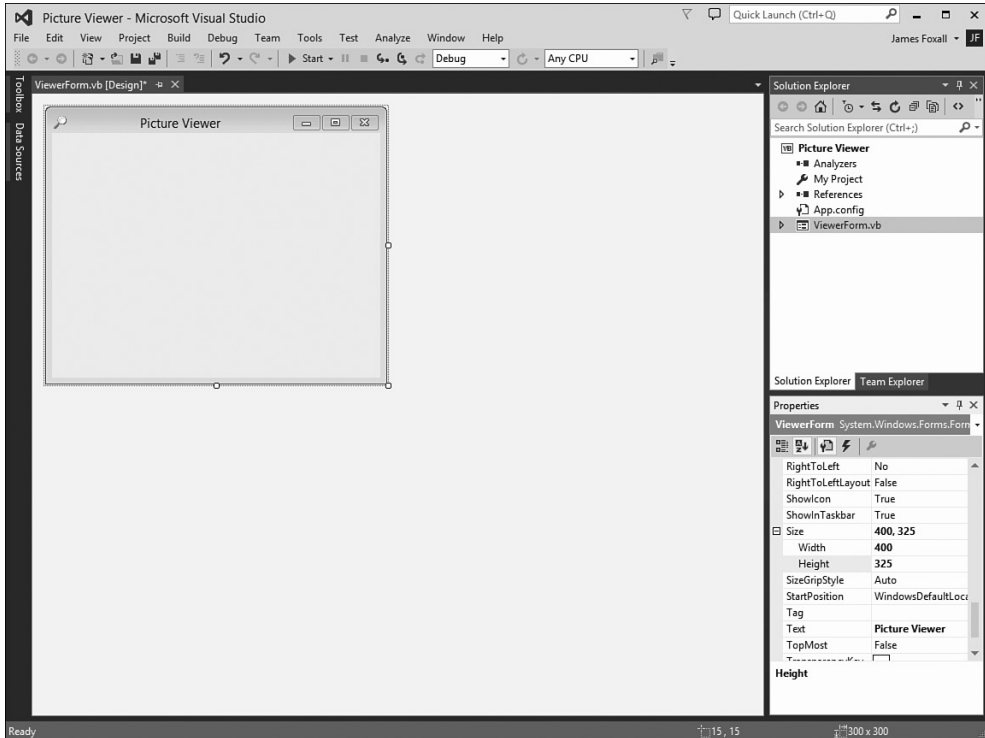
**SLIKA 1.6**

Za neka svojstva postoje podsvojstva.

UZGRED BUDI REČENO

Piksel je merna jedinica za merenje slike koja se prikazuje na monitoru. Piksel je najmanja vidljiva „tačka“ na ekranu monitora. Rezolucija monitora se uvek izražava u pikselima, na primer 800x600 ili 1024x768. Kada vrednost svojstva povećate ili umanjite za 1 piksel, to znači da vrednost svojstva menjate za najmanju moguću veličinu.

Svojstvu `Width` dodelite vrednost 400, a svojstvu `Height` vrednost 325 tako što ćete te vrednosti uneti u odgovarajuća polja pored imena svojstava. Potvrdite izmene, tako što ćete pritisnuti taster `Enter` ili taster `Tab`, ili tako što ćete kliknuti neko drugo svojstvo ili prozor. Na ekranu bi trebalo da bude prikazano ono što vidite na slici 1.7.



SLIKA 1.7

Izmene koje naćinite u prozoru Properties će biti uoćljive ćim ih „primenite“.

UZGRED BUDI REĆENO

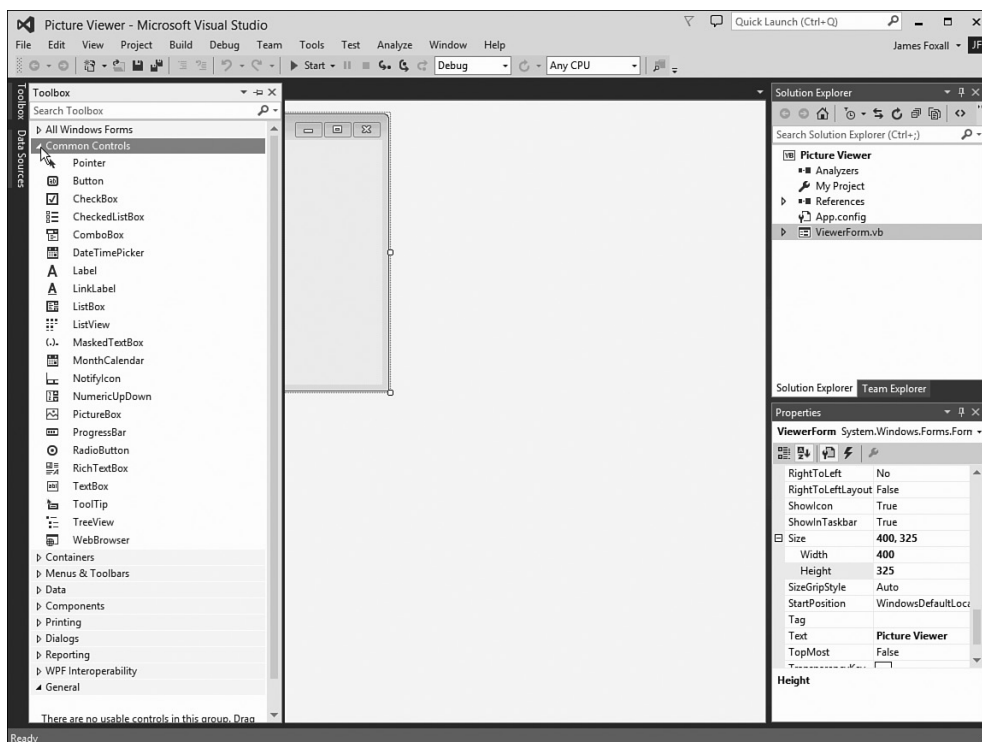
Velićinu formulara moćžete promeniti tako Ńto ćete povući njegove bordure, Ńto ćete naućiti u Lekciji 2, ili koristeći programski kod pomoću kojeg ćete promeniti vrednosti svojstava, Ńto ćete naućiti u Lekciji 5..

ZapiŃite projekat na hard disk, tako Ńto ćete u meniju File odabrati Save All ili tako Ńto ćete na paleti alatki kliknuti komandu Save All – predstavljena je ikonom na kojoj su prikazana dva flopi diska.

Postavljanje kontrola na formular

PoŃto ste zadali vrednosti za inicijalna svojstva formulara, vreme je da napravite korisnićki interfejs tako Ńto ćete na formular postaviti objekte. Objekti koje moćžete da postavite na formular se nazivaju *kontrola*. Za neke kontrole postoji vidljiv interfejs koji korisnik moće da koristi, dok se druge nikada ne prikazuju korisniku. U ovom primeru ćete upotrebiti oba tipa kontrola. Na levo

strani ekrana nalazi se vertikalna kartica, označena sa Toolbox. Kliknite karticu Toolbox kako biste otvorili prozor Toolbox, a da biste i videli kontrole koje se najčešće koriste otvorite odeljak Common Controls (pogledajte sliku 1.8).



SLIKA 1.8

Karticu Toolbox koristite da biste birali komponente koje će sačinjavati korisnički interfejs.

Ukoliko kliknete izvan prozora Toolbox, onda će taj prozor isčeznuti sa ekrana. Da bi kartica Toolbox stalno bila prikazana, čak i kada kliknete neki drugi element korisničkog interfejsa IDE-a, kliknite sličicu čiode koja se nalazi u naslovnoj liniji prozora Toolbox.

Ne želim da kontrole postavite na formular (barem ne još uvek, kasnije ću vam objasniti kako to da uradite), ali treba da znate da će se korisnički interfejs programa Picture Viewer sastojati od sledećih kontrola:

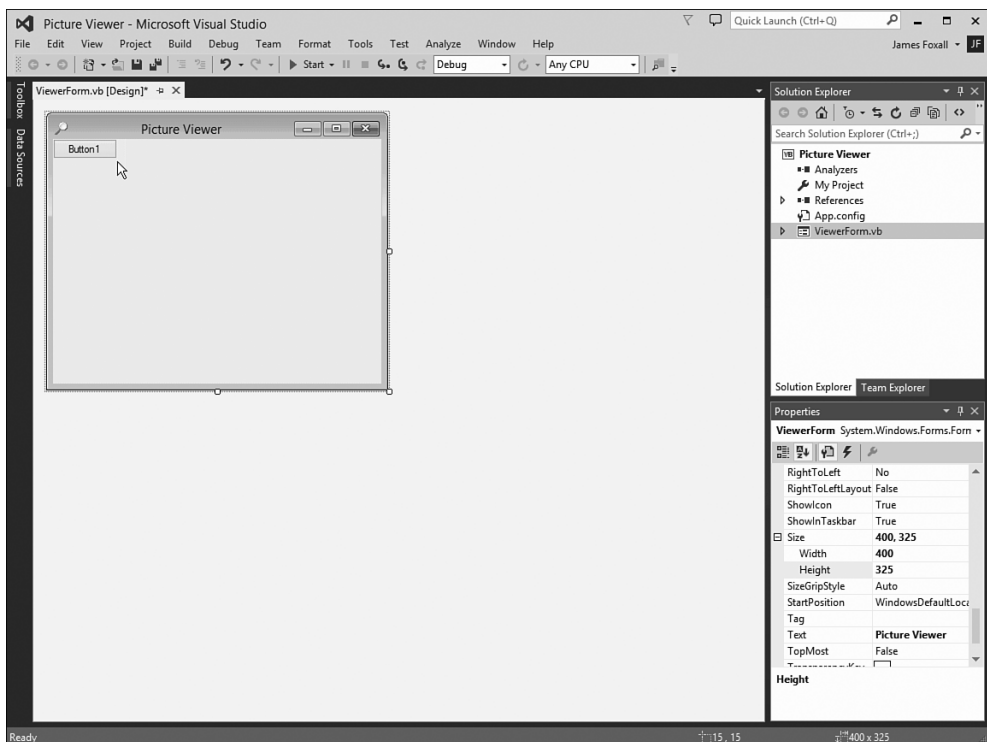
- ▶ **Dve kontrole Button:** standardne kontrole (komandna dugmad) na koja ste navikli u gotovo svakom programu koji ste pokrenuli u operativnom sistemu Windows.
- ▶ **Kontrola PictureBox:** kontrola koja se koristi za prikazivanje slika.
- ▶ **Kontrola OpenFileDialog:** skrivena kontrola koja omogućava da upotrebite Windowsov okvir za dijalog Open File.

Dizajniranje korisničkog interfejsa

Najbolje bi bilo da prvo dizajnirate korisnički interfejs formulara, pa tek onda da napišete programski kod koji će taj interfejs učiniti funkcionalnim. U narednim odeljcima ćete napraviti korisnički interfejs.

Postavljanje vidljivih kontrola na formular

Prvo ćete na formular postaviti kontrolu `Button`. To ćete uraditi tako što ćete u prozoru `Toolbox` dva puta kliknuti element `Button`. `Visual Basic` će napraviti novo komandno dugme i postaviti ga u gornji levi ugao formulara. Kliknite bilo gde formular da biste zatvorili prozor `Toolbox` i tako videli gde je postavljena kontrola (pogledajte sliku 1.9).



SLIKA 1.9

Kontrola, koju dva puta kliknete u prozoru `Toolbox`, će biti postavljena u gornji levi ugao formulara.

U prozoru `Properties` zadajte svojstva za kontrolu `Button`, onako kako je prikazano u sledećoj tabeli. Pošto ste kliknuli formular i tako sakrili prozor `Toolbox`, kliknite kontrolu `Button` na formularu pa joj, zatim, promenite svojstva. Upamtite, kada su svojstva poredana abecedno, svojstvo `Name` je prvo u spisku svojstava, pa ga nemojte tražiti u spisku. U suprotnom, potrošićete nešto vremena pre nego što ga pronađete.

Svojstvo	vrednost
Name	btnSelectPicture
Location	295,10 (295 je x koordinata: 10 je y koordinata)
Size	85,23
Text	Select Picture

UZGRED BUDI REČENO

Ukoliko na komandnom dugmetu možete da vidite samo reč Select, vrlo je verovatno da ste u operativnom sistemu Windows veličinu slova podesili tako da je veličina veća od standardne veličine. Desnim tasterom miša kliknite radnu površinu i u kontekstnom meniju odaberite Personalize. Zatim, kliknite Display u donjem desnom uglu i promenite veličinu slova u okviru za dijalog Display.

Sada ćete napraviti komandno dugme koje će korisnik koristiti da prekine izvršavanje programa Picture Viewer. Iako na formular možete postaviti drugo komandno dugme tako što ćete dva puta kliknuti kontrolu `Button` u prozoru Toolbox, ovoga puta ćete kontrolu na formular postaviti tako što ćete napraviti kopiju komandnog dugmeta koje ste upravo napravili. Na ovaj način ćete lako napraviti komandno dugme čija veličina i drugi atributi odgovaraju komandnom dugmetu koje ste ranije napravili.

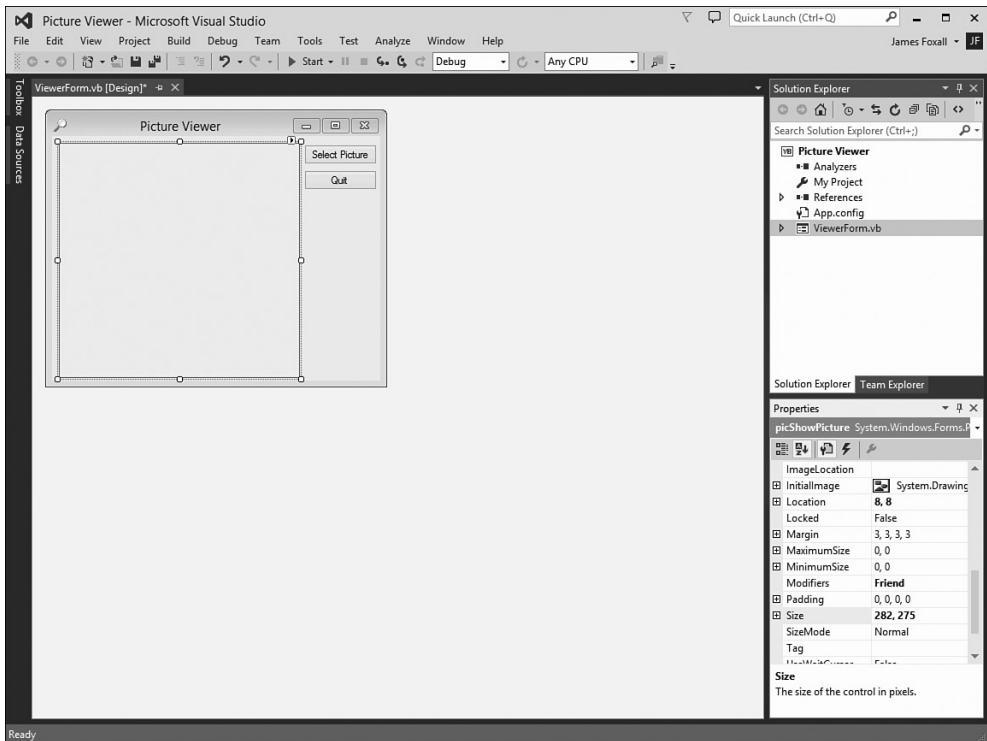
Da biste to uradili, desnim tasterom miša kliknite komandno dugme `Select Picture` i u kontekstnom meniju formulara odaberite `Copy`. Zatim, desnim tasterom miša kliknite formular i u kontekstnom meniju formulara odaberite `Paste`. (Takođe, možete upotrebiti tastaturne prečice `Ctrl+C` da biste kopirali kontrolu i `Ctrl+V` da biste zalepili kontrolu.) Novo komandno dugme će biti postavljeno na sredini formulara i odmah će biti selektovano. Primetićete da zadržava gotovo sva svojstva originalnog komandnog dugmeta, ali da je ime promenjeno. Za novo komandno dugme zadajte sledeća svojstva

Svojstvo	vrednost
Name	btnQuit
Location	295,40
Text	Quit

Poslednja vidljiva kontrola koju treba da postavite na formular jeste kontrola `PictureBox`. Kontrola `PictureBox` ima različite primene, ali je njena osnovna uloga da prikazuje slike, a to je upravo ono zbog čega pravimo ovaj primer. Da biste kontrolu `PictureBox` postavili na formular, dva puta kliknite element `PictureBox` u prozoru Toolbox i zadajte sledeća svojstva za tu kontrolu:

Svojstvo	vrednost
Name	picShowPicture
BorderStyle	FixedSingle
Location	8, 8
Size	282, 275

Pošto za svojstva zadate vrednosti, formular će izgledati kao na slici 1.10. Na paleti alatki kliknite Save All da biste promene zapisali na hard disk.



SLIKA 1.10

Korisnički interfejs aplikacije ne mora biti složen da bi bio koristan.

Postavljanje nevidljivih kontrola na formular

Sve kontrole koje ste do sada koristili se nalaze na formularu i imaju fizičku reprezentaciju kada korisnik pokrene aplikaciju. Međutim, nemaju sve kontrole fizičku reprezentaciju. Takve kontrole se nazivaju nevidljive kontrole (*nonvisual controls*) ili kontrole koje se ne prikazuju tokom izvršavanja programa (*invisible-at-runtime controls*) i nisu dizajnirane tako da korisnik ima direktnu interakciju sa njima. Takve kontrole su dizajnirane da vama, programeru, pruže funkcionalnost koja nije deo standardnih funkcija Visual Basica.

Da biste korisnicima omogućili da odaberu sliku koju će prikazati u programu, na primer, morate im omogućiti da na hard disku pronađu datoteku. Sigurno ste приметili da će, kad god odaberete da otvorite datoteku u bilo kojoj Windows aplikaciji, biti otvoren okvir za dijalog koji je uvek isti. Pošto nema mnogo smisla da primoravate svakog programera da piše programski kod za obavljanje standardnih operacija sa datotekama, Majkrosoft je takvu funkcionalnost učinio dostupnom kroz kontrolu koju koristite u projektima. To je kontrola `OpenFileDialog`, što znači da nećete morati da utrošite desetine sati da biste ostvarili funkcionalnost koja već postoji.

UZGRED BUDI REČENO

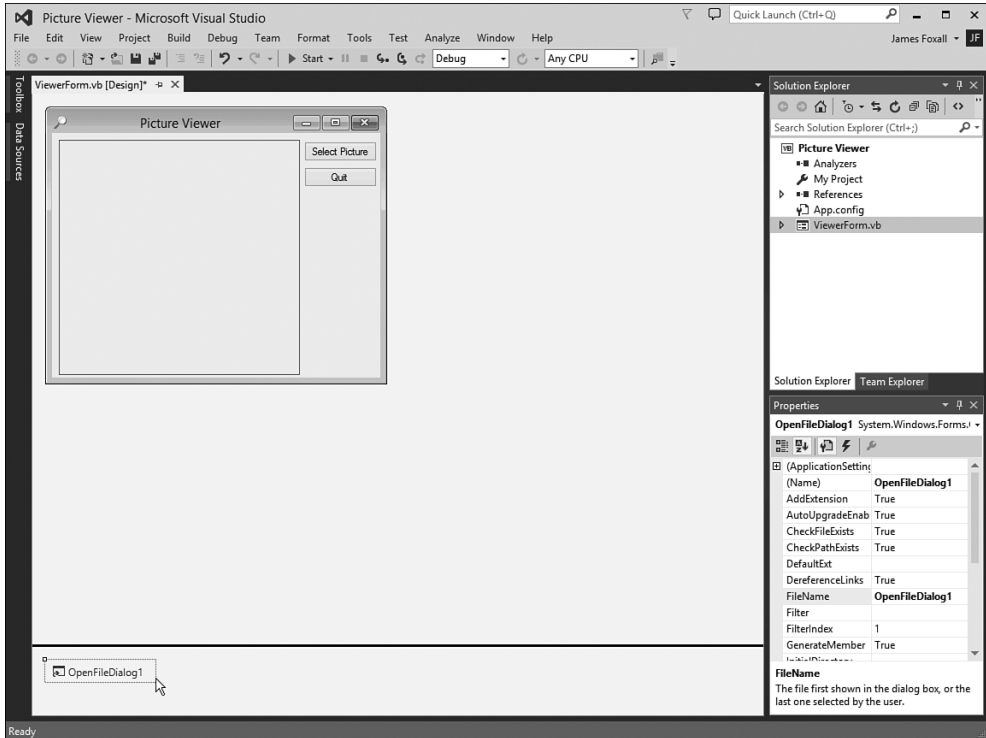
Postoje i druge kontrole, osim kontrole `OpenFileDialog`, koje vam omogućavaju rad sa datotekama. Na primer, kontrola `SaveFileDialog` omogućava korisniku da zada ime datoteke i mesto na koje treba zapisati datoteku.

Otvorite prozor Toolbox i pronađite kategoriju Dialogs. Otvorite kategoriju Dialogs i dva puta kliknite kontrolu `OpenFileDialog` da biste je dodali formularu. Videćete da kontrola neće biti postavljena na formular. Umesto toga, kontrola će biti prikazana u posebnom delu ispod formulara (pogledajte sliku 1.11).

To je zato što za kontrolu `OpenFileDialog` ne postoji interfejs koji se korisniku prikazuje na formularu. Za kontrolu postoji interfejs (okvir za dijalog) koji se prikazuje po potrebi, ali se ne prikazuje direktno na formularu.

Označite (selektujte) kontrolu `OpenFileDialog` i zdajte vrednosti za sledeća svojstva:

Svojstvo	vrednost
Name	<code>ofdSelectPicture</code>
Filename	<code><make empty></code>
Filter	<code>PNG Files *.png Windows Bitmaps *.bmp JPEG Files *.jpg</code>
Title	<code>Select Picture</code>

**SLIKA 1.11**

Kontrolne za koje ne postoji interfejs se prikazuju ispod formulara.

UZGRED BUDI REČENO

Ne treba da <make empty> unosite kao vrednost svojstva `Filename`. Hteo sam da naznačim da treba da obrišete podrazumevanu vrednost i da za svojstvo ne postoji zadata vrednost.

Pomoću svojstva `Filter` ograničavate koji tipovi datoteka će biti prikazani u okviru za dijalog `Open File`. Format u kojem se zadaje vrednost za svojstvo `Filter` je `opis|filter`. Tekst koji je prikazan ispred simbola `|` jeste opis tipa datoteke, dok se tekst iza simbola `|` koristi za filtriranje datoteka. Više kriterijuma filtriranja možete zadati tako što ćete svaki par `opis|filter` od sledećeg odvojiti simbolom `|`. Tekst koji unesete u svojstvo `Title` je tekst koji se prikazuje u naslovnoj liniji okvira za dijalog `Open File`.

Sada smo završili pravljanje korisničkog interfejsa za program `Picture Viewer`. Ukoliko ste u prozoru `Toolbox` kliknuli sličicu čiode, kliknite je ponovo da biste zatvorili prozor `Toolbox`. Na paleti alatki kliknite `Save All` da biste projekat zapisali na hard disk.

Pisanje programskog koda za korisnički interfejs

Da bi program mogao da obavlja određene poslove i da reaguje na komande korisnika, morate da napišete programski kod. Visual Basic je programski jezik koji akcije obavlja kada se pokrenu događaji, što znači da se programski kod izvršava za pokrenute događaje. Događaje može da inicira korisnik, na primer kada korisnik klikne komandno dugme i tako inicira događaj `Click`, ili operativni sistem Windows (pročitajte Lekciju 4, „Događaji“, u kojoj detaljno objašnjavam događaje.) Za sada aplikacija izgleda sasvim dobro, ali ništa se pomoću nje ne može uraditi. Korisnici mogu da kliknu komandno dugme `Select Picture` koliko god puta žele, a time samo rizikuju da dobiju sindrom karpalnog tunela, ali se ništa neće dogoditi, pošto u programu niste zadali šta treba da se dogodi kada korisnik klikne to komandno dugme. Možete se i sami uveriti tako što ćete pokrenuti projekat pritiskom na taster F5. Slobodno kliknite bilo koje komandno dugme i videćete da se ništa neće dogoditi. Kada isprobate program, zatvorite njegov prozor da biste se vratili u režim dizajniranja programa.

Treba da napišete programski kod da bi program mogao da obavi dva zadatka. Prvo, treba korisnicima da omogućite da pregledaju sadržaj hard diska i odaberu datoteku koja sadrži sliku, a zatim da je prikažu u kontroli `PictureBox`. (Ovako rečeno izgleda mnogo teže nego što zaista jeste.) Drugo, za komandno dugme `Quit` treba da napišete programski kod pomoću kojeg se prekida izvršavanje programa kada ga korisnik klikne.

Omogućite korisniku da pronade datoteku

Prvi deo programskog koda koji ćete napisati omogućiće korisnicima da pretražuju hard disk, odaberu datoteku koja sadrži sliku i da sliku prikažu u kontroli `PictureBox`. Programski kod koji budete napisali će se izvršavati kada korisnik bude kliknuo komandnog dugme `Select Picture`. Zbog toga se ovaj programski kod upisuje u događaj `Click` komandnog dugmeta.

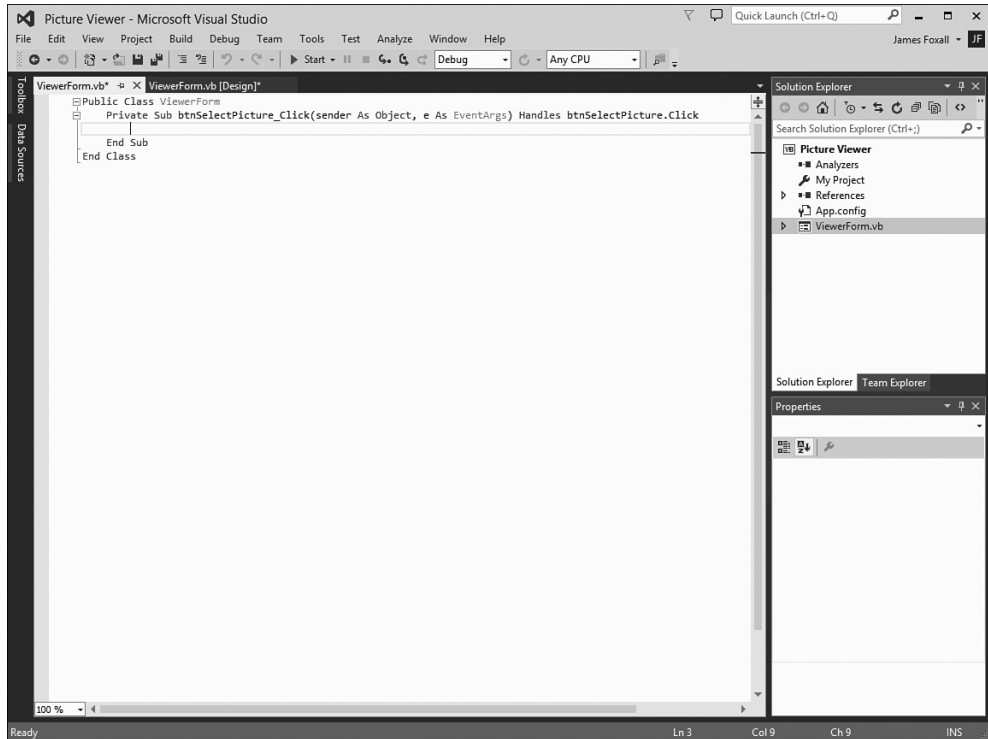
Kada u pogledu `Design` dva puta kliknete formular, u prozoru za pisanje programskog koda će biti prikazan podrazumevani događaj za kontrolu koju ste kliknuli. Podrazumevani događaj za kontrolu `Button` je događaj `Click`, što ima smisla, jer je klik mišem najčešća akcija koju korisnik ima nad komandnim dugmetom. Dva puta kliknite komandno dugme `Select Picture` kako biste u prozoru za pisanje programskog koda prikazali događaj `Click` (pogledajte sliku 1.12).

Kada otvorite događaj, Visual Basic pravi *obradu događaja*, što u osnovi predstavlja šablon procedure u kojoj pišete programski kod koji se izvršava kada se događaj inicira. Cursor se postavlja unutar procedure, tako da preostaje samo da unesete programski kod. Iako ovaj postupak na prvi pogled može da vas odvrti od pisanja programskog koda, ipak ćete, kada budete pročitali knjigu, sa radošću pisati svoj programski kod kako biste aplikacije naterali da rade baš ono što ste zamislili – pa, dobro, u najvećem broju slučajeva. Ipak, za sada, unesite programski kod koji ću vam pokazati.

Veoma je važno da steknete naviku da u programskom kodu pišete komentare, pa će zato prvi programski red sadržati komentar. Takav programski red počinje apostrofom (`'`), što znači da tekst iza apostrofa predstavlja komentar. Kompajler neće ništa uraditi sa takvim programskim

redom, tako da iza apostrofa možete da unesete kakav god tekst želite. Sledeći programski red unesite onako kako je prikazan i potom pritisnite taster Enter:

```
' Show the open file dialog box.
```



SLIKA 1.12

Programski kod ćete unositi u ovakvom prozoru.

Pomoću narednog programskog reda se poziva metod kontrole `OpenFileDialog` koju ste postavili na formular. Metod možete poistovetiti sa mehanizmom koji omogućava da kontrola nešto „obavi“. Metod `ShowDialog()` „govori“ kontroli da treba da otvori okvir za dijalog `Open` i omogući korisniku da odabere datoteku. Metod `ShowDialog()`, kao rezultat rada, proizvodi vrednost koja označava uspešno ili neuspešno izvršavanje, vrednost koju ćete uporediti sa podrazumevanim rezultatom (`DialogResult.OK`). Nemojte mnogo lupati glavu oko toga šta se dešava, u narednim lekcijama ću vam sve detaljnije objasniti. Cilj ove lekcije je da saznate kako se programira. Metod `ShowDialog()` će biti pozvan da bi korisniku omogućio da pronađe datoteku. Ukoliko korisnik odabere datoteku, izvršiće i neki drugi programski kod. Naravno, okvir za dijalog `Open` se može koristiti i u drugim slučajevima pored ovog koji je prikazan u primeru, ali je ovaj programski red sasvim dovoljan za otvaranje okvira za dijalog. Unesite sledeći programski red i potom pritisnite taster Enter. (Ne brinite o velim i malim slovima, Visual Basic će umesto vas sve dovesti u red.)

```
If ofdSelectpicture.ShowDialog = DialogResult.OK Then
```

UZGRED BUDI REČENO

Pošto unesete programski red koji počinje sa `If` i potom pritisnete `Enter`, Visual Basic će automatski, umesto vas, dodati programski red `End If`. Ukoliko ste uneli `End If`, onda će postojati dva programska reda `End If` i programski kod neće moći da bude izvršen. Ukoliko je to slučaj, obrišite jedan programski red `End If`. U Lekciji 13, „Donošenje odluka korišćenjem Visual Basic programskog koda“, ćete naučiti sve što treba da znate o iskazu `If`.

Vreme je da napišete još jedan komentar. Kursor se nalazi između programskih redova `If` i `End If`. Ne pomerajte kursor i unesite sledeći programski red i ne zaboravite da na kraju reda pritisnete taster `Enter`:

```
' Load the picture into the picture box.
```

DA LI STE ZNALI?

Ne morate da pazite na uvlačenje programskih redova korišćenjem tastera `Tab` ili razmaka. Visual Basic automatski pomera programske redove.

Sledeći programski red, koji treba da bude unutar konstrukta `If` (između iskaza `If` i `End If`) je programski red pomoću kojeg se slika prikazuje u kontroli `PictureBox`.

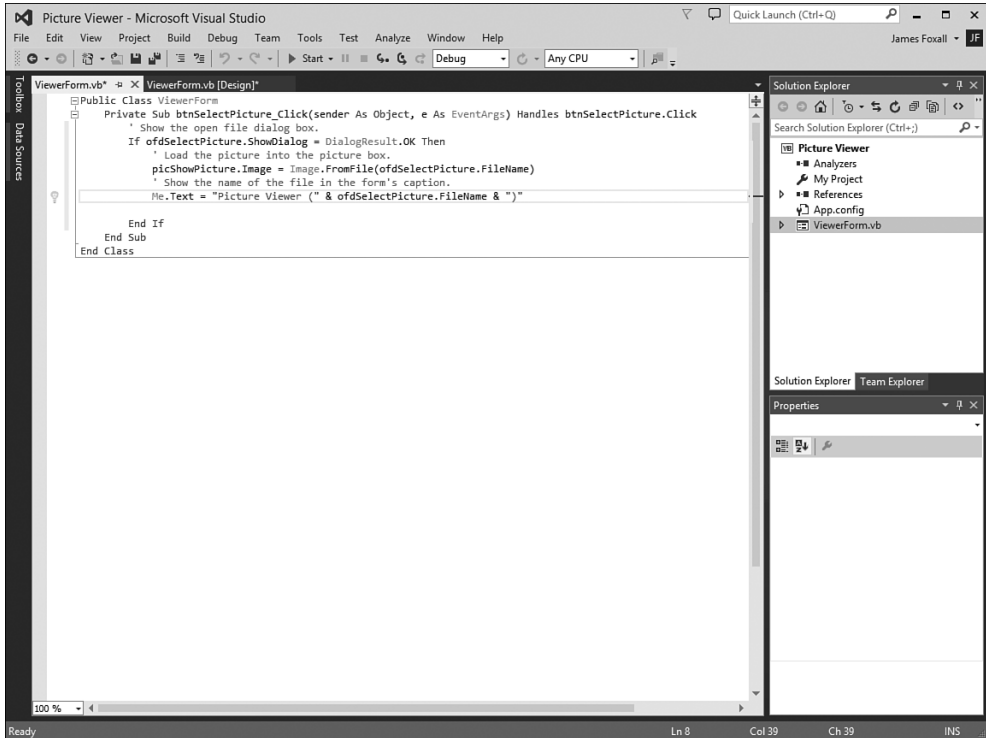
Unesite sledeći programski red i potom pritisnite taster `Enter`:

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

Osim što se u programu prikazuje slika, program će u naslovnoj liniji prikazati lokaciju datoteke i njeno ime. U prozoru `Properties` ste, kada ste napravili formular, promenili vrednost svojstva `Text`. Da biste napravili dinamičke aplikacije svojstva, stalno treba da menjaju vrednosti tokom izvršavanja aplikacije, a to se postiže pisanjem programskog koda. Unesite sledeća dva programska reda i na kraju svakog od njih pritisnite taster `Enter`:

```
' Show the name of the file in the form's caption.  
Me.Text = "Picture Viewer (" & ofdSelectPicture.FileName & ")"
```

Pošto unesete ceo programski kod, prozor za unošenje programskog koda treba da izgleda kao na slici 1.13.



SLIKA 1.13

Postarajte se da programski kod bude identičan programskom kodu koji je prikazan na ovoj slici.

Prekidanje izvršavanja programa pomoću programskog koda

Poslednji deo programskog koda koji ćete napisati služi da se prekine izvršavanje aplikacije kada korisnik klikne komandno dugme Quit. Da biste mogli da unesete programski kod, treba da otvori- te događaj `Click` komandnog dugmeta `btnQuit`. Na vrhu prozora za unošenje programskog koda postoje dve kartice. Na aktivnoj kartici piše `ViewerForm.vb*`. Na ovoj kartici se nalazi prozor za unošenje programskog koda za formular koji je zapisan u datoteci `ViewerForm.vb`. Pored ove kartice je kartica na kojoj piše `ViewerForm.vb [Design]*`. Kliknite tu karticu da biste prešli u prozor za dizajniranje formulara. Ukoliko, kada kliknete ovu karticu, bude prikazana poruka o grešci, to znači da u programskom kodu koji ste uneli postoji greška, pa je potrebno da ispravite programski kod tako da bude identičan programskom kodu koji je prikazan na slici 1.13. Kada budete ispravili programski kod i uspešni da prikazete prozor za dizajniranje formulara, dva puta kliknite komandno dugme Quit da biste otvorili događaj `Click` tog komandnog dugmeta.

Sledeći programski kod unesite u obradu događaja `Click` komandnog dugmeta `Quit` i na kraju svakog programskog reda kliknite taster `Enter`:

```
' Close the window and exit the application
```

```
Me.Close()
```

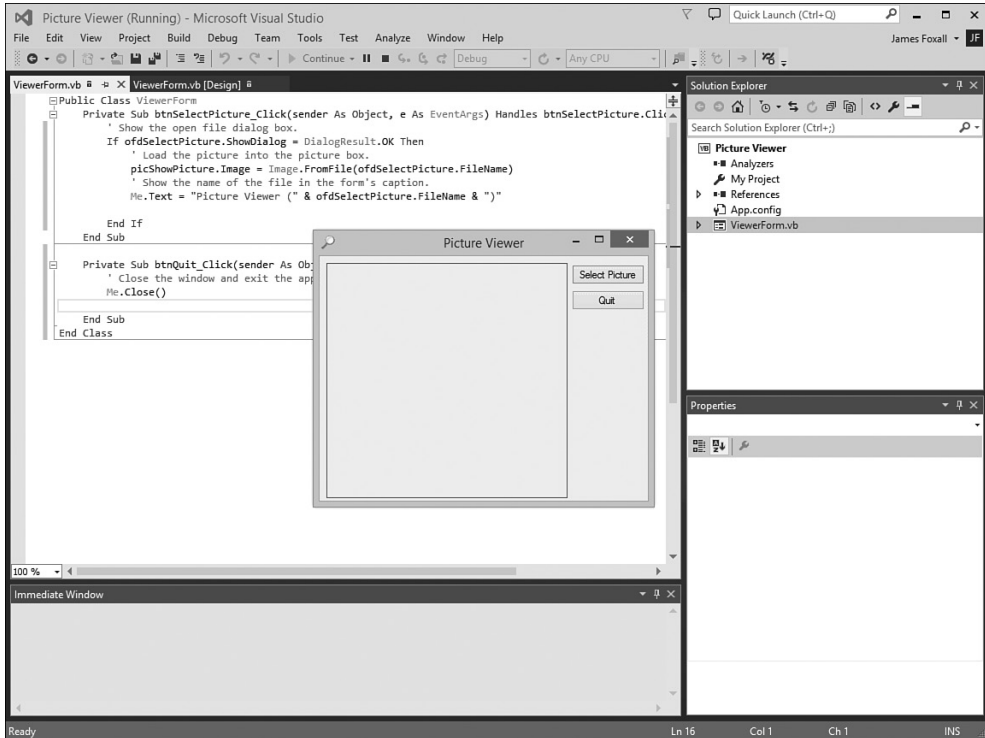
UZGRED BUDI REČENO

Pomoću programskog reda `Me.Close()` zatvarate aktivni formular. Kada zatvorite poslednji učítani formular programa, prekidate izvršavanje aplikacije. U toku izrade robusnijih aplikacija, verovatno ćete obavljati raznorazne procedure pre nego što prekinete izvršavanje aplikacije, međutim, u ovom primeru je dovoljno da zatvorite poslednji učítani formular.

Pokretanje i izvršavanje projekta

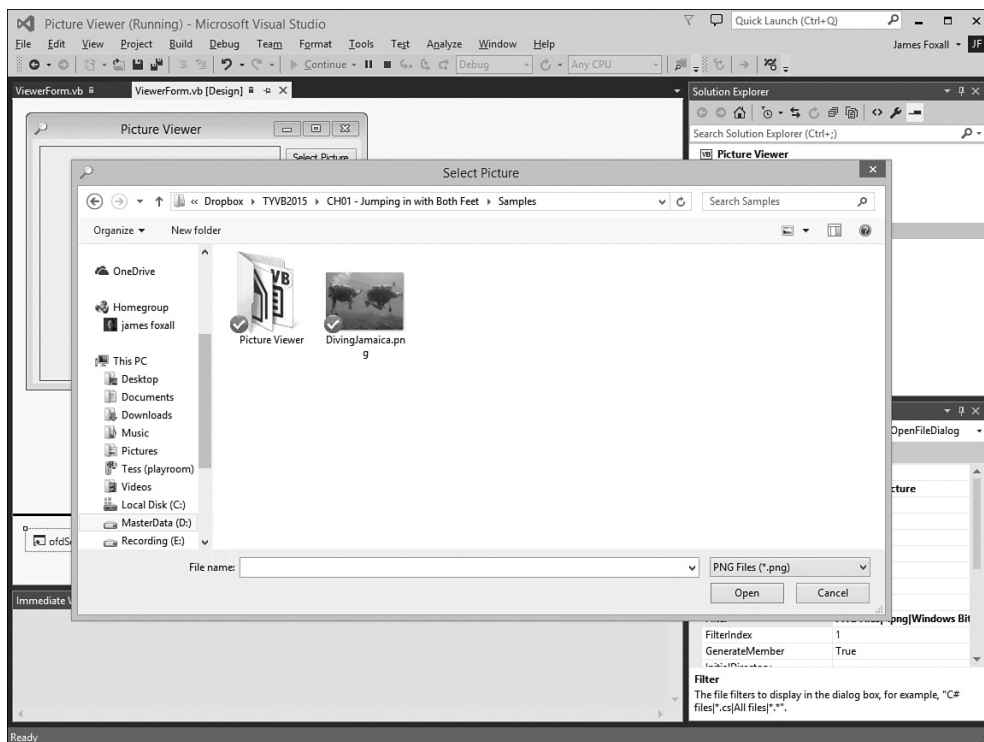
Aplikacija je sada potpuna. Na paleti alatki kliknite komandno dugme `Save All` (komandno dugme koje je predstavljeno sličicom sa dva flopi diska), a zatim pokrenite program tako što ćete pritisnuti taster `F5`. Takođe, program možete pokrenuti tako što ćete na paleti alatki kliknuti komandno dugme sa sličicom trougla okrenutim udesno koja podseća na taster `Play` na DVD plejeru i čiji potpis je `Start`. (Isto komandno dugme se nalazi i u meniju `Debug`.) Ukoliko naučite tastaturne prečice, proces pravljenja aplikacija će biti mnogo brži, te vam preporučujem da ih naučite.

Kada pokrenete program, promeniće se interfejs `Visual Basic`a, a formular koji ste napravili će biti prikazan iznad okruženja za pravljenje aplikacije (pogledajte sliku 1.14).

**SLIKA 1.14**

U režimu Run program se izvršava baš k bi se izvršavao kada ga pokrene krajnji korisnik.

Sada izvršavate program isto onako kako bi ga izvršavao korisnik na svom računaru. Vidite baš ono što bi korisnik video kada bi izvršavao program (samo bez razvojnog okruženja Visual Studio 2015 u pozadini, naravno). Kliknite komandno dugme Select Picture da biste otvorili okvir za dijalog Select Picture (pogledajte sliku 1.15). Koristeći taj okvir za dijalog pronađite datoteku koja sadrži sliku. Kada budete pronašli datoteku, kliknite je dva puta ili je kliknite jednom i potom kliknite komandno dugme Open. Slika koju ste odabrali biće prikazana u kontroli za prikazivanje slika (pogledajte sliku 1.16).



SLIKA 1.15

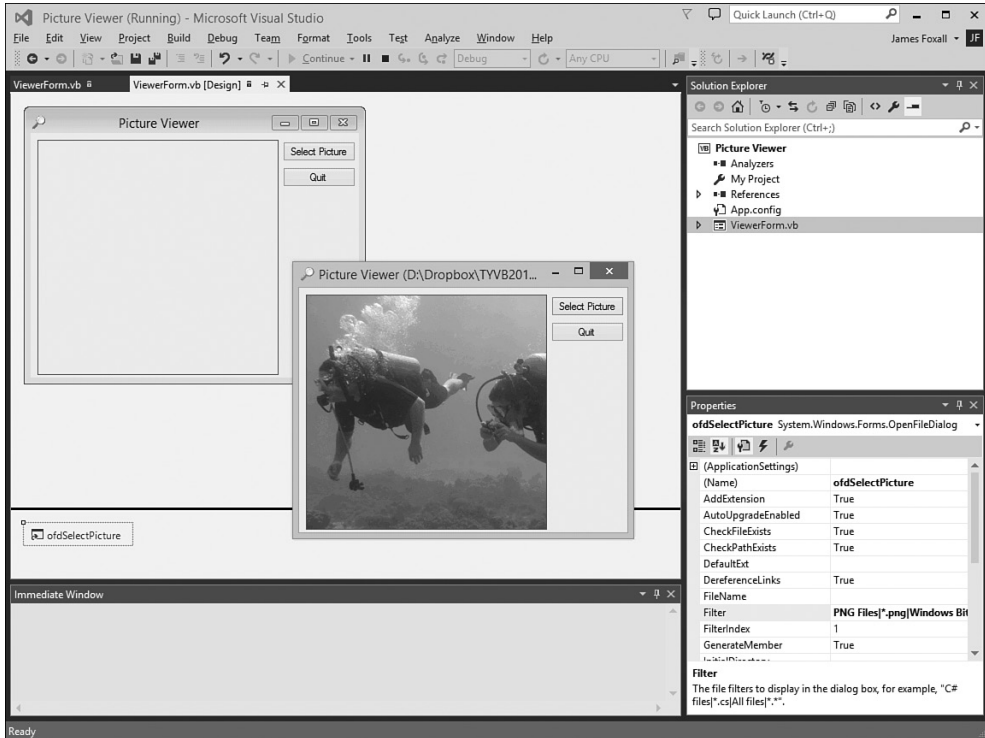
Kontrola OpenFileDialog omogućava da pronađete datoteku. Zgodno, zar ne?

UZGRED BUDI REČENO

Kada kliknete komandno dugme Select Picture, podrazumevana putanja koja je prikazana zavisi od poslednje aktivne putanje koja je korišćena u operativnom sistemu Windows, što znači da na vašem računaru može biti prikazna putanja koja se razlikuje od putanje koju vidite na slici 1.15.

UZGRED BUDI REČENO

Ukoliko želite da odaberete i prikazete sliku koju ste fotografisali koristeći digitalnu kameru, format datoteke je verovatno JPEG, što znači da u spisku Files of Type treba da odaberete odgovarajuću opciju. Takođe, ukoliko je slika koju želite da prikazete veoma velika, biće prikazan samo gornji levi ugao slike (deo koji može da stane u okvir za prikazivanje slike). U narednim lekcijama ćete naučiti kako da celu sliku, menjanjem razmere, prikazete u kontroli za prikazivanje slika, pa čak i da promenite veličinu formulara kako biste velike slike u potpunosti prikazali. Pošto isprobate program, kliknite komandno dugme Quit da biste se vratili u pogled Design.



SLIKA 1.16

Prikazivanje slika je jednostavno ukoliko znate neke tehnike.

Rezime

To je sve! Napravili ste Visual Basic program. Prozor Toolbox ste koristili kako biste napravili korisnički interfejs koji će korisnik koristiti za interakciju sa programom, a za važne obrade događaja ste napisali programski kod kako bi pomoću programa mogli da obavite razne operacije. Sve to su osnove pravljenja programa u Visual Basicu. Čak se i najsloženiji programi prave na ovaj način: pravi se korisnički interfejs i potom piše programski kod pomoću kojeg se obavljaju razne operacije. Naravno, pisanje programskog koda pomoću kojeg se obavljaju operacije *baš* onako kako ste zamislili predstavlja izazov, a mi smo spremni za takav izazov.

Ukoliko pažljivo pogledate kako su osmišljene lekcije, videćete da sam odlučio da vas prvo upoznam sa okruženjem Visual Basic (Visual Studio .NET). Zatim ću vas naučiti kako da napravite korisnički interfejs, a potom i kako da pišete programski kod. Takav sled sam namerno odabrao. Možda ste nestrpljivi pa želite odmah da pišete „ozbiljan“ programski kod, ali treba da znate da je pisanje programskog koda samo jedan deo u pravljenju aplikacija – ne zaboravite da u nazivu Visual Basic postoji reč *Visual*. Čitajući lekcije steći ćete solidne sposobnosti za pravljenje aplikacija.

Pitanja i odgovori

Pitanje – Mogu li da prikažem bitmape (slike) drugih tipova, a ne samo BMP, JPG i PNG?

Odgovor – Da. Pomoću kontrole `PictureBox` možete prikazati slike tipa BMP, JPG, ICO, EMF, WMF, PNG i GIF. Pomoću kontrole `PictureBox` možete i da zapišete slike u datoteke u navedenim formatima.

Pitanje – Da li je moguće da i u drugim kontrolama prikažem slike?

Odgovor – Kada želite da prikažete slike, onda koristite kontrolu `PictureBox`. Mađutim, mnoge druge kontrole vam omogućavaju da prikažete slike kao deo kontrole. Na primer, na kontroli možete da prikažete sliku tako što ćete svojstvu `Image` kontrole zadati sliku odgovarajuće tipa.

Radionica

Test

1. Kakav tip Visual Basic projekta pravite kada odaberete opciju za standardni Windows program?
2. Koji prozor koristite da biste promenili attribute (poziciju, veličinu i tako dalje) formulara ili kontrole u IDE-u?
3. Kako ćete otvoriti podrazumevanu obradu događaja za kontrolu?
4. Kom svojstvu kontrole zadajete vrednost da biste mogli da prikažete sliku?
5. Koja obrada događaja je podrazumevana za komandno dugme?

Odgovori

1. Windows Forms Application
2. Prozor Properties
3. Dva puta kliknite kontrolu u prozoru za dizajniranje
4. Svojstvo `Image`
5. Događaj `Click`

Vežbanja

1. Izmenite program Picture Viewer tako da korisnik može da pronađe i odabere GIF datoteke. (Savet: Treba da izmenite svojstvo `Filter` kontrole `OpenFileDialog`.)
2. Napravite nov projekat. Na formular u novom projektu postavite dva komandna dugmeta, jedno iznad drugog. Zatim, komandnu dugmad postavite jedno pored drugog.

