



IZDVOJENA ISKUSTVA ZAJEDNICE

C# 6 i .NET Core 1.0

Moderno međuplatformsko programiranje

Kreirajte moćne međuplatformske aplikacije, koristeći C# 6, .NET Core 1.0, ASP.NET Core 1.0 i Visual Studio 2015

Mark J. Price

 **kompjuter
biblioteka**

[PACKT]
PUBLISHING

IZDVOJENA ISKUSTVA ZAJEDNICE

C# 6 i .NET Core 1.0

Moderno međuplatformsko programiranje

Mark J. Price



 kompjuter
biblioteka

[PACKT]
PUBLISHING
BIRMINGHAM - MUMBAI

Izdavač:



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: Mark J. Price

Prevod: Slavica Prudkov

Lektura: Miloš Jevtović

Slog : Zvonko Aleksić

Znak Kompjuter biblioteke:

Miloš Milosavljević

Štampa: „Pekograf“, Zemun

Tiraž: 500

Godina izdanja: 2016.

Broj knjige: 484

Izdanje: Prvo

ISBN: 978-86-7310-507-9

C# 6 and .NET Core 1.0

by Mark J. Price

ISBN 978-1-78528-569-1

Copyright © 2016 Packt Publishing

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 „Packt Publishing“, Copyright © 2015.

Sva prava zadržana. Nije dozvoljeno da nijedan deo ove knjige bude reprodukovano ili snimljen 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 „Packt Publishing“ 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



Kratak sadržaj

POGLAVLJE 1

Zdravo C#!, dobrodošao .NET Core! 1

POGLAVLJE 2

Govoriti C# jezikom..... 43

Kontrolisanje toka, konvertovanje vrsta i rukovanje izuzecima 77

POGLAVLJE 3

Upotreba uobičajenih .NET vrsta 107

POGLAVLJE 4

Upotreba specijalizovanih .NET vrsta 133

POGLAVLJE 5

**Kreiranje sopstvenih vrsta pomoću objektno-orijentisanog
programiranja 165**

POGLAVLJE 6

Implementiranje interfejsa i nasleđivanje klasa..... 195

POGLAVLJE 7

Upotreba relacionih podataka pomoću Entity Frameworka 221

POGLAVLJE 8

Slanje upita i manipulisanje podacima pomoću LINQ-a 261

POGLAVLJE 9**Upotreba fajlova, nizova i serijalizacije 283****POGLAVLJE 10****Zaštita podataka i aplikacija 309****POGLAVLJE 11****Poboljšanje performanse i skalabilnosti korišćenjem
višeprogramskog rada 329****POGLAVLJE 12****Kreiranje Universal Windows Platform aplikacije pomoću XAML-a . 349****POGLAVLJE 13****Kreiranje web aplikacija i servisa pomoću ASP.NET Corea 371****POGLAVLJE 14****Upotreba C#-a na različitim platformama 409****POGLAVLJE 15****Izgradnja kviz aplikacije 437****POGLAVLJE 16****Odgovori na pitanja iz odeljka „Testirajte svoje znanje“ 471****DODATAK A****Kreiranje virtuelne mašine za razvojno okruženje 489****DODATAK B****INDEKS 507**



Sadržaj

POGLAVLJE 1

Zdravo C#!, dobrodošao .NET Core!	1
Podešavanje razvojnog okruženja	2
Upotreba alternativnih IDE-ova za C#	2
Upotreba Visual Studioa 2015 na Windowsu 10	3
Instaliranje Microsoft Visual Studioa 2015	3
Biranje standardne instalacije	4
Biranje prilagođene instalacije	4
Završavanje instalacije	6
Prijava u Visual Studio	6
Biranje razvojnih podešavanja	7
Ažuriranje ekstenzija i proizvoda	8
Upotreba starijih verzija Visual Studioa	10
Razumevanje .NET Frameworka, .NET Corea i .NET Nativea	10
Razumevanje .NET Framework platforme	10
Razumevanje Mono projekta	10
Razumevanje .NET Core platforme	11
Modernizacija .NET-a	11
Budućnost .NET-a	12
Razumevanje .NET Native platforme	12
Upoređivanje .NET tehnologija	13
Pisanje i prevođenje koda pomoću Developer Command Prompta	13
Pisanje koda pomoću Notepad-a	13
Prevođenje koda pomoću	
Developer Command Prompta	15
Ispravljanje grešaka kompajlera	16
Dekompajliranje koda pomoću ILDASM-a	17
Razdvajanje prevedenih sklopova	17
Pisanje i prevođenje koda pomoću Microsoft Visual Studioa 2015	19
Pisanje koda pomoću Visual Studioa 2015	19
Prevođenje koda pomoću Visual Studioa	22
Ispravljanje grešaka pomoću liste grešaka	23

Eksperimentisanje sa C# Interactiveom	23
Ostali korisni prozori	25
Ciljanje .NET Corea	26
Razumevanje .NET Portability Analyzera	26
Instaliranje .NET Portability Analyzera	26
Konfigurisanje .NET Portability Analyzera	27
Analiziranje rešenja	28
.NET Portability i ova knjiga	29
Kreiranje novih projekata za .NET Core	29
Upravljanje .NET Core programiranjem pomoću alatki komandne linije	29
Instaliranje alatki .NET komandne linije.....	30
Kreiranje .NET Core aplikacije pomoću alatki komandne linije	30
Kreiranje .NET Core aplikacije pomoću Visual Studioa 2015	33
Upravljanje izvornim kodom pomoću GitHuba	36
Upotreba prozora Team Explorer	36
Kloniranje GitHub skladišta	37
Upravljanje GitHub skladištem	38
Vežbanje i istraživanje	39
Vežba 1.1 – Testirajte svoje znanje	39
Vežba 1.2 – Vežbajte upravljanje porzorima Visual Studioa	39
Vežba 1.3 – Vežbajte kodiranje bilo gde	40
Vežba 1.4 – Istražite teme	40
Rezime	41

POGLAVLJE 2

Govoriti C# jezikom..... 43

Razumevanje osnova C# jezika	44
Gramatika C# jezika	47
Iskazi	47
Blokovi	48
C# rečnik	49
Pisanje koda	50
Glagoli su metodi	51
Imenice su vrste, polja i promenljive	52
Brojanje vrsta i metoda	52
Deklarisanje promenljivih	54
Imenovanje promenljivih	54
Čuvanje teksta	54
Čuvanje brojeva	55
Čuvanje celih brojeva	55
Čuvanje realnih brojeva	56
Čuvanje Bulovih vrednosti	59
Vrsta object.....	60
Vrsta dynamic.....	60
Lokalne promenljive	60

Izvođenje vrste lokalne promenljive	61
Kreiranje vrste vrednosti koje prihvataju null	62
Čuvanje više vrednosti u višedimenzionalnoj promenljivoj	62
Kreiranje aplikacija konzole	63
Prikazivanje ispisa za korisnika	63
Dobijanje unosa od korisnika	64
Importovanje imenskih prostora	64
Pojednostavljivanje upotrebe konzole u C#-u 6	65
Upotreba operatora u promenljivim.....	70
Eksperimentisanje sa unarnim operatorima	71
Eksperimentisanje sa aritmetičkim operatorima	72
Upoređivanje i Bulovi operatori	73
Vežbanje i istraživanje	73
Vežba 2.1 – Testirajte svoje znanje.....	73
Vežba 2.2 – Vežbajte veličine brojeva i raspone	73
Vežba 2.3 – Istražite teme	74
Rezime	75

POGLAVLJE 3

Kontrolisanje toka, konvertovanje vrsta i rukovanje izuzecima 77

Iskazi za selekciju	77
Iskaz if-else	78
Iskaz switch-case	79
Automatsko formatiranje koda	80
Iskazi ponavljanja	81
Iskaz while	81
Iskaz do-while	82
Iskaz for	82
Iskaz foreach	82
Kako funkcioniše iskaz foreach?	83
Pomeranje i konvertovanje	
između vrsta	84
Pomeranje od brojeva do brojeva	84
Upotreba vrste Convert	85
Zakruživanje brojeva	86
Konvertovanje iz bilo koje vrste u niz	86
Raščlanjavanje iz nizova u brojeve	
ili datume i vremena	87
Rukovanje izuzecima	89
Iskaz try-catch	89
Presretanje svih izuzetaka	90
Presretanje specifičnih izuzetaka	90
Iskaz finally	92
Pojednostavljenje odlaganja pomoću iskaza using	93
Provera prekoračenja	94
Iskaz checked	94
Iskaz unchecked	96
Potražite pomoć	97

MSDN	97
Dobijanje definicije koda	97
StackOverflow	99
Google	100
Obrazac dizajna	102
Vežbanje i istraživanje	102
Vežba 3.1 – Testirajte svoje znanje	102
Vežba 3.2 – Istražite petlje i prekoračenja	102
Vežba 3.3 – Vežbajte petlje i operatore	103
Vežba 3.4 – Vežbajte rukovanje izuzecima.....	104
Vežba 3.5 – Istražite teme	104
Rezime	105

POGLAVLJE 4

Upotreba uobičajenih .NET vrsta 107

Upotreba sklopova i imenskih prostora	107
Upoređivanje .NET Frameworka sa .NET Coreom	108
Osnovne biblioteke klasa i CoreFX	108
Sklopovi	108
Imenski prostori	108
Referenciranje sklopa	109
Referenciranje Microsoft Core biblioteke	109
Primer reference sklopa	109
Povezivanje sklopova i imenskih prostora	110
Importovanje imenskog prostora	111
Pretraživanje sklopova i imenskih prostora	113
Povezivanje C# ključnih reči sa .NET vrstama	115
Čuvanje teksta i manipulisanje tekstom	117
Dobijanje dužine niza	117
Dobijanje karaktera niza.....	117
Razdvajanje niza	118
Ekstrahovanje dela niza	118
Provera sadržaja niza	118
Ostali članovi niza	119
Efikasno kreiranje nizova	119
Validacija unosa pomoću regularnih izraza	120
Sintaksa regularnog izraza	121
Primeri regularnih izraza	122
Čuvanje podataka pomoću kolekcija	122
Razumevanje kolekcija	123
Liste	123
Rečnici	124
Grupe	124
Redovi	124
Setovi	125
Upotreba lista	125
Upotreba rečnika	126
Sortiranje kolekcija	127

Izbegavajte stare kolekcije	128
Upotreba specijalizovanih kolekcija	128
Vežbanje i istraživanje	129
Vežba 4.1 – Testirajte svoje znanje	129
Vežba 4.2 – Vežbajte regularne izraze	129
Vežba 4.3 – Istražite teme	130
Rezime	131

POGLAVLJE 5

Upotreba specijalizovanih

.NET vrsta 133

Ispravljanje grešaka i dijagnostika	133
Ispravljanje greške u aplikaciji	133
Podešavanje tačke prekida	134
Linija sa alatkama za ispravljanje grešaka	135
Prozori za ispravljanje grešaka	135
Pregled koda	136
Prilagođavanje tačaka prekida	138
Praćenje performanse i upotrebe resursa	139
Procena efikasnosti vrsta	139
Praćenje performanse i upotrebe memorije	140
Merenje efikasnosti obrade nizova	142
Kontrola pomoću klasa Debug i Trace	144
Pisanje u standardni prijemnik praćenja	144
Konfigurisanje prijemnika praćenja	145
Konfigurisanje simbola kompajlera za .NET Framework	147
Definisanje simbola kompajlera za .NET Core	148
Provera simbola kompajlera	149
Prebacivanje nivoa praćenja	150
Testiranje koda aplikacije	152
Kreiranje jedinice koda koja treba da se testira	152
Kreiranje projekta za testiranje koda	152
Pokretanje testova koda	154
Internacionalizacija aplikacije	155
Globalizovanje aplikacije	156
Lokalizovanje aplikacije	157
Interoperabilnost sa neupravljanim kodom	159
Automatizacija Microsoft Excela pomoću	
COM Interopa	160
Pristup Win32 API-u pomoću P/Invokea	161
Vežbanje i istraživanje	163
Vežba 5.1 – Testirajte svoje znanje	163
Vežba 5.2 – Vežbajte upotrebu metoda	
Debug i Trace	163
Vežba 5.3 – Istražite teme	163
Rezime	164

POGLAVLJE 6

Kreiranje sopstvenih vrsta pomoću objektno-orientisanog programiranja 165

Koncepti objektno-orientisanog programiranja (OOP-a)	165
Kreiranje biblioteka klasa	166
Kreiranje biblioteke klase za deljenje koda	166
Definisanje klase	167
Instanciranje klase	168
Čuvanje podataka pomoću polja	171
Definisanje polja	171
Čuvanje vrednosti pomoću ključne reči enum	173
Čuvanje više vrednosti pomoću kolekcija	175
Učinite polje statičnim	176
Učinite polje konstantnim	177
Učinite polja samo za čitanje (read-only).....	178
Pokretanje polja pomoću konstruktora	178
Pisanje i pozivanje metoda	179
Preklapanje metoda	181
Opcioni parametri i imenovani argumenti	181
Razdvajanje klasa korišćenjem metoda partial	183
Kontrolisanje pristupa pomoću parametara i indeksa	184
Definisanje read-only parametara	184
Definisanje podesivih parametara	185
Definisanje indeksa	186
Pojednostavljenje metoda	
pomoću operatora	188
Podizanje događaja i rukovanje njima	189
Pozivanje metoda pomoću delegata	189
Definisanje događaja	190
Vežbanje i istraživanje	192
Vežba 6.1 – Testirajte svoje znanje	192
Vežba 6.2 – Vežbajte pisanje matematičkih metoda	192
Vežba 6.3 – Istražite teme	193
Rezime	194

POGLAVLJE 7

Implementiranje interfejsa i nasleđivanje klasa..... 195

Implementiranje interfejsa	195
Uobičajeni interfejs	196
Upoređivanje objekata prilikom sortiranja	196
Definisanje posebnog poređivača	198
Upravljanje memorijom pomoću reference i vrste vrednosti	199
Definisanje vrste pomoću ključne reči struct	200
Opuštanje neupravljanih resursa	201
Obavezno pozivanje metoda Dispose	203
Nasleđivanje iz klasa	203
Proširivanje klasa	205

Skrivanje članova	205
Zamena vrednosti članova	206
Sprečavanje nasleđivanja i zamene vrednosti	207
Polimorfizam	208
Eksplicitno konvertovanje unutar hijerarhije nasleđivanja	209
Implicitno konvertovanje	209
Eksplicitno konvertovanje	209
Rukovanje izuzecima eksplicitnog konvertovanja	210
Dokumentovanje vrsta	211
Nasleđivanje i proširenje .NET vrsta	213
Nasleđivanje iz klase Exception	214
Proširenje vrsta kada se ne možete nasleđivati	215
Upotreba statičnih metoda za ponovnu upotrebu funkcija	215
Upotreba proširenih metoda za ponovnu upotrebu funkcija	216
Vežbanje i istraživanje	217
Vežba 7.1 – Testirajte svoje znanje	218
Vežba 7.2 – Vežbajte kreiranje hijerarhije nasleđivanja	218
Vežba 7.3 – Istražite teme	218
Rezime	219

POGLAVLJE 8

Upotreba relacionih podataka pomoću Entity Frameworka 221

Sistemi za upravljanje relacionim bazama podataka	221
Povezivanje na Microsoft SQL Server - LocalDb	221
Northwind uzorak baze podataka	222
Biranje dobavljača .NET podataka	226
Upotreba ADO.NET-a	226
Povezivanje sa bazom podataka	227
228	
Izvršavanje komandi i čitanje setova rezultata	228
Učitavanje niza konekcije iz konfiguracije	231
232	
Upotreba Entity Frameworka 6	232
Kreiranje Entity Framework 6 modela	233
Entity Framework Code First modeli	237
EF Code First konvencija	237
EF atributi napomena	237
EF Fluent API	238
Evidentiranje SQL iskaza	240
241	
Manipulisanje podacima pomoću	
Entity Data Modela	243
Ubacivanje objekata	243
Ažuriranje objekata	244
Transakcije	245
Definisanje eksplicitne transakcije	246
Šabloni učitavanja u EF-u	247

Odloženo učitavanje objekata	247
Poželjno učitavanje objekata	250
Eksplcitno učitavanje objekata	251
Upotreba Entity Framework Corea	254
Vežbanje i istraživanje	258
Vežba 8.1 – Testirajte svoje znanje	258
Vežba 8.2 – Istražite EF Core dokumentaciju	258
Vežba 8.3 – Istražite teme	259
Rezime	259

POGLAVLJE 9

Slanje upita i manipulisanje

podacima pomoću LINQ-a 261

Pisanje LINQ upita	261
Produženje sekvenci pomoću klase Enumerable	262
Filtriranje objekata pomoću metoda Where	262
Upotreba imenovanog metoda	263
Pojednostavljenje koda uklanjanjem eksplicitnog instanciranja delegata	264
Upotreba lambda izraza	264
Sortiranje sekvenci pomoću metoda OrderBy	264
Upotreba setova	266
Projektovanje objekata pomoću iskaza Select	267
Spajanje i grupisanje	270
„Zaslađivanje“ sintakse pomoću sintaksnog „slatkiša“	272
Upotreba više programskih niti pomoću Parallel LINQ-a	274
Kreiranje sopstvenih proširenih	
LINQ metoda	276
Upotreba LINQ to XML-a	278
Generisanje XML-a pomoću LINQ to XML-a	278
Čitanje XML-a upotrebom LINQ to XML-a	279
Vežbanje i istraživanje	279
Vežba 9.1 – Testirajte svoje znanje	280
Vežba 9.2 – Vežbajte slanje upita pomoću LINQ-a	280
Vežba 9.3 – Istražite teme	281
Rezime	281

POGLAVLJE 10

Upotreba fajlova, nizova i serijalizacije 283

Upravljanje fajl sistemom	283
Upravljanje direktorijumima	283
Upravljanje putanjama	285
Dobijanje informacija o fajlu	286
Čitanje i pisanje pomoću tokova	286
Pisanje u tekstualne i XML tokove	288
Kompresovanje tokova	290
Kodiranje teksta	291
Kodiranje nizova kao višedimenzionalnih promenljivih bajtova	292

Kodiranje i dekodiranje teksta u fajlovima	295
Serijalizacija grafikona objekta	295
Serijalizacija pomoću XML-a	296
Deserijalizacija pomoću XML-a	299
Serijalizacija pomoću JSON-a	300
Serijalizacija pomoću drugih formata	301
Serijalizacija pomoću serijalizera pri pokretanju	302
Vežbanje i istraživanje	304
Vežba 10.1 – Testirajte svoje znanje	304
Vežba 10.2 – Vežbajte serijalizaciju kao XML	305
Vežba 10.3 – Istražite formate serijalizacije	305
Vežba 10.4 – Istražite „Microsoftove“ System.IO vrste	306
Vežba 10.5 – Istražite teme	307
Rezime	307

POGLAVLJE 11

Zaštita podataka i aplikacija 309

Razumevanje rečnika zaštite	310
Ključevi i njihove veličine	310
Inicijalizacioni vektori i veličine bloka	311
Salt vrednost	311
Generisanje ključeva i inicijalizacionih vektora	311
Enkripcija i dekripcija podataka	312
Simetrična enkripcija pomoću AES-a	312
Podaci heširanja	315
Heširanje pomoću algoritma SHA256	316
Potpisivanje podataka	319
Potpisivanje pomoću algoritama SHA256 i RSA	319
Provera identiteta i autorizacija korisnika	321
Upravljanje lokalnim nalogima	321
Provera identiteta pomoću Windowsa	322
Autorizacija pomoću Windowsa	324
Vežbanje i istraživanje	326
Vežba 11.1 – Testirajte svoje znanje	326
Vežba 11.2 – Vežbajte zaštitu podataka pomoću enkripcije i heširanja	327
Vežba 11.3 – Vežbajte zaštitu podataka pomoću dekripcije	327
Vežba 11.4 – Istražite teme	327
Rezime	328

POGLAVLJE 12

Poboljšanje performanse i skalabilnosti korišćenjem višeprogramskog rada..... 329

Razumevanje procesa i programskih niti	329
Asinhrono pokretanje zadataka	330
Sinhronizovano pokretanje više akcija	330
Asinhrono pokretanje više akcija pomoću zadataka	331

Čekanje zadataka	333
Nastavak sa drugim zadatkom	334
Ključne reči <code>async</code> i <code>await</code>	336
Kreiranje GUI-ja koji blokira	336
Kreiranje GUI-ja koji ne blokira	339
Ostale vrste sa metodima <code>Async</code>	340
Ključna reč <code>await</code> u blokovima <code>catch</code>	341
Poboljšanje skalabilnosti za aplikacije klijent-server	341
Sinhronizovanje pristupa deljenim resursima	341
Pristupanje resursima iz više programskih niti	342
Primena međusobnog ekskluzivnog zaključavanja na resurse	344
Razumevanje iskaza <code>lock</code>	344
Kreiranje operacije tako da budu atomske	346
Primena drugih vrsta sinhronizacije	346
Vežbanje i istraživanje	347
Vežba 12.1 – Testirajte svoje znanje	347
Vežba 12.2 – Istražite teme	347
Rezime	348

POGLAVLJE 13

Kreiranje Universal Windows Platform aplikacije pomoću XAML-a . 349

Razumevanje UWP-a	350
Prilagođavanje rasporeda aplikacije	350
Upotreba jedinstvenih mogućnosti uređaja	350
Razumevanje XAML-a	350
Pojednostavljenje koda pomoću XAML-a	351
Kreiranje Windows desktop aplikacije pomoću WPF-a	351
Biranje uobičajenih kontrola	354
Kreiranje aplikacije za Universal Windows platformu	355
Analiza prenosivosti UWP-a	358
Upotreba resursa i šablona	358
Deljenje resursa	358
Zamena kontrolnog šablona	359
Povezivanje podataka	360
Povezivanje sa elementima	360
Povezivanje sa podacima	361
Animiranje pomoću scenarija	363
Testiranje u emulatorima	368
Vežbanje i istraživanje	370
Vežba 13.1 – Testirajte svoje znanje	370
Vežba 13.2 – Vežbajte izgradnju univerzalnog tip kalkulatora	370
Vežba 13.3 – Istražite teme	370
Rezime	370

POGLAVLJE 14

Kreiranje web aplikacija i servisa pomoću ASP.NET Corea 371

Razumevanje ASP.NET Corea	372
Klasični ASP.NET, nasuprot modernog ASP.NET Corea	373
Web programiranje na strani klijenta	373
Instaliranje Web Essentialsa 2015	374
HyperText Transfer Protocol (HTTP)	374
Kreiranje projekta web aplikacije	376
Istraživanje delova ASP.NET Core web aplikacije	380
Pokretanje ASP.NET Corea	381
Razumevanje standardnog usmeravanja	382
Kontroleri ASP.NET Corea	383
Definisanje akcija kontrolera Home	383
Modeli ASP.NET Corea	384
Kreiranje Entity modela za Northwind	385
Konfigurisanje Entity Framework Corea kao servisa	385
Kreiranje modela prikaza za zahteve	388
Dodavanje modela u kontroler	388
ASP.NET Core prikazi	389
Renderovanje prikaza Home kontrolera	389
Deljenje rasporeda između prikaza	390
Definisanje sopstvenih stilova	391
Definisanje unetog prikaza	392
Dalji pregled ASP.NET Corea	394
Prosleđivanje parametara pomoću vrednosti usmeravanja.....	394
Prosleđivanje parametara pomoću niza upita	396
Označavanje modela	399
ASP.NET Core Web API	400
Postavljanje API kontrolera	401
Pozivanje Web API servisa iz UWP aplikacije	402
Vežbanje i istraživanje	406
Vežba 14.1 – Testirajte svoje znanje	406
Vežba 14.2 – Vežbajte izgradnju web aplikacije koju pokreću podaci	406
Vežba 14.3 – Istražite teme	406
Rezime	407

POGLAVLJE 15

Upotreba C#-a na različitim platformama 409

Razumevanje međuplatformskog programiranja	410
Visual Studio Code	410
.NET Core 1.0	410
ASP.NET Core 1.0	410
Entity Framework Core 1.0	410
NET Native	411
Xamarin	411
Universal Windows platforma (UWP) na različitim uređajima	411
Instaliranje Visual Studio Codea i ASP.NET Corea	412

Instaliranje Visual Studio Codea	412
Upravljanje razvojem iz terminala	413
Instaliranje Homebrewa	414
Instaliranje ASP.NET Corea i .NET Version Managera	415
Prebacivanje izvršenja sa Mono na .NET Core	416
Izgradnja aplikacije konzole	417
Kreiranje izvornih fajlova	417
Editovanje koda	417
Preuzimanje paketa zavisnosti i kompajliranje	420
Kreiranje međuplatformskih web aplikacija pomoću ASP.NET Corea	420
Kreiranje najjednostavnije ASP.NET Core web aplikacije	420
Instaliranje Yeoman i sličnih alati	425
Postavljanje projekata pomoću Yeomana	426
Editovanje projekata pomoću Visual Studio Codea	429
Razumevanje alati komandne linije .NET Corea	431
Razumevanje DNX alati	431
Razumevanje CLI alati	431
Uobičajene DNX komande	431
Uobičajene CLI komande	432
Vežbanje i istraživanje	434
Vežba 15.1 – Testirajte svoje znanje	434
Vežba 15.2 – Vežbajte prenos postojeće ASP.NET aplikacije	434
Vežba 15.3 – Istražite teme	435
Rezime	435

POGLAVLJE 16

Izgradnja kviz aplikacije 437

Dizajniranje kviz aplikacije	437
Biranje platforme	437
Odlučivanje o funkcionalnim zahtevima	438
Izdvajanje problema	438
Izgradnja kviz rešenja	438
Definisanje modela objekata	439
Kreiranje skladišta podataka	440
Kreiranje web aplikacije	441
Definisanje modela prikaza	442
Dodavanje uzorka kviz pitanja	443
Konfigurisanje stanja sesije	447
Dodavanje prilagođenih akcija kontrolera	447
Dodavanje prilagođenih prikaza	452
Pokretanje kviza	456
Konfigurisanje projekta za korišćenje Kestrel web servera i .NET Corea	456
Hostovanje ASP.NET Core web aplikacija na Microsoft Azureu	460
Registracija Azure naloga	460
Kreiranje Azure web aplikacije	461
Publikovanje ASP.NET web aplikacije na Web Appu	464

Vežbanje i istraživanje	468
Vežba 16.1 – Testirajte svoje znanje	468
Vežba 16.2 – Vežbajte proširenje kviz web aplikacije	468
Vežba 16.3 – Istraživanje tema	468
Rezime	469

DODATAK A

Odgovori na pitanja iz odeljka „Testirajte svoje znanje“ 471

Poglavlje 1 - Zdravo C#!, dobrodošao .NET Core!	471
Poglavlje 2 - Govoriti C# jezikom	472
Poglavlje 3 - Kontrolisanje toka, konvertovanje vrsta i rukovanje izuzecima	473
Vežba 3.2.....	474
Poglavlje 4 - Upotreba uobičajenih .NET vrsta	475
Poglavlje 5 - Upotreba specijalizovanih .NET vrsta	476
Poglavlje 6 - Kreiranje sopstvenih vrsta pomoću objektno-orijentisanog programiranja	477
Poglavlje 7 - Implementiranje interfejsa i nasleđivanje klasa	479
Poglavlje 8 - Upotreba relacionih podataka pomoću Entity Frameworka	480
Poglavlje 9 - Slanje upita i manipulisanje podacima pomoću LINQ-a	481
Poglavlje 10 - Upotreba fajlova, nizova i serijalizacije	482
Poglavlje 11 - Zaštita podataka i aplikacija	484
Poglavlje 12 - Poboljšanje performanse i skalabilnosti korišćenjem višeprogramskog rada	484
Poglavlje 13 - Kreiranje Universal Windows Platform aplikacije pomoću XAML-a ...	485
Poglavlje 14 - Kreiranje web aplikacija i servisa pomoću ASP.NET Corea	485
Poglavlje 15 - Upotreba C#-a na različitim platformama	487
Poglavlje 16 - Izgradnja kviz aplikacije	487

DODATAK B

Kreiranje virtuelne mašine za razvojno okruženje 489

Prijava na Microsoft nalog	489
Kreiranje Microsoft Azure prijave	490
Upravljanje Microsoft Azure nalogom	493
Kreiranje virtuelne mašine	494
Povezivanje na virtuelnu mašinu pomoću udaljenog desktopa	499
Podržavanje drugih platformi	504
Rezime	505

INDEKS 507



UVOD

Postoji mnoštvo knjiga za C# (neke od njih imaju i više od hiljadu strana) koje „preten-
duju“ da budu sveobuhvatne reference za C# programski jezik i .NET Framework.

Ova knjiga je drugačija. Ona je koncizna - cilj mi je da bude jednostavna za čitanje i da ima praktične vodiče. Napisao sam ovu knjigu da bude najbolji vodič, korak-po-korak, za modernu međuplatformsku C# i .NET praksu.

Istaći ću sve interesantne delove C#-a da biste mogli da impresionirate svoje kolege i zaposlene i da brzo postanete produktivni. Da vam ne bih dosadivao i detaljno objašnjavao sve sitnice, pretpostavljam da, ako je neki termin koji koristim za vas nov, znate kako da ga potražite u pretraživaču, kao što je DuckDuckGo.

Na kraju svakog poglavlja postoji odeljak „Vežbanje i istraživanje“, koji sadrži pitanja pomoću kojih ćete moći da testirate svoje znanje. Uglavnom postoji i praktična vežba, a istražićete teme detaljno sami, uz moju malu pomoć, tako što ću vas usmeriti u odgovarajućem pravcu.

Možete da preuzmete rešenja za vežbe iz skladišta GitHub na adresi <https://github.com/markjprice/cs6dotnetcore>. Obezbediću vam i instrukcije kako da to uradite, koristeći Visual Studio 2015 na kraju Poglavlja 1, „Zdravo C#!, dobrodošao .NET Core!“.

ŠTA OBUHVATA OVA KNJIGA?

U Poglavlju 1, „Zdravo C#!, dobrodošao .NET Core!“, biće reči o podešavanju razvojnog okruženja i upotrebi različitih alata za kreiranje najjednostavnijih mogućih aplikacija pomoću C#-a. Naučićete kako da sastavite C# u komandnoj liniji i kako da napišete i prevedete kod, koristeći Visual Studio. Takođe ćete naučiti više o različitim .NET platformama, kao što su .NET Framework, .NET Core i .NET Native.

Poglavlje 2, „Govoriti C#“, posvećeno je C# jeziku, gramatici i rečniku koji ćete koristiti svakog dana za pisanje izvornog koda za vaše aplikacije. Konkretno, naučićete kako da deklarirate i upotrebite različite vrste promenljivih.

U Poglavlju 3, „Kontrolisanje toka, konvertovanje vrsta i rukovanje izuzecima“, tema je pisanje koda koji odlučuje, ponavlja blokove iskaza, konvertuje vrste i rukuje greškama. Takođe ćete naučiti koja su najbolja mesta na kojima možete zatražiti pomoć.

U Poglavlju 4, „*Upotreba uobičajenih .NET vrsta*“, saznaćete kako se vrste .NET-a odnose prema C#-u. Upoznaćete .NET Framework, .NET Core i njihove biblioteke sklopova vrsta, koje omogućavaju aplikacijama da međusobno povežu postojeće komponente za izvršavanje uobičajenih zadataka.

Poglavlje 5, „*Upotreba specijalizovanih .NET vrsta*“, posvećeno je vrstama .NET-a koje se koriste za dijagnostiku problema, podržavanje više jezika i kultura i pristup funkcijama i aplikacijama van .NET-a.

U Poglavlju 6, „*Kreiranje sopstvenih vrsta pomoću objektno-orijentisanog programiranja*“, upoznaćete različite kategorije članova koje mogu imati vrste, uključujući polja za čuvanje podataka i metode za izvršavanje akcija. Koristićete OOP koncepte, kao što su agregacija i kapsuliranje.

Poglavlje 7, „*Implementiranje interfejsa i nasleđivanje klasa*“, posvećeno je isporučivanju novih vrsta iz postojećih pomoću OOP-a. Naučićete kako da implementirate interfejs, šta su osnovne i izvedene klase, kako da prepisete član vrste, kako da upotrebite polimorfizam i kako da se postavite između klasa u hijerarhiji nasleđivanja.

Temu Poglavlja 8, „*Upotreba relacionih podataka pomoću Entity Frameworka*“, predstavljaju čitanje i pisanje na Microsoft SQL Server (i druge baze podataka) pomoću klasičnog ADO.NET-a i objektno-relacione tehnologije mapiranja, koja je poznata kao Entity Framework.

U Poglavlju 9, „*Slanje upita i manipulisanje podacima pomoću LINQ-a*“, biće reči o LINQ-u (Language Integrated Queries) – ekstenzijama jezika koje dodaju mogućnost upotrebe sekvenci stavki, filtriranja i sortiranja i projektuju stavke u različite ispile.

U Poglavlju 10, „*Upotreba fajlova, nizova i serijalizacije*“, upoznaćete čitanje i pisanje u fajlove i nizove, šifriranje teksta i serijalizaciju.

Poglavlje 11, „*Zaštita podataka i aplikacija*“, posvećeno je zaštiti podataka pomoću enkripcije i heširanja i proveravanju ko pokreće aplikaciju i šta mu je dozvoljeno da uradi.

U Poglavlju 12, „*Poboljšanje performanse i skalabilnosti korišćenjem višeprogramskog rada*“, saznaćete kako se omogućava izvršavanje više akcija istovremeno za poboljšanje performanse, skalabilnosti i produktivnosti korisnika.

U Poglavlju 13, „*Kreiranje Universal Windows Platform aplikacije pomoću XAML-a*“, upoznaćete osnove XAML-a, koje možete da upotrebite za definisanje korisničkog interfejsa za grafičke aplikacije za Universal Windows Platform (UWP). Ova aplikacija može da se pokrene nakon kreiranja i na Windowsu 10, Windowsu 10 Mobile, Xbox Oneu, pa, čak, i na HoloLensu.

Poglavlje 14, „*Kreiranje web aplikacija i servisa pomoću ASP.NET Core-a*“, posvećeno je izgradnji web aplikacija i servisa pomoću moderne HTTP arhitekture na strani servera korišćenjem Microsoft ASP.NET Core-a 1.0. Naučićete više o modelima, prikazima i kontrolama koje čine MVC i Web API.

U Poglavlju 15, „*Upotreba C#-a na različitim platformama*“, predstavljen je način upotrebe C#-a na različitim platformama kada se koriste .NET Core, ASP.NET Core 1.0, Entity Framework Core 1.0 i Visual Studio Code.

U Poglavlju 16, „*Izgradnja aplikacije za testiranje*“, biće reči o dizajniranju i izgradnji aplikacije za testiranje koja će učenicima pomoći da nauče C#, .NET Core i slične teme.

Dodatak A, „*Odgovori na pitanja 'Testirajte svoje znanje'*“, sadrži odgovore na pitanja testa koja se nalaze na kraju svakog poglavlja.

U Dodatku B, „*Kreiranje virtuelne mašine za razvojno okruženje*“, prikazano je kako se podešava virtuelna mašina u Microsoft Azureu da bi bila upotrebljena kao razvojno okruženje.

ŠTA VAM JE POTREBNO ZA OVU KNJIGU?

Iako možete da programirate i koristite C# na mnogim platformama, uključujući i Mac OS X i Linux, za učenje je najbolje da upotrebite sledeća dva softvera:

- Microsoft Windows 10
- Microsoft Visual Studio Community 2015 sa Update 1 (ili noviji)

Najbolja verzija Windowsa koju možete da upotrebite je Microsoft Windows 10, zato što je ona potrebna da biste kreirali Universal Windows Platform aplikacije u Poglavlju 13, „*Kreiranje Universal Windows Platform aplikacije pomoću XAML-a*“. Starije verzije Windowsa, kao što su 7 ili 8.1, biće dovoljne za sva ostala poglavlja. Ako nemate Windows računar, možete da upotrebite virtuelnu mašinu na kojoj je pokrenut Windows u cloudu. Za više detalja pogledajte Dodatak B, „*Kreiranje virtuelne mašine za razvojno okruženje*“.

Najbolja verzija Visual Studioa koju možete da upotrebite je Microsoft Visual Studio Community 2015 sa Update 1 (ili noviji). Ja sam nju koristio prilikom pisanja ove knjige. U Poglavlju 15, „*Upotreba C#-a na različitim platformama*“, predstaviću Visual Studio Code, koji se pokreće na Windows, Linux i Mac OS X sistemu, a možete ga upotrebiti za kreiranje međuplatformskih aplikacija za sve ove operativne sisteme.

ZA KOGA JE OVA KNJIGA?

Ako već znate da je C# popularan osnovni, međuplatformski programski jezik koji se koristi za kreiranje svega, od poslovnih aplikacija, web sajtova i servisa, do igara za mobilne uređaje, Xbox One i Windows 10 (za desktop, tablet i telefon), onda je ova knjiga za vas.

KONVENCIJE

U ovoj knjizi pronaći ćete više različitih stilova za tekst koje sam upotrebio za različite vrste informacija. Evo nekih primera ovih stilova i objašnjenja njihovog značenja.

Reči koda u tekstu, nazivi tabela baze podataka, nazivi direktorijuma, nazivi fajlova, ekstenzije fajla, nazivi putanja, kratki URL-ovi, korisnički unos i Twitter statusi su prikazani na sledeći način: „*Kod koji ćete napisati će biti sačuvan u fajl pod nazivom Program.cs*“.

Blok koda je postavljen na sledeći način:

```
// the best nieces and nephews in the world
names[0] = "Kate";
names[1] = "Jack";
names[2] = "Rebecca";
names[3] = "Tom";
```

Kada treba privući vašu pažnju na određeni deo bloka koda, relevantne linije ili stavke će biti ispisane masnim slovima:

```
// the best nieces and nephews in the world
names[0] = "Kate";
names[1] = "Jack";
names[2] = "Rebecca";
names[3] = "Tom";
```

Svi unosi ili ispisi komandne linije napisani su na sledeći način:

```
csc my.cs /target:library
```

Novi termini i važne reči su napisani masnim slovima. Reči koje vidite na ekranu (na primer, u menijima ili okvirima za dijalog) biće prikazane u tekstu na sledeći način: „U **Visual Studio**u iz menija **View** izaberite **Other Windows**, a zatim **C# Interactive**“.



Upozorenja ili važne napomene su prikazani u ovakvom okviru.



Saveti i trikovi su prikazani ovako.

POVRATNE INFORMACIJE ČITALACA

Povratne informacije od naših čitalaca su uvek dobrodošle. Obavestite nas šta mislite o ovoj knjizi – šta vam se dopalo ili šta vam se možda nije dopalo. Povratne informacije su nam važne da bismo ubuduće kreirali naslove od kojih ćete dobiti maksimum.

Da biste nam poslali povratne informacije, jednostavno nam pošaljite e-mail na adresu feedback@packtpub.com; u naslovu poruke napišite naslov knjige.

Ako postoji tema za koju ste specijalizovani ili ste zainteresovani da pišete ili sarađujete na nekoj od knjiga, pogledajte vodič za autore na adresi www.packtpub.com/authors.

KORISNIČKA PODRŠKA

Sada, kada ste ponosni vlasnik „Packt knjige“, mi imamo mnogo štošta da vam ponudimo da bismo vam pomogli da dobijete maksimum iz svoje narudžbine.

Preuzimanje primera koda

Možete da preuzmete fajlove sa primerima koda za ovu knjigu sa vašeg naloga na adresi <http://www.packtpub.com>. Ako ste ovu knjigu kupili na drugom mestu, možete da posetite stranicu <http://www.packtpub.com/support> i da se registrujete da biste fajlove dobili e-mailom.

Možete da preuzmete fajlove sa primerima koda, prateći sledeće korake:

1. Prijavite se ili registrujte na našem web sajtu, koristeći svoju e-mail adresu i lozinku.
2. Postavite kursor na karticu **SUPPORT** na vrhu stranice.
3. Kliknite na **Code Downloads & Errata**.
4. U polje **Search** unesite naslov knjige.
5. Izaberite knjigu za koju želite da preuzmete fajlove sa primerima koda.
6. Iz padajućeg menija izaberite mesto na kojem ste kupili knjigu.
7. Kliknite na **Code Download**.

Kada su fajlovi preuzeti, ekstrahirajte direktorijum, koristeći najnoviju verziju:

- ▣ WinRAR / 7-Zip za Windows
- ▣ Zipeg / iZip / UnRarX za Mac
- ▣ 7-Zip / PeaZip za Linux

Štamparske greške

Iako smo preduzeli sve mere da bismo obezbedili tačnost sadržaja, moguće su greške. Ako pronađete grešku u nekoj od naših knjiga (možda u tekstu ili u kodu), bićemo vam zahvalni ako nam to prijavite. Na taj način ćete da poštediti druge čitaoce od frustracija, a nama ćete da pomognete da poboljšamo naredne verzije ove knjige. Ako pronađete neku štamparsku grešku, molimo vas da nas obavestite, tako što ćete posetiti stranicu <http://www.packtpub.com/submit-errata>, selektovati knjigu, kliknuti na link Errata Submission Form i uneti detalje o grešci koju ste pronašli. Kada je greška verifikovana, vaša prijava će biti prihvaćena i greška će biti aploudovana na naš web sajt ili dodata u listu postojećih grešaka, pod odeljkom „Errata“ za određeni naslov.

Da biste pregledali prethodno prijavljene greške, posetite stranicu <https://www.packtpub.com/books/content/support> i unesite naslov knjige u polje za pretragu. Tražena informacija će biti prikazana u odeljku „Errata“.

Piraterija

Piraterija autorskog materijala na Internetu je aktuelan problem na svim medijima. Mi u „Packtu“ zaštitu autorskih prava i licenci shvatamo veoma ozbiljno. Ako pronađete ilegalnu kopiju naših knjiga, u bilo kojoj formi, na Internetu, molimo vas da nas o tome obavestite i da nam pošaljete adresu lokacije ili naziv web sajta da bismo mogli da podnesemo tužbu.

Molimo vas, obratite nam se na adresi copyright@packtpub.com i pošaljite nam link ka sumnjivom materijalu.

Bićemo vam zahvalni za pomoć u zaštiti naših autora i mogućnost da vam pružimo vredan sadržaj.

Pitanja

Možete da sa nama kontaktirate na adresi questions@packtpub.com ako imate bilo kakvih problema sa bilo kojim aspektom knjige - mi ćemo dati sve od sebe da te probleme rešimo.



1

Zdravo C#!, dobrodošao .NET Core!

Ovo poglavlje posvećeno je podešavanju razvojnog okruženja, razumevanju sličnosti i razlika između .NET Corea, .NET Frameworka i .NET Nativea i upotrebi različitih alati za kreiranje najjednostavnijih mogućih aplikacija pomoću C#-a.

Većina korisnika uči komplikovane teme imitacijom i ponavljanjem, umesto da pročita detaljno objašnjenje teorije. Dakle, neću objašnjavati svaku ključnu reč i svaki korak. Cilj je da vas naučim da napišete kod i izgradite i pokrenete aplikaciju. Još ne treba da znate detalje kako sve to funkcioniše.

Kao što je rekao za sebe Samuel Johnson, autor knjige „English dictionary of 1755“, verovatno sam i ja napravio nekoliko kardinalnih grešaka i napisao nekoliko apsurdnih izraza, od čega ni jedan posao ovakvog opsega nije imun. Preuzimam potpunu odgovornost za konkretne greške i nadam se da ćete ceniti moj trud uloženi pri pisanju knjige o .NET Coreu 1.0 tokom njegovog „rađanja“ početkom 2016. godine.

Ovo poglavlje obuhvata sledeće teme:

- ▣ podešavanje razvojnog okruženja
- ▣ razumevanje .NET Frameworka, .NET Corea i .NET Nativea
- ▣ pisanje i prevođenje koda pomoću Developer Command Prompta
- ▣ pisanje i prevođenje koda pomoću Microsoft Visual Studioa 2015
- ▣ ciljanje .NET Corea
- ▣ upravljanje izvornim kodom pomoću GitHuba

PODEŠAVANJE RAZVOJNOG OKRUŽENJA

Pre nego što započnete da programirate, potrebno je da podesite **Interactive Development Environment (IDE)**, koji uključuje editor koda za C#. Najbolji IDE koji možete da izaberete je Microsoft Visual Studio 2015, ali se on pokreće samo na Windows operativnom sistemu.

Upotreba alternativnih IDE-ova za C#

Postoje i alternativni IDE-ovi za C# - na primer, **MonoDevelop**, **JetBrains Project Rider** i **Microsoft Visual Studio Code**. Svaki od njih ima dostupne verzije za Windows, Linux i OS X operativne sisteme, omogućavajući da se kod piše na jednom i upotrebi na istom ili drugom operativnom sistemu:

- ▣ za MonoDevelop IDE posetite stranicu <http://www.monodevelop.com/>
- ▣ za JetBrains Project Rider posetite stranicu <http://blog.jetbrains.com/dotnet/2016/01/13/project-rider-a-csharp-ide/>
- ▣ za Microsoft Visual Studio Code posetite stranicu <https://code.visualstudio.com/>.

Cloud9 je IDE zasnovan na pretraživaču, pa je, stoga, najprilagođeniji za upotrebu na različitim platformama. Evo i linka: <https://c9.io/web/sign-up/free>.

U Poglavlju 15, „*Upotreba C#-a na različitim platformama*“, pokazaću kako se upotrebljava Visual Studio Code koji se pokreće na Mac OS X operativnom sistemu za kreiranje ASP.NET Core 1.0 web aplikacije, koja može da se upotrebi na Windows, Mac OS X ili Linux operativnom sistemu, ili za Docker kontejnere.



Linux i Docker su popularne platforme za hostovanje servera, zato što su relativno mali i jeftiniji za proširenje kada se uporede sa platformama operativnih sistema koji su dizajnirani za krajnje korisnike, kao što su Windows i Mac OS X.

Upotreba Visual Studioa 2015 na Windowsu 10

Možete da upotrebite Windows 7 ili noviju verziju da biste izvršili većinu zadataka iz poglavlja ove knjige, ali bilo bi najbolje da koristite Windows 10.



Ako nemate Windows, preporučujem da kreirate virtuelnu mašinu (VM), koju ćete upotrebiti za programiranje. Možete da izaberete bilo kojeg cloud provajdera, ali Microsoft Azure ima unapred konfigurisane virtuelne mašine koje uključuju pravilno licenciranje Windowsa i Visual Studioa 2015. Potrebno je da se plati za samo onoliko utrošenih minuta koliko je pokrenuta virtuelna mašina, pa je to dobar način za korisnike Linux, Mac OS X i starijih Windows operativnih sistema da iskoriste sve prednosti upotrebe Visual Studioa 2015. Za više informacija pogledajte Dodatak B, „*Kreiranje virtuelne mašine za razvojno okruženje*“

U oktobru 2014. godine „Microsoft“ je napravio izdanje profesionalnog kvaliteta Visual Studioa koje je dostupno za svakog korisnika besplatno. Ovo izdanje se zove **Community Edition**.



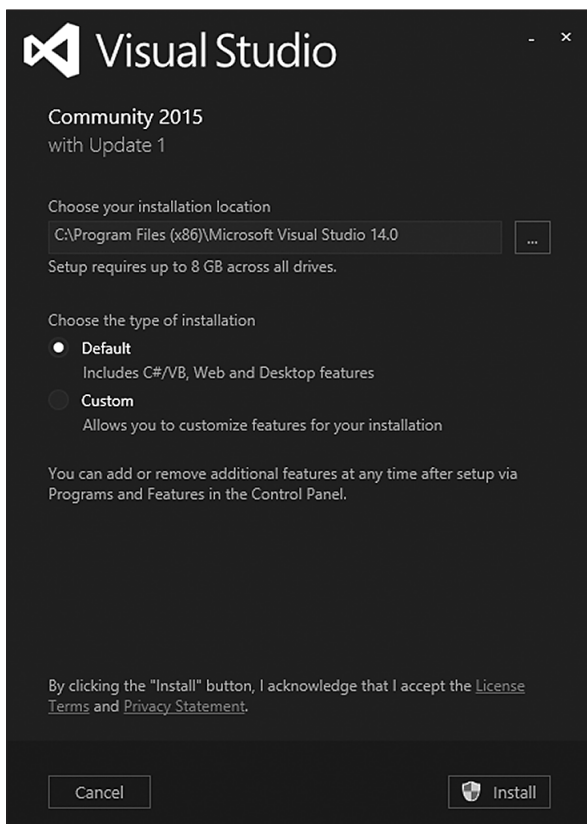
„Microsoft“ je ukombinovao sve besplatne programerske alate u program pod nazivom Visual Studio Dev Essentials. Ovaj paket uključuje Community Edition, besplatan nivo Visual Studio Team Servicesa, Azure kredite za testiranje i razvoj i besplatan trening u Pluralsightu, Wintellectu i Xamarinu.

Instaliranje Microsoft Visual Studioa 2015

Preuzmite i instalirajte Microsoft Visual Studio Community 2015 with Update 1 ili noviji.

Biranje standardne instalacije

Ako vam se više dopada manja i brža instalacija, izaberite Default, zato što ona zahteva samo 8 GB i treba da se završi za jedan sat. Ovim izborom, kao što je prikazano na sledećoj slici, moći ćete da izvršite zadatke iz dve trećine ove knjige. Da biste završili ostale zadatke iz knjige, možete kasnije da dodate funkcije, kao što je podrška za izgradnju Universal Windows Platform (UWP) aplikacija, koristeći **Programs and Features** u Control Panelu:



Ako izaberete opciju **Default**, pređite na odeljak „Završavanje instalacije“.

Biranje prilagođene instalacije

Ako imate 34 GB slobodnog prostora na hard disku i možete da sačekate nekoliko sati, u okviru za dijalog za instalaciju izaberite opciju Custom, a zatim kliknite na Next. Selektujte sledeće funkcije:

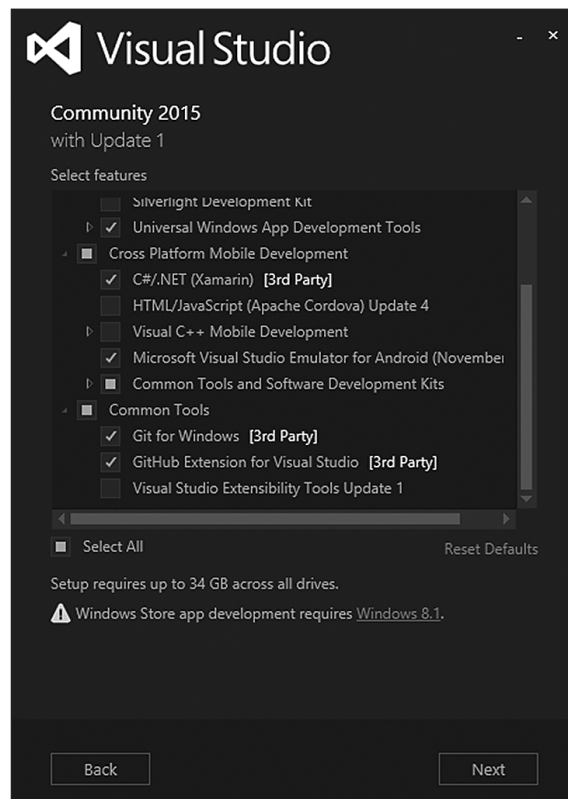
- U odeljku Windows and Web Development potvrdite sledeća polja za potvrđivanje:
 - Microsoft SQL Server Data Tools

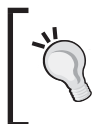
- Microsoft Web Developer Tools
- Universal Windows App Development Tools
- ▣ U odeljku Cross Platform Mobile Development potvrdite sledeće polje za potvrđivanje:
 - C#/.NET (Xamarin)
- ▣ U odeljku Common Tools potvrdite sledeća polja za potvrđivanje:
 - Git for Windows
 - GitHub Extension for Visual Studio

Kliknite na Next:



Ako imate Home izdanje Windowsa 10, nećete moći da instalirate emulator za Windows Phone, zato što oni zahtevaju Hyper-V podršku.

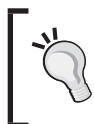




Možete da izaberete da instalirate sve ako želite podršku za jezike kao što su C++, Python i F#, ali oni neće biti opisani u ovoj knjizi.

Završavanje instalacije

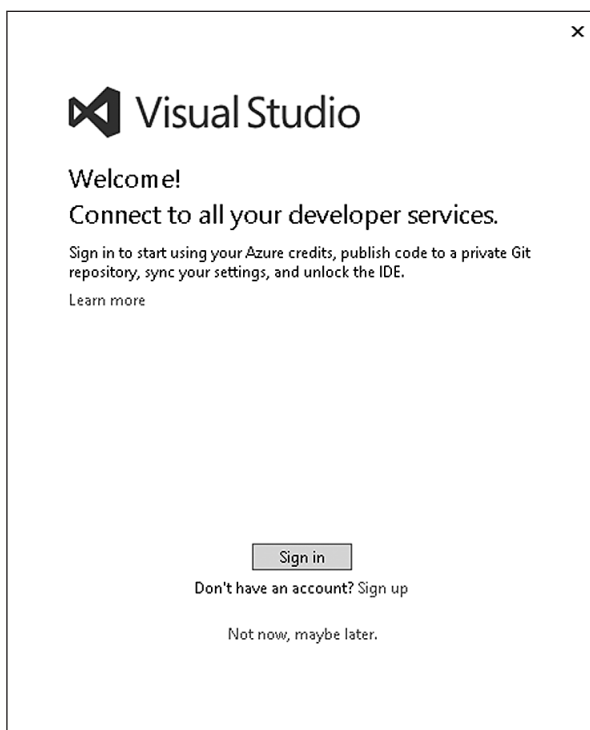
U okviru za dijalog za uslove licenciranja kliknite na **Install**. Sačekajte da fajlovi budu preuzeti i instalirani.



Dok čekate da se instalira Visual Studio, možete da pređete na odeljak „Razumevanje .NET Frameworka, .NET Corea i .NET Nativea“ u ovom poglavlju.

Prijava u Visual Studio

Kada prvi put pokrenete Visual Studio, bićete upozoreni da treba se prijavite:

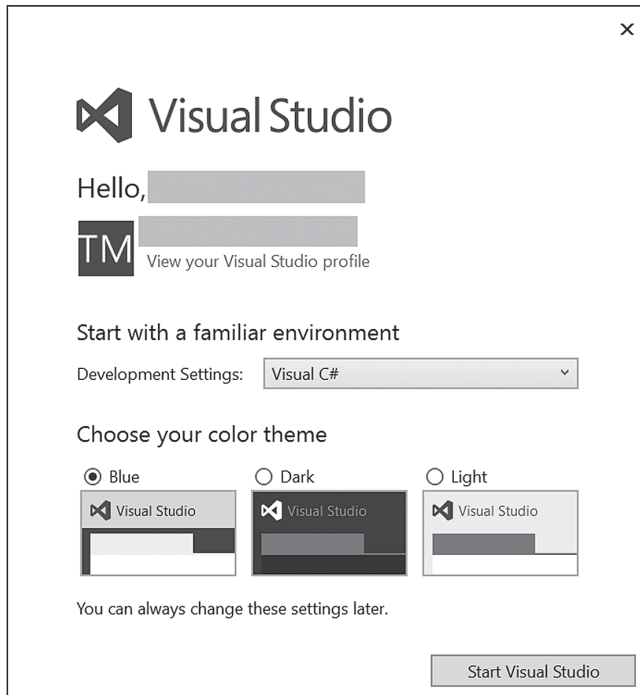


Ako imate Microsoft nalog, na primer, Hotmail, MSN, Live ili Outlook e-mail adresu, možete da ga upotrebite. Ako nemate, registrujte se za novi nalog na sledećem linku:

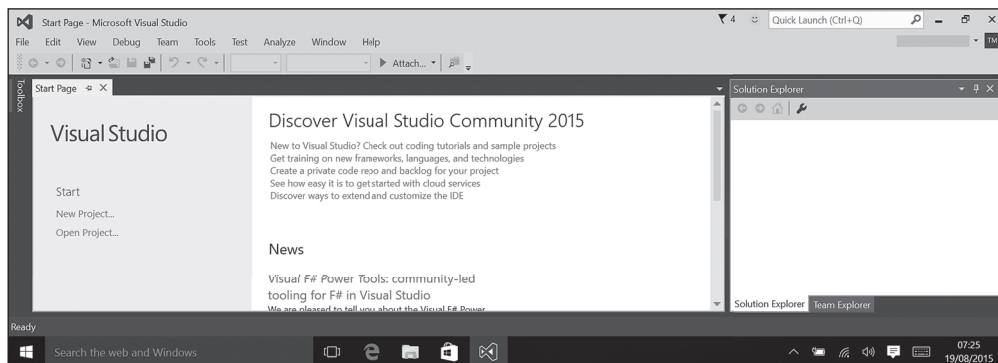
<https://signup.live.com/>

Biranje razvojnih podešavanja

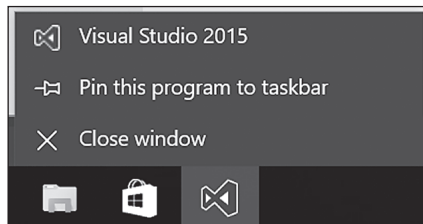
Nakon prijave na nalog, potrebno je da izaberete **Development Settings**. Izaberite Visual C# da biste konfigurisali prečice na tastaturi i standardni razmeštaj prozora koji najviše odgovara C# programerima, kao što je prikazano na sledećoj slici.



Nakon što kliknete na **Start Visual Studio**, videćete korisnički interfejs Visual Studioa sa otvorenom početnom stranicom u centralnom području. Kao i većina Windows desktop aplikacija, i Visual Studio ima liniju menija, liniju alatki za uobičajene komande i statusnu liniju na dnu prozora. Sa desne strane nalazi se Solution Explorer prozor, koji će izlistati sve otvorene projekte, kao što je prikazano na sledećoj slici.



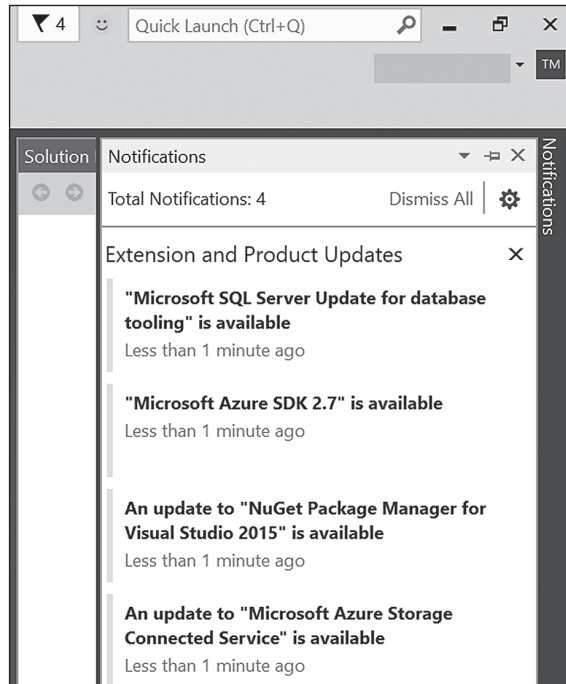
Da biste ubuduće imali brz pristup Visual Studiou, kliknite desnim tasterom miša na unos u Windows taskbaru i izaberite opciju **Pin this program to taskbar**:



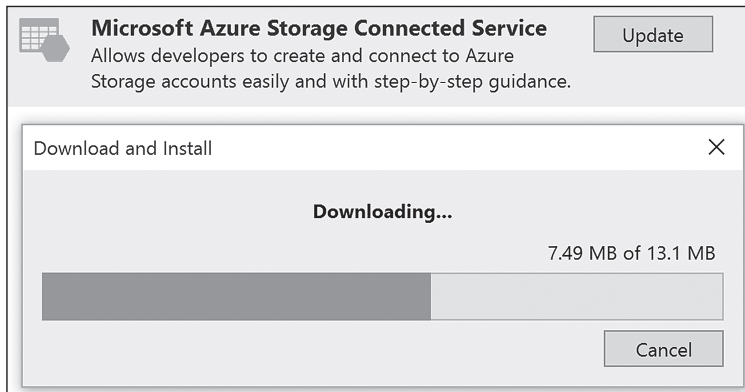
Ažuriranje ekstenzija i proizvoda

U gornjem desnom uglu prozora Visual Studioa videćete malu zastavicu pored koje se nalazi broj. To su obaveštenja o ekstenzijama i ažuriranjima proizvoda.

Kliknite na zastavicu da biste prikazali prozor B, kao što je prikazano na sledećoj slici.



U prozoru Notifications kliknite na svaki unos da biste instalirali poslednju verziju programa. Možete da instalirate ove verzije i kasnije, ako želite, ali je najbolje da program bude ažuran, jer ćete tako izbeći greške i probleme u zaštiti njegove bezbednosti. Na sledećoj slici prikazani su preuzimanje i instaliranje ekstenzije.



Upotreba starijih verzija Visual Studioa

Ako želite da upotrebite besplatnu verziju Visual Studioa izdatu pre 2015. godine, možete upotrebite jedno od više ograničenih Express izdanja. Većina kodova iz ove knjige će funkcionisati i na starijim verzijama ako imate na umu kada su sledeće funkcije izdate:

GODINA	C#	FUNKCIJE
2005	2	generičke komponente <T>
2008	3	lambda izrazi sa => i manipulisanje sekvencama pomoću LINQ-a (from, in, where, orderby, ascending, descending, select, group, into)
2010	4	dinamičko kucanje pomoću funkcije dynamic i višenitni rad pomoću Taska
2012	5	pojednostavljen višenitni rad pomoću funkcija async i await
2015	6	interpolacija niza pomoću \$ " ", importovanje statičnih vrsta pomoću funkcije using static i druga poboljšanja

RAZUMEVANJE .NET FRAMEWORKA, .NET COREA I .NET NATIVEA

.NET Framework, .NET Core i .NET Native su slične platforme koje programeri koriste za razvoj aplikacija i servisa.

Razumevanje .NET Framework platforme

„Microsoftova“ .NET Framework platforma je razvojna platforma koja uključuje Common Language Runtime (CLR), koji upravlja izvršenjem koda, i bogatu biblioteku klasa za izgradnju aplikacija.

„Microsoft“ je dizajnirao platformu .NET Framework da može da bude međuplatformska, ali se potrudio i da ona najbolje funkcioniše na Windows operativnom sistemu.

Praktično, .NET Framework je platforma samo za Windows operativne sisteme.

Razumevanje Mono projekta

Zajednica otvorenog koda je razvila međuplatformsku .NET implementaciju pod nazivom Mono projekat (<http://www.mono-project.com/>).

Mono je međuplatformski projekat, koji savršeno odgovara „Microsoftovoj“ implementaciji .NET Frameworka i odlična je osnova za **Xamarin** mobilnu platformu.

Razumevanje .NET Core platforme

Danas živimo u pravom međuplatformskom svetu. Usled modernog razvoja mobilnih i cloud platformi, Windows je postao daleko manje važan operativni sistem. Dakle, „Microsoft“ je vredno nastojao da razdvoji usku vezu .NET Frameworka sa Windows operativnim sistemom.

Dok je menjao .NET, tako da bude stvarno međuplatformski, „Microsoft“ je iskoristio mogućnost da ga preradi i ukloni glavne delove koji se više ne smatraju jezgrom.

Ovaj novi proizvod, koji je nazvan .NET Core 1.0, uključuje međuplatformsku implementaciju CLR-a, poznatijeg kao **CoreCLR**, i modernizovanu biblioteku klasa, koja je poznata kao **CoreFX**.

Modernizacija .NET-a

.NET Core 1.0 je mnogo manji od aktuelne verzije .NET Frameworka, zato što je iz ove platforme uklonjeno dosta sadržaja.

Na primer, Windows Forms i Windows Presentation Foundation (WPF) mogu da se upotrebe za izgradnju aplikacije grafičkog korisničkog interfejsa (GUI), ali su usko vezani za Windows, pa su, stoga, uklonjeni iz .NET Corea. Najnovija tehnologija za izgradnju Windows aplikacija je Universal Windows Platform (UWP), o kojoj ćete saznati više u Poglavlju 13, „*Kreiranje Universal Windows Platform aplikacije pomoću XAML-a*“.

ASP.NET Web Forms i Windows Communication Foundation (WCF) su stare tehnologije web aplikacija i servisa koje programeri sve manje koriste, pa su i one uklonjene iz .NET Corea. Umesto njih, programeri radije koriste ASP.NET MVC i ASP.NET Web API. Ove dve tehnologije su prerađene i ukombinovane u novi proizvod ASP.NET Core 1.0, koji se pokreće na .NET Coreu. Naučićete više o ovom proizvodu u Poglavlju 14, „*Kreiranje web aplikacija i servisa pomoću ASP.NET Corea*“.

Entity Framework (EF) 6.x je objektno-relaciona tehnologija mapiranja za rad sa podacima koji su sačuvani u relacionim bazama podataka, kao što su Oracle i Microsoft SQL Server. Tokom godina je ova tehnologija prilično proširena, pa je, stoga, međuplatformska verzija sada smanjena i nazvana Entity Framework Core 1.0. Naučićete više o ovoj tehnologiji u Poglavlju 8, „*Upotreba relacionih podataka pomoću Entity Frameworka*“.

Neke vrste podataka u .NET-u koje su uključene u .NET Framework i .NET Core su pojednostavljene uklanjanjem pojedinih članova. Na primer, u .NET Frameworku klasa `File` sadrži metod `Close` i `Dispose` i bilo koji metod može da se upotrebi za brisanje izvora fajla. U .NET Coreu postoji samo metod `Dispose`, što smanjuje utrošak memorije sklopa i pojednostavljuje API.



Veličina .NET Frameworka 4.6 iznosi oko 200 MB. .NET Core 1.0 je veliki oko 11 MB. Na kraju, .NET Core može da ima sličnu veličinu. Cilj „Micro-softa“ nije da .NET Core učini manjim od .NET Frameworka, već da on sadrži komponente koje podržavaju moderne tehnologije i da ima manje zavisnosti, tako da upotreba zahteva samo one komponente koje su aplikaciji stvarno potrebne.

Budućnost .NET-a

.NET Core je budućnost .NET-a. .NET Core 1.0 je odličan početak, ali će biti potrebne još jedna ili dve verzije da bi postao razvijen kao i aktuelna verzija .NET Frameworka.

U ovoj knjizi ćemo se fokusirati na dostupne funkcije u .NET Coreu, ali ćemo upotrebiti .NET Framework kada funkcije nisu (još) implementirane u .NET Core.



Jedan od razloga za izbor konkretne slike na koricama ove knjige je da biste mogli da zamislite .NET Core kao novo svitanje za .NET. .NET Framework je postao preveliki, kao gusta šuma, a .NET Core je kao pronalazač-nje proplanka unutar te šume.

Razumevanje .NET Native platforme

Još jedna .NET inicijativa je .NET Native, koji prevodi C# kod u izvorne CPU instrukcije pre vremena (AoT), umesto da koristi CLR za prevođenje IL-a tačno na vreme (JIT) na izvorni kod.

.NET Native prevodilac poboljšava brzinu izvršenja i smanjuje upotrebu memorije za aplikacije. Podržava sledeće:

- UWP aplikacije za Windows 10, Windows 10 Mobile, Xbox One, HoloLens i Internet of Things (IoT) uređaje, kao što je Raspberry Pi
- web razvoj na strani servera pomoću ASP.NET Core 1.0
- aplikacije za konzole za upotrebu u komandnoj liniji

Upoređivanje .NET tehnologija

U sledećoj tabeli rezimirane su i upoređene tri „Microsoftove“ .NET tehnologije:

PLATFORMA	SET FUNKCIJA	C# PREVODI NA	HOST OPERATIVNI SISTEMI
.NET Framework	razvijen i proširiv	Intermediate Language (IL) kod	samo Windows
.NET Core	nov i na neki način ograničen		Windows, Linux, Mac OS X, Docker
.NET Native	nov i veoma ograničen	izvorni kod	

PISANJE I PREVOĐENJE KODA POMOĆU DEVELOPER COMMAND PROMPTA

Kada instalirate Visual Studio, instalirane su i druge alatke. Jedna od tih alatki je **Developer Command Prompt for VS2015**, koji ima podešenu putanju za pronalaženje programerskih alatki, kao što je C# prevodilac.

Pisanje koda pomoću Notepada

Otvorite Notepad (ili vaš omiljeni editor za tekst) i unesite sledeći kod:

```
class Program { static void Main() { System.Console.WriteLine("Hello C#!"); } }
```

Možete da kucate ceo kod u jednoj liniji ili da ga rastavite u više linija i uvučete linije da biste olakšali čitanje.

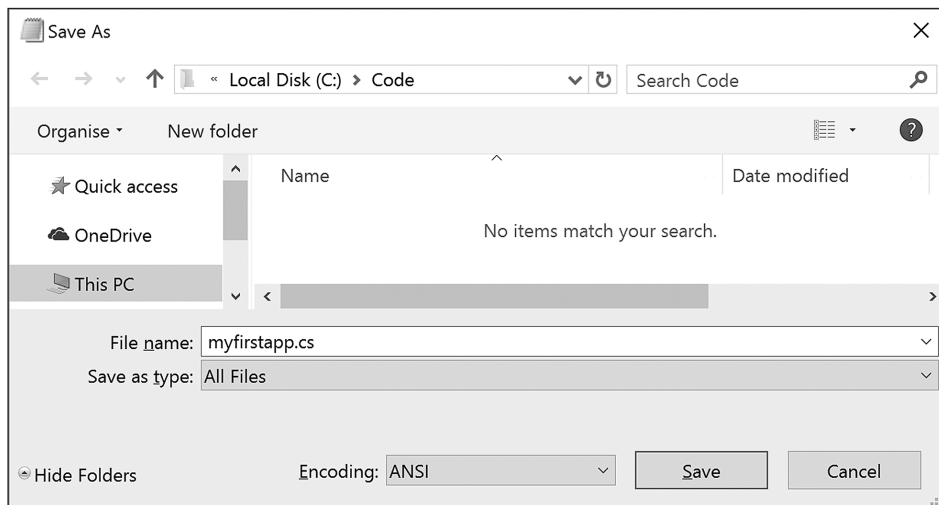


C# je osetljiv na veličinu slova, što znači da morate da kucate karaktere velikih slova i malih slova tačno kao što je prikazano u prethodnom kodu. Nije osetljiv na razmake, što znači da nije važno da li koristite tabulator i razmake, pa možete da razmestite kod kako god želite.

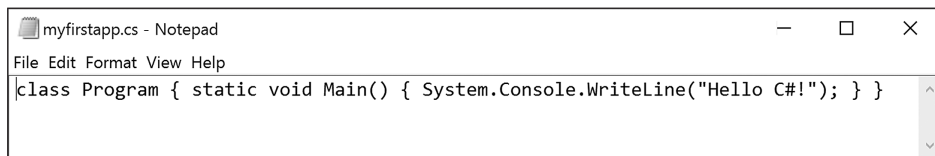
Iz menija **File** izaberite opciju **Save As**.

U okviru za dijalog promenite putanju na drajv C: (ili bilo koji drugi drajv koji želite da upotrebite za snimanje projekata), kliknite na dugme **New Folder** i dodelite direktorijumu naziv Code.

U polju **Save as type** izaberite **All Files** iz padajuće liste da biste izbegli dodavanje ekstenzije .txt i unesite naziv fajla `myfirstapp.cs`, kao što je prikazano na sledećoj slici:

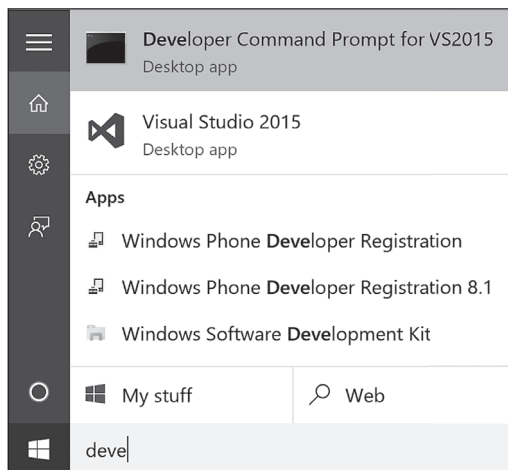


Kod u Notepadu treba da izgleda kao na sledećoj slici.



Prevođenje koda pomoću Developer Command Prompta

Pokrenite Developer Command Prompt for VS2015, tako što ćete ukucati slova **deve** u polje Search Windowsa 10, kao što možete da vidite na sledećoj slici. Takođe možete da ga pronađete u meniju **Start** ili **Start Screen** koji je izlistan u **All apps** u direktorijumu Visual Studio 2015:



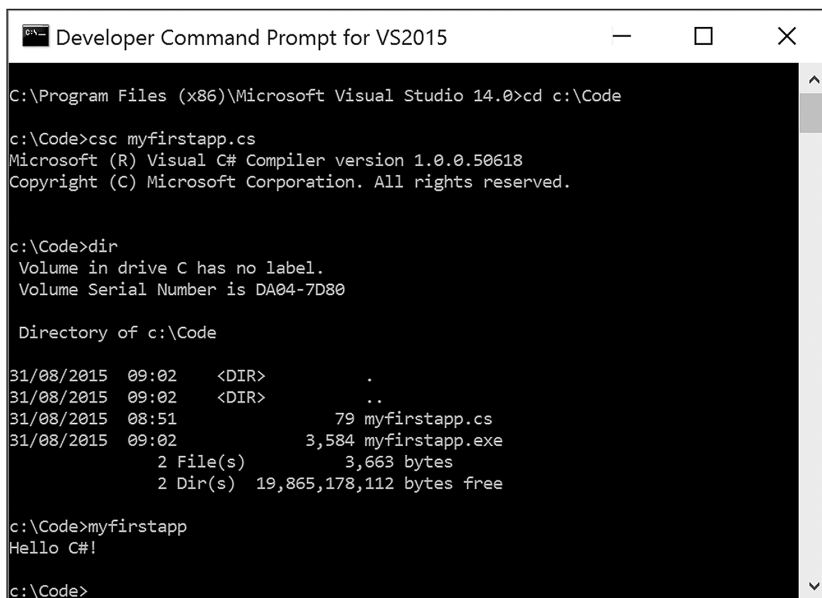
U Command Promptu unesite komande da biste:

- ▣ promenili direktorijum na **C:\Code**
- ▣ preveli izvorni kod, koristeći C# kompajler
- ▣ zatražili listu direktorijuma
- ▣ pokrenili aplikaciju unošenjem naziva EXE fajla

Ovo su komande:

```
cd C:\Code
csc myfirstapp.cs
dir
myfirstapp
```

Ispis u prozoru Command Prompt treba da izgleda ovako:



```
Developer Command Prompt for VS2015

c:\Program Files (x86)\Microsoft Visual Studio 14.0>cd c:\Code

c:\Code>csc myfirstapp.cs
Microsoft (R) Visual C# Compiler version 1.0.0.50618
Copyright (C) Microsoft Corporation. All rights reserved.

c:\Code>dir
Volume in drive C has no label.
Volume Serial Number is DA04-7D80

Directory of c:\Code

31/08/2015  09:02    <DIR>          .
31/08/2015  09:02    <DIR>          ..
31/08/2015  08:51                79 myfirstapp.cs
31/08/2015  09:02            3,584 myfirstapp.exe
                2 File(s)          3,663 bytes
                2 Dir(s)  19,865,178,112 bytes free

c:\Code>myfirstapp
Hello C#!

c:\Code>
```

Vidite da je izvorni kod fajl pod nazivom `myfirstapp.cs` preveden u sklop pod nazivom `myfirstapp.exe`. Kada unesete naziv sklopa, .Net Framework 4.6 i njegov CLR će učitati i izvršiti sklop.

Možete da kopirate fajl `myfirstapp.exe` na bilo koji računar koji pokreće Windows 10 i on će se pravilno pokrenuti, zato što sve kopije Windowsa 10 imaju instaliran .NET Framework 4.6. U stvari, možete da kopirate fajl i na računar sa starijom verzijom Windowsa ukoliko postoji instaliran .NET Framework 4.0 ili noviji.

Ispravljanje grešaka kompajlera

Ako kompajler prikaže greške, pročitajte ih pažljivo i ispravite ih u Notepadu. Snimite promene i rekompajlirajte.



U Command Prompt prozoru možete da pritisnete taster strelice nagore ili nadole na tastaturi da biste kružili kroz komande koje ste prethodno uneli.

Uobičajene greške nastaju zbog pogrešne komande, ili zato što nedostaje tačka-zarez na kraju linije ili se ne podudara par velikih zagrada. Na primer, ako ste pogrešno otkucali malo slovo `m` za metod `Main`, videćete sledeću poruku o grešci:

```
error CS5001: Program does not contain a static 'Main' method  
suitable for an entry point
```

Dekompajliranje koda pomoću ILDASM-a

C# kompajler konvertuje izvorni kod u Intermediate Language (IL) kod i čuva IL u sklopu (DLL ili EXE fajl).

Iskazi IL koda su slični instrukcijama jezika za sklop, ali ih izvršava .NET virtuelna mašina, poznatija kao Common Language Runtime (CLR).

Prilikom izvršavanja CLR učitava IL kod iz sklopa, koji JIT prevodi u izvorne CPU instrukcije, a CPU ga izvršava na računaru.

Prednost ovog procesa prevođenja, koji se sastoji iz dva koraka, je što „Microsoft“ može da kreira CLR-ove za Linux i Mac OS X, kao i za Windows. Isti IL kod može da se pokrene bilo gde, zbog drugog procesa prevođenja, koji generiše kod za set instrukcija za izvorni operativni sistem i CPU.

Bez obzira u kojem je jeziku pisan izvorni kod, sve .NET aplikacije koriste IL kod za instrukcije koje se čuvaju u sklopu. „Microsoft“ ima alatku koja može da otvori sklop i otkrije ovaj IL kod.



U stvari, ne koriste sve .NET aplikacije IL kod. Neke koriste novi .NET Native kompajler za generisanje izvornog koda, umesto IL koda, što poboljšava performanse i smanjuje upotrebu memorije, ali cena je nedostatak prenosivosti na druge platforme.

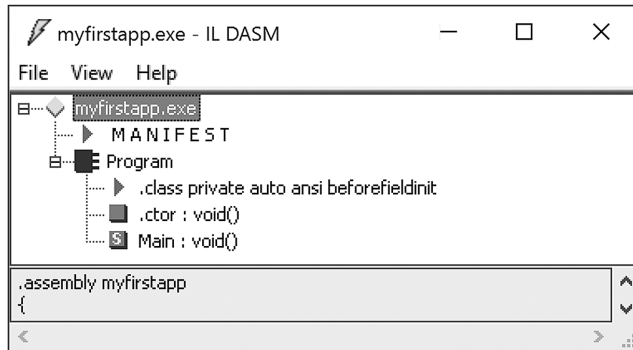
Razdvajanje prevedenih sklopova

Razdvajanje prevedenih sklopova je napredna tehnika. Predstaviću vam ovaj proces, ali nemojte brinuti ako još ne razumete u potpunosti ono što vidite!

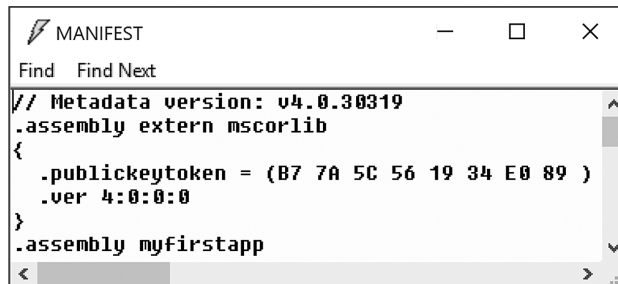
Pokrenite IL Disassembler, tako što ćete u prozor Developer Command Prompta uneti sledeće:

```
ildasm myfirstapp.exe
```

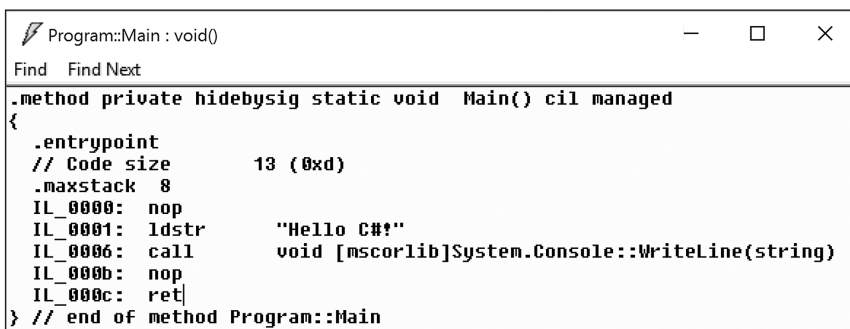
Videćete alatu IL DASM sa učitanim kompajliranim EXE sklopom:



Dvostruko kliknite na MANIFEST u prikazu stabla da biste prikazali metadata verziju (4.0.30319) .NET-a i verziju Microsoft Core Library (mscorlib) sklopa (4.0.0.0) koju ovaj sklop treba da pokrene. Na sledećoj slici vidite poruku da je, ako želimo da pokrenemo ovaj sklop, potrebno da bude instaliran .NET Framework 4.0 ili noviji:



Zatvorite prozor MANIFEST, proširite stavku Program, a zatim dvostruko kliknite na metod Main. Vidite IL instrukcije: ldstr (učitavanje niza), nop (nema operacije) i ret (vraćanje). Ne zaboravite da je IL jezik sklopa koji izvršava .NET CLR:



Svaki korisnik može da upotrebi ovu alatku da bi video IL kod .NET sklopa. Ne paničite! To nije razlog da izbegavate C#.

Sve aplikacije su liste instrukcija koje moraju da budu vidljive na mašini koja pokreće aplikacije. Ako mašina može da čita ove instrukcije, moći će da čita i sve ostalo. Stoga, svi programi mogu da budu obrnuto kreirani. Pitanje je samo koliko je potrebno truda. .NET vam sve to olakšava!



IL Spy je popularna alatka otvorenog koda koja može da izvršava sve što i IL DASM, a može i da preokrene IL kod u C# ili Visual Basic .NET izvorni kod. Ja sam koristio ovu alatku kada sam radio za klijenta koji je izgubio izvorni kod za sklop, koji je trebalo da bude vraćen.

PISANJE I PREVOĐENJE KODA POMOĆU MICROSOFT VISUAL STUDIOA 2015

Sada ćemo ponovo kreirati istu aplikaciju pomoću Microsoft Visual Studioa 2015.

Ja sam više od jedne decenije podučavao studente da koriste Visual Studio i uvek se iznenadim koliko programera ne uspeva da koristi ovu alatku.

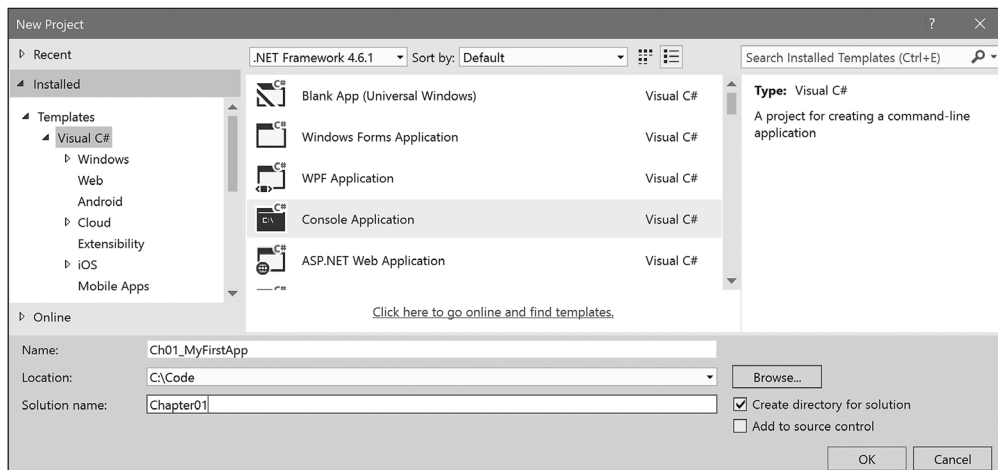
Na sledećih nekoliko stranica polako ćete savladati pisanje linije koda. Možda će vam to izgledati suvišno, ali ćete sigurno promeniti mišljenje kada vidite kakvu pomoć i informacije Visual Studio pruža kada unesete kod. Ako želite da postanete brz, precizan programer, velika je prednost da omogućite da Visual Studio ispiše vaš kod umesto vas.

Pisanje koda pomoću Visual Studioa 2015

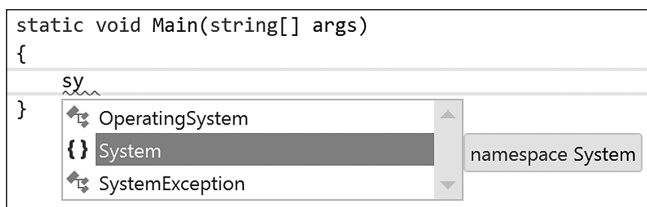
Pokrenite Microsoft Visual Studio 2015.

Kliknite na **File | New | Project** ili pritisnite prečicu **Ctrl + Shift + N**.

Na vrhu okvira za dijalog New Project izaberite **.NET Framework 4.6.1** (ili noviji). Iz liste **Installed Templates**, koja se nalazi na levoj strani prozora, izaberite **Visual C#**. Iz liste u centru prozora izaberite opciju **Console Application**. Unesite naziv **Ch01_MyFirstApp**, podesite lokaciju na **C:\Code**, unesite **Chapter01** kao naziv i kliknite na **OK** ili pritisnite **Enter**.

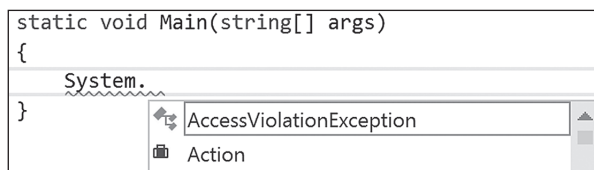


U editoru koda, unutar metoda `Main`, pritisnite **Enter** da biste ubacili novu liniju između unutrašnjih zagrada `{ }`, a zatim ukucajte slova `sy`, kao što možete da vidite na sledećoj slici.

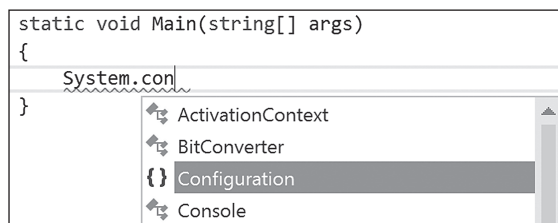


Videćete da IntelliSense prikazuje listu ključnih reči, imenskih prostora i vrsta, koja sadrži slova `sy` i ističe prvu stavku koja započinje slovima `sy`, što je, ujedno, i imenski prostor koji mi tražimo – `System`.

Ukucajte tačku (takođe je poznata kao decimalna tačka). IntelliSense automatski završava reč `System`, umesto vas, unosi tačku i prikazuje listu vrsta i imenskih prostora (kao što su `AccessViolationException` i `Action`) u nazivu `System`, kao na sledećoj slici.

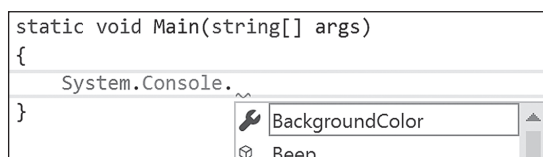


Ukucajte slova `con` i IntelliSense će prikazati listu podudarajućih vrsta i imenskih prostora:



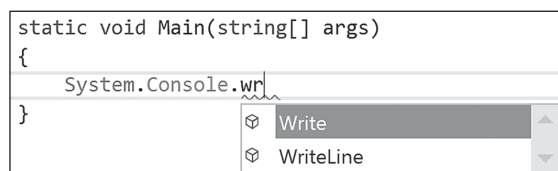
Pritisnite taster strelice nadole na tastaturi da biste istakli stavku **Console**, a zatim ukucajte tačku.

IntelliSense prikazuje listu članova klase `Console`:



Članovi uključuju parametre (atribute objekta, kao što je `BackgroundColor`), metode (akcije koje objekat može da izvrši, kao što je `Beep`) i slično.

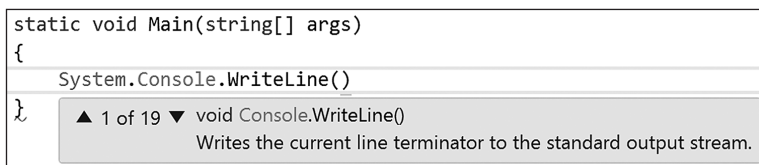
Ukucajte slova `wr`. IntelliSense prikazuje dva člana koja se podudaraju i sadrže ova slova:



Upotrebite taster strelice nadole da biste selektovali **WriteLine**, a zatim ukucajte otvorenu zagradu (`(`).

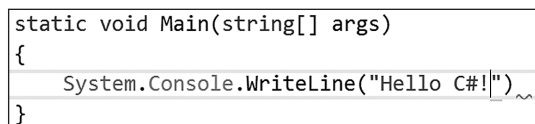
IntelliSense automatski popunjava `WriteLine` i unosi par zagrada.

Takođe ćete videti opis alatke koji ukazuje da metod `WriteLine` ima 19 varijacija:



Ukucajte dvostruki navodnik (`"`). IntelliSense unosi par dvostrukih navodnika, umesto vas, i ostavlja kursor između njih.

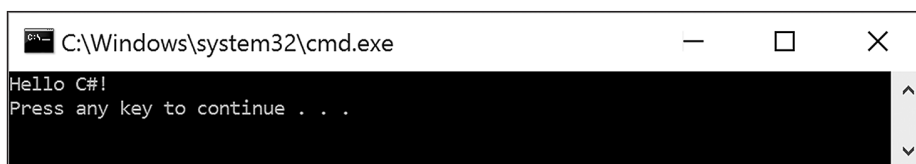
Ukucajte tekst `Hello C#!`, kao što je prikazano na sledećoj slici.



Crvena škrabotina na kraju linije ukazuje na grešku, zato što svaki `C#` iskaz treba da se završi znakom tačka-zarez. Pomerite kursor na kraj linije, tako što ćete pritisnuti taster `End` i ukucajte znak tačka-zarez da biste ispravili grešku.

Prevođenje koda pomoću Visual Studioa

Iz menija `Debug` izaberite **Start Without Debugging** ili pritisnite **Ctrl + F5**. Završena aplikacija biće pokrenuta u prozoru konzole i zatvoriće se kada pritisnete bilo koje dugme:



Da biste uštedeli prostor i učinili ispis jasnijim, obično neću prikazati slike ispisa iz konzole, kao što sam to radio ranije. Umesto toga, ispis ću prikazati ovako:

Hello C#!

Ispravljanje grešaka pomoću liste grešaka

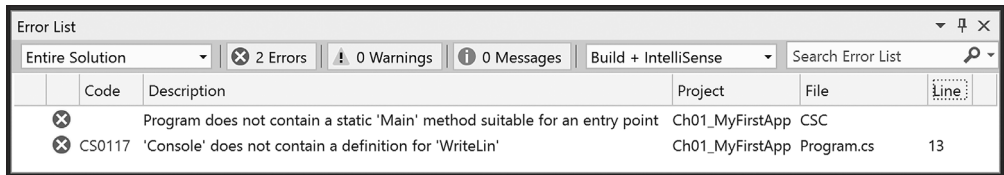
Hajde da namerno napravimo dve greške.

Promenite slovo `M` u metodu `Main` na malo slovo `m`.

Izbrišite slovo `e` na kraju naziva metoda `WriteLine`.

Iz menija **Build** izaberite opciju **Build Ch01_MyFirstApp** ili pritisnite **Shift + F6**.

Nakon nekoliko sekundi, statusna linija ukazuje da je izgradnja neuspešna i da je lista grešaka aktivirana. Takođe možete da prikazete listu grešaka ako pritisnete tastere **Ctrl + W, E**:



Error listu možete da filtrirate da prikazuje greške, upozorenja i informativne poruke, tako što ćete kliknuti na dugmad za prekopčavanje na vrhu prozora.

Ako greška prikaže broj linije - na primer, Line 13 na prethodnoj slici, možete dvostruko da kliknete na grešku da bi kursor preskočio do linije koja izaziva problem.

Ako je greška uobičajena, kao što je nedostatak metoda `Main`, kompajler neće moći da prikaže broj linije. Možda ćete želeti metod pod nazivom `main`, kao i metod pod nazivom `Main` (ne zaboravite da je C# osetljiv na veličinu slova, pa vam je dozvoljeno da upotrebite metode `main` i `Main`).

Ispravite ove dve greške (kao što je prikazano na prethodnoj slici) pre nego što nastavite rad. Vidite da je lista grešaka ažurirana i da ne prikazuje greške.

Eksperimentisanje sa C# Interactiveom

Iako je Visual Studio uvek imao Immediate prozor sa ograničenom podrškom **REPL** (**read-eval-print loop**), Visual Studio 2015 with Update 1 uključuje poboljšan prozor, koji sadrži ceo IntelliSense i kolorni kod sintakse pod nazivom **C# Interactive**.

U meniju **View** izaberite **Other Windows**, a zatim kliknite na **C# Interactive**.

Napisaćemo interaktivni kod za preuzimanje stranice About sa javnog web sajta „Microsoft“.



Ovo je samo primer; ne treba još da razumete kod.

U komandnu liniju C# Interactive unesite komande za:

- referenciranje `System.Net.Http` sklopa
- importovanje `System.Net.Http` imenskog prostora
- deklarisanje i instanciranje promenljive HTML klijenta
- podešavanje glavne adrese klijenta na web sajtu „Microsoft“
- asinhrono čekanje odgovora za GET upit za stranicu About
- čitanje koda statusa koji je vratio web server
- čitanje zaglavlja sadržaja
- čitanje sadržaja HTML stranice kao niza

Ukucajte svaku od sledećih komandi iza > prompta, a zatim pritisnite **Enter**:

```
> #r "System.Net.Http"
> using System.Net.Http;
> var client = new HttpClient();
> client.BaseAddress = new Uri("http://www.microsoft.com/");
> var response = await client.GetAsync("about");
> response.StatusCode
OK
> response.Content.Headers.GetValues("Content-Type")
string[1] { "text/html" }
> await response.Content.ReadAsStringAsync()
"<!DOCTYPE html ><html xmlns:mscom="http://schemas.microsoft.com/
CMSvNext" xmlns:md="http://schemas.microsoft.com/mscom-data"
lang="en" xmlns="http://www.w3.org/1999/xhtml"><head><meta
http-equiv="X-UA-Compatible" content="IE=edge" /><meta
charset="utf-8" /><meta name="viewport" content="width=device-
width, initial-scale=1.0" /><link rel="shortcut icon" href="//
www.microsoft.com/favicon.ico?v2" /><script type="text/javascript"
src="http://ajax.aspnetcdn.com/ajax/jquery/jquery-1.7.2.min.js">\
r\n      // Third party scripts and code linked to or referenced
from this website are licensed to you by the parties that own
such code, not by Microsoft. See ASP.NET Ajax CDN Terms of Use -
http://www.asp.net/ajaxlibrary/CDN.ashx.\r\n    </script><script
type="text/javascript" language="javascript">/*! [CDATA[*
if($(document).bind("mobileinit",function(){$.mobile.
autoInitializePage=!1}),navigator.userAgent.match(/IEMobile\\/10\\.0/))
{var msViewportStyle=document.createElement("style ...
```



```

C# Interactive
Microsoft (R) Roslyn C# Compiler version 1.1.0.51109
Loading context from 'CSharpInteractive.rsp'.
Type "#help" for more information.
> #r "System.Net.Http"
> using System.Net.Http;
> var client = new HttpClient();
> client.BaseAddress = new Uri("http://www.microsoft.com/");
> var response = await client.GetAsync("about");
> response.StatusCode
OK
> response.Content.Headers.GetValues("Content-Type")
string[1] { "text/html" }
> await response.Content.ReadAsStringAsync()
"<!DOCTYPE html ><html xmlns:mscom="http://schemas.microsoft.com..."

```

Ostali korisni prozori

Visual Studio ima mnogo drugih korisnih prozora, uključujući sledeće:

- ▣ prozor **Solution Explorer** za upravljanje svim projektima i fajlovima na kojima radite
- ▣ prozor **Team Explorer** za alate za upravljanje izvornim kodom, kao što je GitHub
- ▣ prozor **Server Explorer** za upravljanje konekcijama baze podataka

Ako se desi da ne možete da vidite prozor koji vam je potreban, kliknite na meni **View** da bi prozor ponovo bio prikazan ili pritisnite prečicu na tastaturi, kao što je prikazano na sledećoj slici.

View	Project	Build	Debug	Team	Tools
<>	Code			F7	
	Solution Explorer			Ctrl+W, S	
	Team Explorer			Ctrl+, , Ctrl+M	
	Server Explorer			Ctrl+W, L	
	SQL Server Object Explorer			Ctrl+, , Ctrl+S	
	Call Hierarchy			Ctrl+W, K	
	Class View			Ctrl+W, C	
	Code Definition Window			Ctrl+W, D	
	Object Browser			Ctrl+W, J	
	Error List			Ctrl+W, E	
	Output			Ctrl+W, O	



Ako se vaše prečice na tastaturi razlikuju od onih koje su prikazane na prethodnoj slici, razlog je što ste izabrali drugi set kada ste instalirali Visual Studio. Možete da resetujete prečice na tastaturi da se podudaraju sa onima koje ćemo upotrebiti u ovoj knjizi - kliknite na meni Tools, pa na Import and Export Settings..., izaberite Reset all settings, a zatim izaberite resetovanje Visual C# kolekciju podešavanja.

CILJANJE .NET COREA

U proteklih 15 godina .NET programeri su kodirali za .NET Framework, pa, stoga, postoji mnogo projekata koji treba da se promene - da se ne koriste samo na Windowsu, već i u međuplatformskom okruženju preusmeravanjem na .NET Core.

Razumevanje .NET Portability Analyzera

Da biste bili sigurni da će projekat funkcionisati na različitim platformama, možete da instalirate i pokrenete Visual Studio 2015 ekstenziju pod nazivom .NET Portability Analyzer.

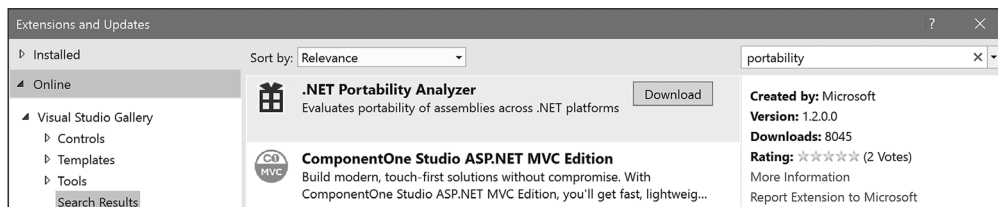
Ova ekstenzija omogućava da izaberete više platformi za koje biste želeli da programirate, a zatim skenira prevedene sklopove, prosleđuje listu vrsta i metoda koje koristite za web servis i kreira izveštaj koji lista koji od metoda nisu podržani na svakoj platformi, zajedno sa preporukom za ispravljanje.

Instaliranje .NET Portability Analyzera

U Visual Studiou kliknite na meni Tools i izaberite Extensions and Updates....

Sa leve strane okvira za dijalog kliknite na Online. U polje Search Visual Studio Gallery ukucajte portability i pritisnite Enter.

Kliknite na unos .NET Portability Analyzer, a zatim kliknite na dugme Download. Kada je ekstenzija instalirana, restartujte Visual Studio:



Konfigurisanje .NET Portability Analyzera

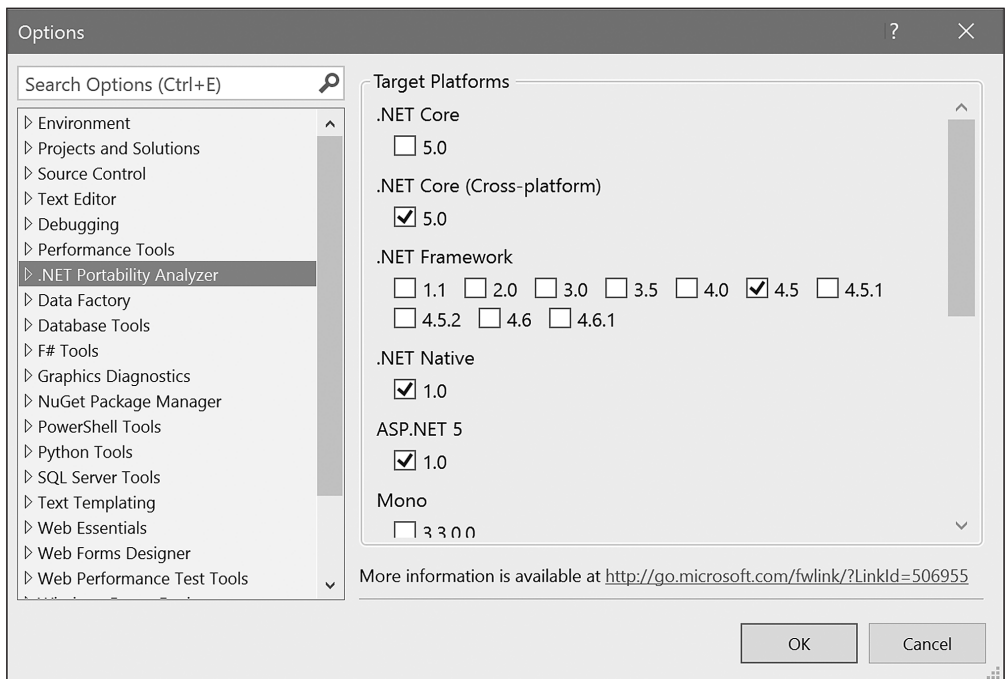
U prozoru Visual Studioa kliknite na meni **Tools** i izaberite **Options**.

U okviru za dijalog Options skrolujte dole u listi sa leve strane okvira za dijalog i izaberite **.NET Portability Analyzer**. U odeljku Target Platforms deselektujte sve, osim sledećih opcija:

- ☒ .NET Core (Cross-platform) 1.0 (prikazana kao 5.0 na sledećoj slici)
- ☒ .NET Framework 4.5
- ☒ .NET Native 1.0
- ☒ ASP.NET Core 1.0 (prikazana kao ASP.NET 5 na sledećoj slici):



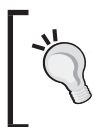
Microsoft Azure ima Platform-as-a-Service (PaaS) funkciju pod nazivom Cloud Services, koja je ograničena na .NET Framework 4.5, zbog čega je to uobičajena ciljna platforma za proveru pomoću analizera.



Analiziranje rešenja

U Visual Studiou otvorite prethodno rešenje pod nazivom Chapter01. U prozoru Solution Explorer kliknite desnim tasterom miša na projekat pod nazivom **Ch01_MyFirstApp** (ne na rešenja), izaberite **Analyze**, a zatim izaberite **Analyze Assembly Portability**.

Nakon nekoliko sekundi biće prikazan izveštaj u glavnom području.



Budite strpljivi kada koristite .NET Portability Analyzer, zato što on treba da pošalje podatke web servisu i da čeka odgovor.

Ne zaboravite da klasu Console u imenskom prostoru System i njen metod WriteLine podržavaju:

- ▣ .NET Core (Cross-Platform) 1.0 (prikazana kao 5.0 na sledećoj slici)
- ▣ .NET Framework 4.5
- ▣ ASP.NET Core 1.0 (prikazana kao ASP.NET 5 na sledećoj slici),

ali ih ne podržava .NET Native 1.0:

.NET Portability Report					
Submission Id c5a18f49-a75c-49bc-b3ef-fb17ed93ca81					
Contents					
Portability Summary					
Portability Summary					
Assembly	.NET Core (Cross-platform),Version=v5.0	.NET Framework,Version=v4.5	.NET Native,Version=v1.0	ASP.NET 5,Version=v1.0	
Ch01_MyFirstApp, Version=1.0.0.0, Culture=neutral, PublicKeyToken=null (.NETFramework,Version=v4.6)	100.00 %	100.00 %	93.94 %	100.00 %	
Ch01_MyFirstApp, Version=1.0.0.0, Culture=neutral, PublicKeyToken=null (.NETFramework,Version=v4.6)					
Target type	.NET Core (Cross-platform),Version=v5.0	.NET Framework,Version=v4.5	.NET Native,Version=v1.0	ASP.NET 5,Version=v1.0	Recommended changes
System.Console	✓	✓	✗	✓	
WriteLine (System.String)	✓	✓	✗	✓	



U vreme kada je ova knjiga pisana .NET Portability Analyzer nije bio ažuriran da prikazuje nove verzije za .NET Core i srodne proizvode, koji se mogu naći na adresi <http://www.hanselman.com/blog/ASPNET5IsDeadIntroducingASPNETCore10AndNETCore10.aspx>.

Pritisnite **Ctrl + W, E** da biste prikazali listu grešaka koja je, takođe, popunjena. Kao što ste već ranije videli, dvostrukim klikom na grešku editor koda se prebacuje na liniju koja možda izaziva problem:



.NET Portability i ova knjiga

Za pisanje svih kodova u ovoj knjizi ja sam upotrebio .NET Portability Analyzer da bih omogućio, kad god je to moguće, da prikazem kod koji će funkcionisati na različitim platformama, uz upotrebu .NET Core 1.0. Ako postoji neka korisna funkcija koja još nije međuplatformska, to ću istaći u tekstu. Nadam se da će ove funkcije biti implementirane u .NET Core u budućim verzijama.

Kreiranje novih projekata za .NET Core

Kreirajmo sada aplikaciju konzole, ali ovoga puta ćemo od početka ciljati .NET Core.

Upravljanje .NET Core programiranjem pomoću alati komandne linije

Postoje dva seta alati komandne linije koja možete da upotrebite za upravljanje .NET Coreom:

- ▣ alatke **.NET Version Manager**, **.NET Execution Environment**, **.NET Development Utilities** (`dnvm`, `dnx`, `dnu`), koje je koristio „Microsoft“ u toku početnog razvoja .NET Core 1.0 između 2013. i 2015. godine; instalirane su kao deo Visual Studioa 2015 i biće već zastarele u finalnom izdanju .NET Core 1.0

- ▣ drajver **.NET CLI** (`dotnet`) koji će upotrebiti finalno izdanje .NET Corea obezbeđuje pojednostavljen sloj povrh drugih alatki; mora biti posebno instaliran.



U vreme pisanja ove knjige DNX alatke su naprednije od .NET CLI-a, pa ću, stoga, u Poglavlju 15, „Upotreba C#-a na različitim platformama“, pokazati kako da upotrebite oba seta alatki. Za ovo poglavlje možemo da upotrebimo noviju CLI alatku, zato što ćemo koristiti samo osnovne funkcije. Za više informacija o promenama između vremena pisanja ove knjige i izdanja finalne verzije .NET Corea 1.0 i njegovih tehnologija i alatki preporučujem da pregledate stranicu <http://cs6dotnetcore.azurewebsites.net/>.

Instaliranje alatki .NET komandne linije

Prvo, treba da instalirate .NET Command Line Tools (CLI).

Pokrenite Microsoft Edge i otvorite stranicu <https://github.com/dotnet/cli>.

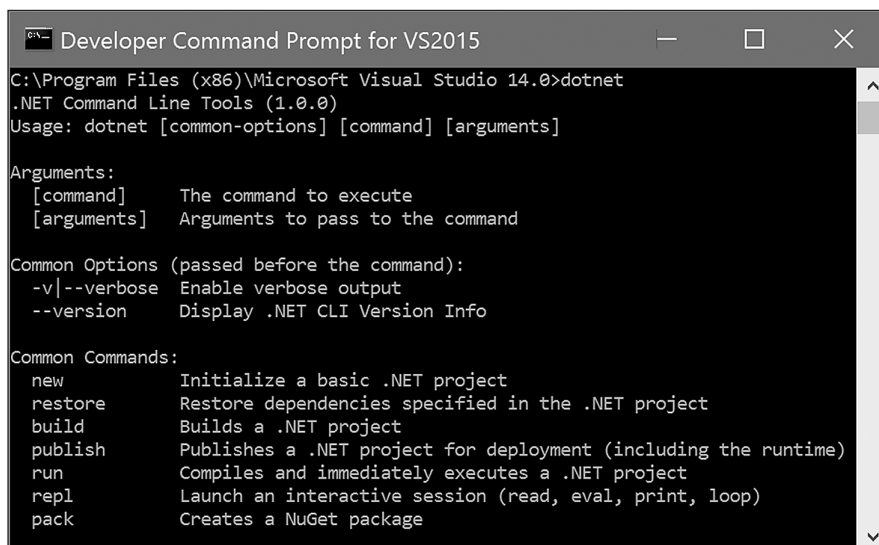
Skrolujte stranicu da biste pronašli odeljak Installers i kliknite na link da biste preuzeli MSI fajl za Windows. Otvorite fajl i instalirajte ga.

Kreiranje .NET Core aplikacije pomoću alatki komandne linije

Pokrenite Developer Command Prompt for VS2015. U komandnu liniju unesite sledeću komandu:

```
dotnet
```

Trebalo bi da vidite sledeći ispis:



```

Developer Command Prompt for VS2015
C:\Program Files (x86)\Microsoft Visual Studio 14.0>dotnet
.NET Command Line Tools (1.0.0)
Usage: dotnet [common-options] [command] [arguments]

Arguments:
  [command]      The command to execute
  [arguments]    Arguments to pass to the command

Common Options (passed before the command):
  -v|--verbose   Enable verbose output
  --version      Display .NET CLI Version Info

Common Commands:
  new            Initialize a basic .NET project
  restore        Restore dependencies specified in the .NET project
  build          Builds a .NET project
  publish        Publishes a .NET project for deployment (including the runtime)
  run            Compiles and immediately executes a .NET project
  repl           Launch an interactive session (read, eval, print, loop)
  pack           Creates a NuGet package
  
```

Da biste kreirali novi direktorijum, promenili putanju na novi direktorijum, kreirali novu aplikaciju konzole u direktorijumu, a zatim izlistali fajlove koje je aplikacija kreirala, u komandnu liniju unesite sledeći kod:

```

cd C:\Code\Chapter01
mkdir Ch01_MySecondApp
cd Ch01_MySecondApp
dotnet new
dir
  
```

Trebalo bi da vidite da je alatka dotnet kreirala tri nova fajla: NuGet.Config, Program.cs i project.json.

Da biste pokrenuli Notepad za editovanje klase Program, u komandnu liniju unesite sledeći kod:

```
notepad Program.cs
```

U Notepadu modifikujte iskaz WriteLine na sledeći način:

```
Console.WriteLine("Hello C#, Welcome .NET Core!");
```

Snimate promene i zatvorite Notepad.

U komandnu liniju unesite sledeće komande:

```

dotnet restore
dotnet run
  
```

Nakon nekoliko sekundi, svi paketi zavisnosti će biti preuzeti i aplikacija će biti pokrenuta, prikazujući sledeći ispis:

```
Developer Command Prompt for VS2015
C:\Code\Chapter01\Ch01_MySecondApp>dotnet restore
info : Running restore with 16 concurrent jobs.
info : Restoring packages for C:\Code\Chapter01\Ch01_MySecondApp\project.json...

info : All packages are compatible with DNXCore,Version=v5.0.
info : All packages are compatible with DNXCore,Version=v5.0 (win7-x64).
info : All packages are compatible with DNXCore,Version=v5.0 (win7-x86).
info : Committing restore...
info : Restore completed in 2549ms.

C:\Code\Chapter01\Ch01_MySecondApp>dotnet run
Hello C#, Welcome .NET Core!
```

Da biste preveli izvorni kod u sklop koji sadrži IL kod, unesite sledeću komandu:

dotnet build

Ispis u prozoru komandne linije treba da izgleda ovako:

```
Developer Command Prompt for VS2015
C:\Code\Chapter01\Ch01_MySecondApp>dotnet build

Project Ch01_MySecondApp will be compiled because expected outputs are missing.

Compiling Ch01_MySecondApp for DNXCore,Version=v5.0

Compilation succeeded.
    0 Warning(s)
    0 Error(s)

Time elapsed 00:00:02.5705911
```

Da biste se prebacili na poddirektorijum, izlistali fajlove koje on sadrži i izvršili prevedenu aplikaciju konzole, unesite sledeće komande:

cd bin\Debug\dnxcore50

dir

Ch01_MySecondApp



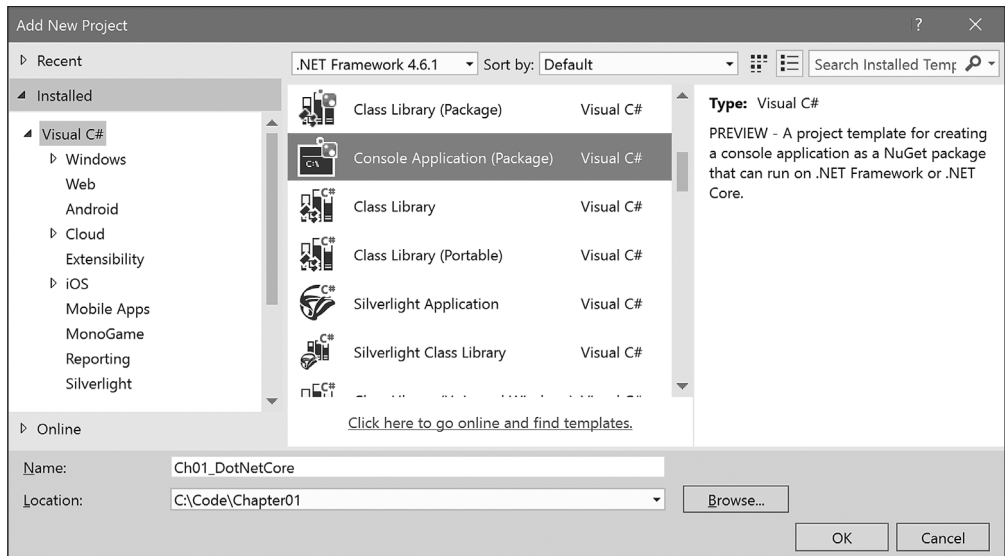
Debug je naziv konfiguracije, a dnxcore50 je naziv radnog okvira. To će biti promenjeno u finalnom izdanju .NET Core.

Kreiranje .NET Core aplikacije pomoću Visual Studioa 2015

Pokrenite Visual Studio 2015 i otvorite Chapter01 rešenje.

U prozoru Visual Studioa kliknite na **File | Add | New Project**.

U okviru za dijalog **Add New Project**, u listi **Installed Templates**, koja se nalazi na levoj strani okvira za dijalog, izaberite **Visual C#**. U listi koja se nalazi u centru okvira za dijalog izaberite opciju **Console Application (Package)**. Unesite naziv **Ch01_DotNetCore** i kliknite na **OK** ili pritisnite **Enter**:



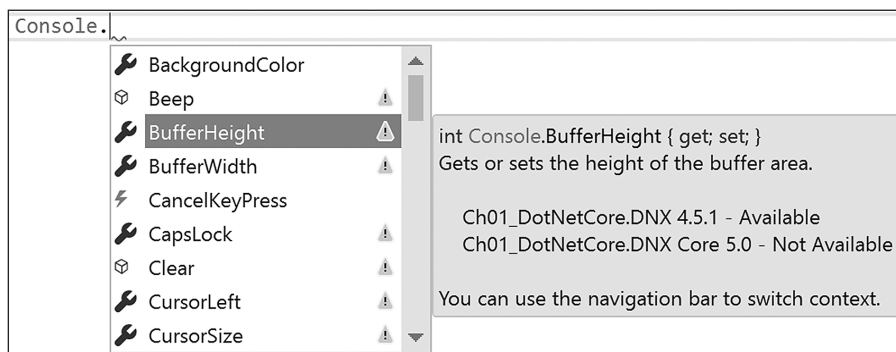
U prozoru Solution Explorer kliknite desnim tasterom miša na rešenje pod nazivom **Chapter01** i izaberite **Properties**.

Podesite opciju Startup Project na **Current** i kliknite na **OK**.

Unutar metoda Main klase Program unesite sledeće iskaze:

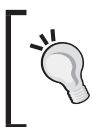
```
Console.WriteLine("Welcome, .NET Core!");
Console.ReadLine();
```

Dok kucate kod, videćete da Visual Studio prikazuje koje vrste i metodi možda neće biti dostupni kada preusmeravate .NET Core. Na primer, parametar `BufferHeight` je dostupan kada ciljate .NET Framework, ali ne i kada ciljate .NET Core, kao što je prikazano na sledećoj slici kao DNX 4.5.1 i DNX Core 5.0.



Možete da zatražite od kompajlera da uključi iskaze koji su dostupni samo kada je dostupan .NET Framework, koristeći simbole uslovnog prevođenja, kao što su sledeći:

```
#if DNX451
    Console.BufferHeight = 300;
#elseif DNXCORE50
    // some alternative for .NET Core
#endif
```



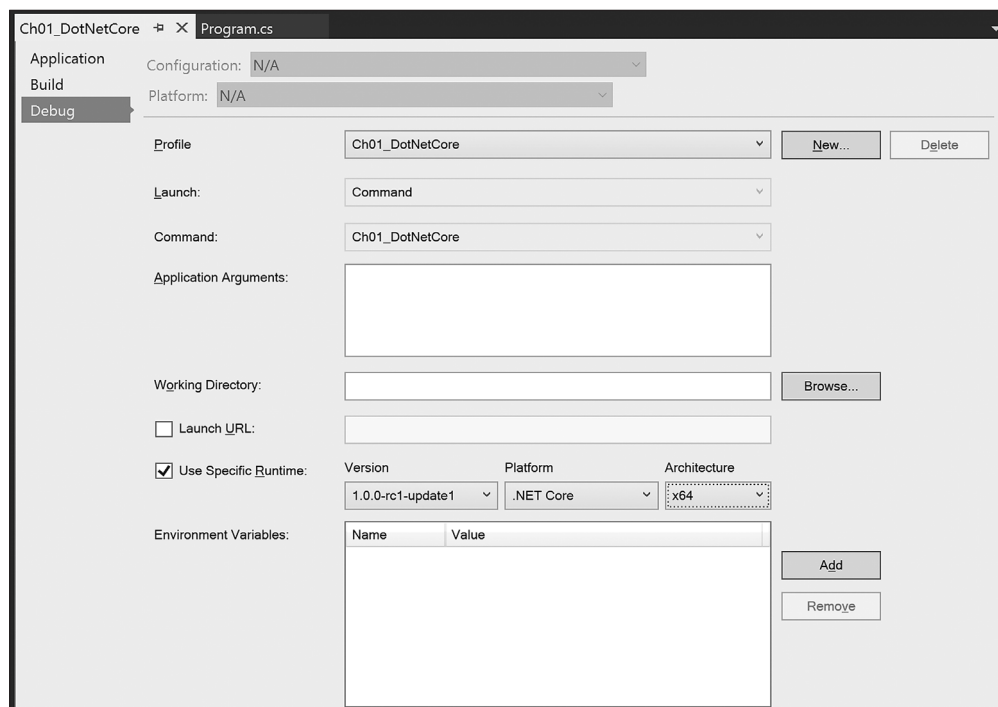
U Poglavlju 5, „Upotreba specijalizovanih .NET vrsta“, naučićete više o simbolima uslovnog prevođenja da biste mogli da kreirate projekte koji ciljaju .NET Framework i .NET Core, koristeći jednu osnovu koda.



Nazivi simbola uslovnog prevođenja će se u budućnosti menjati. Za više informacija posetite stranicu <https://github.com/dotnet/corefx/blob/master/Documentation/architecture/netplatform-standard.md>.

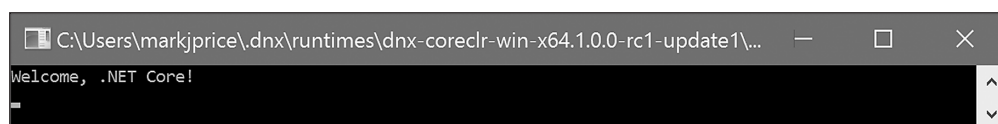
U prozoru Solution Explorer dvostruko kliknite na **Properties** za projekat **Ch01_DotNetCore**.

U prozoru Properties kliknite na karticu **Debug**, selektujte polje za potvrđivanje **Use Specific Runtime** i selektujte najnoviju verziju .NET Core platforme za x64 CPU arhitekturu, kao što je prikazano na sledećoj slici.



U **Visual Studio Debug** meniju izaberite **Start** ili pritisnite **F5**.

Videćete da prozor komandne linije prikazuje da aplikacija konzole pokreće **dnx-coreclr-win-x64.1.0.0-rc1-update1**:



U prvoj polovini ove knjige ja ću prikazati primer koda kreiranjem aplikacija konzole, pa možete da odlučite da li ćete da ciljate .NET Framework, tako što ćete izabrati Console Application, ili ćete da ciljate .NET Core, tako što ćete izabrati Console Application (Package).

UPRAVLJANJE IZVORNIM KODOM POMOĆU GITHUBA

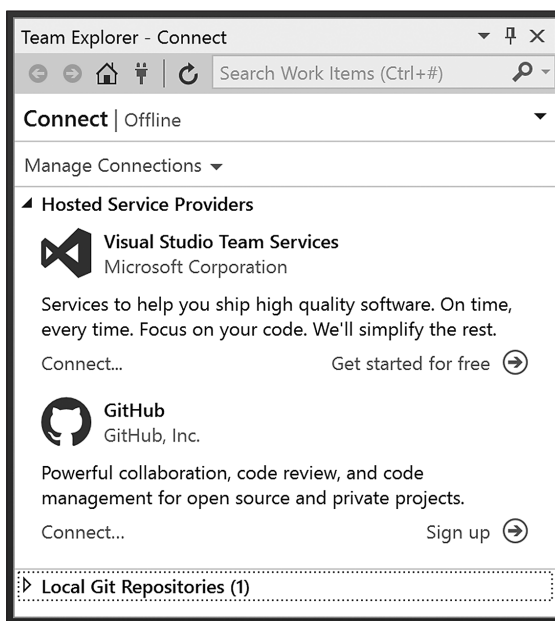
Git je sistem za upravljanje izvornim kodom koji se najčešće koristi. „GitHub“ je kompanija, web sajt i desktop aplikacija koja olakšava upravljanje pomoću Gita.

Visual Studio 2015 ima ugrađenu podršku za upotrebu Gita pomoću GitHuba, kao i pomoću „Microsoftovog“ sistema za upravljanje izvornim kodom, pod nazivom Visual Studio Team Services.

Ja sam upotrebio GitHub za čuvanje rešenja za sve praktične vežbe na kraju svakog poglavlja.

Upotreba prozora Team Explorer

U prozoru Visual Studioa kliknite na **View | Team Explorer** da biste prikazali prozor Team Explorer:



Iako bi bilo dobro da se prijavite provajderu za online sistem za upravljanje izvornim kodom, možete da klonirate GitHub skladište, bez prijavljivanja i otvaranja naloga.

Kloniranje GitHub skladišta

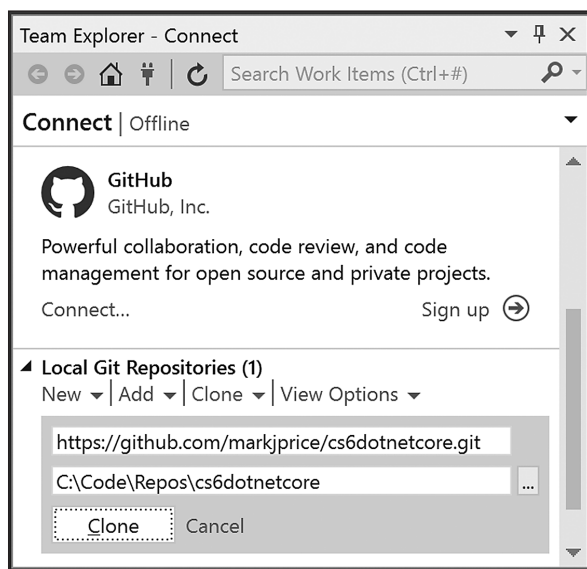
U prozoru Team Explorer raširite stavku **Local Git Repositories**, kliknite na meni **Clone**, a zatim unesite sledeći URL za Git skladište da biste ga klonirali:

`https://github.com/markjprice/cs6dotnetcore.git`

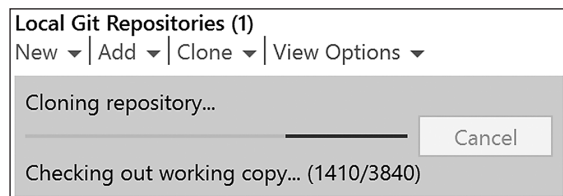
Unesite putanju za klonirano Git skladište

`C:\Code\Repos\cs6dotnetcore`

Kliknite na dugme **Clone**:



Sačekajte da se Git skladište klonira lokalno:



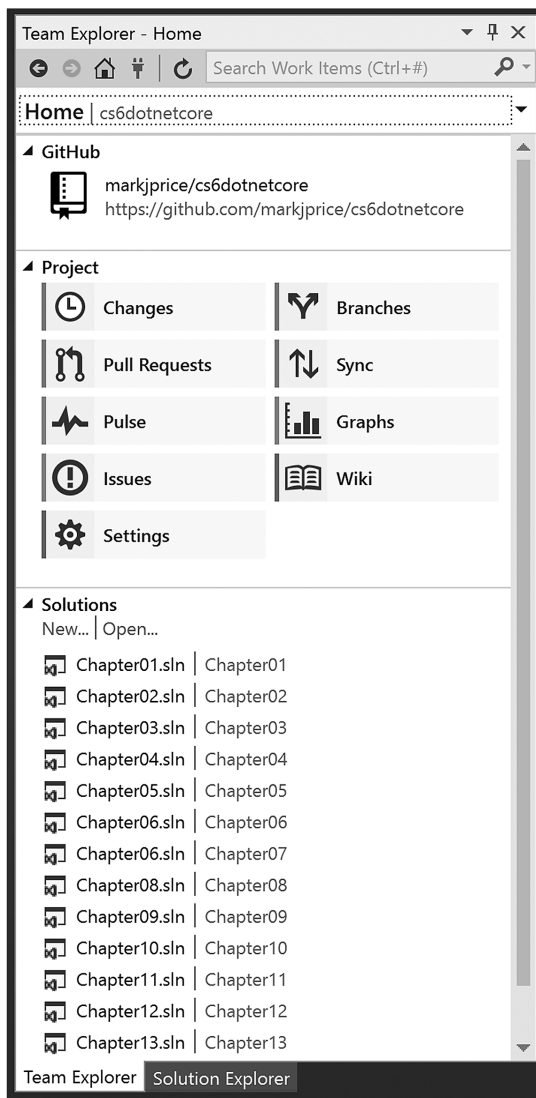
Sada ćete imati lokalnu kopiju kompletnih rešenja za sve praktične vežbe iz ove knjige.

Upravljanje GitHub skladištem

Dvostruko kliknite na skladište cs6dotnetcore da biste otvorili detaljni prikaz.

Možete da kliknete na opcije u odeljku **Project** da biste prikazali **Pull Requests** i **Issues** i druge aspekte skladišta.

Možete dvostruko da kliknete na unos u odeljku **Solutions** da biste ga otvorili u prozoru **Solution Explorer**:



VEŽBANJE I ISTRAŽIVANJE

Testirajte svoje znanje, tako što ćete odgovoriti na postavljena pitanja, uraditi praktičnu vežbu i istražiti detaljno teme koje su opisane u ovom poglavlju.

Vežba 1.1 – Testirajte svoje znanje

Odgovorite na sledeća pitanja:

1. Zašto programer može da upotrebi različite jezike za pisanje aplikacija koje se pokreću na .NET platformi?
2. Šta treba da ukucate u komandnu liniju da biste preveli C# izvorni kod?
3. Koje su prečice na tastaturi za Visual Studio 2015 za snimanje, prevođenje i pokretanje aplikacije, bez priključivanja funkcije za uklanjanje grešaka?
4. Koja je prečica na tastaturi za Visual Studio 2015 za prikaz liste grešaka?
5. Šta izvršava fajl ildasm.exe?
6. Da li je .NET Core bolji od .NET Frameworka?
7. Po čemu se .NET Native razlikuje od .NET Corea?
8. Šta izvršava .NET Portability Analyzer?
9. Koja je razlika između Gita i GitHuba?
10. Koji je naziv metoda za unos .NET aplikacije i kako treba da bude deklarisan?

Vežba 1.2 – Vežbajte upravljanje porzorima Visual Studioa

Vežbajte zatvaranje i prikazivanje prozora, kao što su Error List i Toolbox.

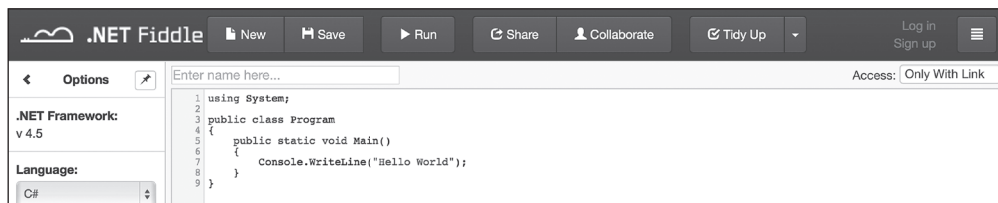
Prozore možete da prikačite i otkočite, da ih učitine plutajućim i da ih preuredite povlačenjem njihovih linija naslova. Dok povlačite liniju naslova prozora, Visual Studio će prikazati plave okvire i preklapanja, tako da ćete moći da razumete na koju lokaciju će prozor biti pomeren kada otpustite taster miša.



Uvek možete da resetujete Visual Studio na jedan od standardnih rasporeda, tako što ćete kliknuti na meni Tools, a zatim na opciju Import and Export Settings....

Vežba 1.3 – Vežbajte kodiranje bilo gde

Nije vam potreban Visual Studio da biste vežbali pisanje C#-a. Jednostavno, otvorite web sajt .NET Fiddle (<https://dotnetfiddle.net/>) i započnite kodiranje. Ovo je takođe odličan način za deljenje blokova C# koda sa drugim programerima:



Vežba 1.4 – Istražite teme

Upotrebite sledeće linkove da biste saznali više informacija o temama koje su opisane u ovom poglavlju:

- **Visual Studio IDE User's Guide:** <https://msdn.microsoft.com/en-us/library/dn762121.aspx>
- **Solutions and Projects:** <https://msdn.microsoft.com/en-us/library/b142f8e7.aspx>
- **Using IntelliSense:** <https://msdn.microsoft.com/en-us/library/hcw1s69b.aspx>
- **Identifying and Customizing Keyboard Shortcuts in Visual Studio:** <https://msdn.microsoft.com/en-us/library/5zwses53.aspx>
- **Making your libraries compatible with .NET Core and other .NET Platforms:** <https://github.com/dotnet/corefx/blob/master/Documentation/project-docs/support-dotnet-core-instructions.md>
- **.NET Core: A general purpose managed framework:** <http://dotnet.github.io>
- **Leveraging existing code across .NET platforms:** <http://blogs.msdn.com/b/dotnet/archive/2014/08/06/leveraging-existing-codeacross-net-platforms.aspx>
- **.NET Command Line Interface (CLI):** <https://github.com/dotnet/cli>
- **Use Visual Studio and Team Foundation Server with Git:** <https://msdn.microsoft.com/Library/vs/alm/Code/git/overview>
- **The easiest way to connect to your GitHub repositories in Visual Studio:** <https://visualstudio.github.com/>

REZIME

U ovom poglavlju smo podesili razvojno okruženje, upotrebili Developer Command Prompt za prevod i dekompajliranje aplikacije, upotrebili Visual Studio za kreiranje iste aplikacije, istražili funkcije koje su jednostavne za upotrebu pri programiranju i uočili razlike između .NET Frameworka, .NET Corea i .NET Nativea.

U sledećem poglavlju ćete naučiti da govorite C# jezikom.



IZDVOJENA ISKUSTVA ZAJEDNICE

C# 6 i .NET Core 1.0

Moderno međuplatformsko programiranje

Kreirajte moćne međuplatformske aplikacije, koristeći C# 6, .NET Core 1.0, ASP.NET Core 1.0 i Visual Studio 2015

Mark J. Price

 **komputer
biblioteka**

[PACKT]
PUBLISHING