Sadržaj ukratko

Deo 1: Osnove programiranja

Poglavlje 1 Početak 2
Poglavlje 2 Šta je programiranje? 18
Poglavlje 3 Pisanje programa 42
Poglavlje 4 Rad sa podacima u programu 68
Poglavlje 5 Odlučivanje u programu 100
Poglavlje 6 Ponavljanje postupaka sa petljama 134
Poglavlje 7 Korišćenje nizova 172

Deo 2: Napredno programiranje

Poglavlje 8 Korišćenje metoda za pojednostavljenje programa 212
Poglavlje 9 Izrada strukturnih tipova podataka 246
Poglavlje 10 Klase i reference 288
Poglavlje 11 Izrada rešenja sa objektima 336

Deo 3: Pravljenje igrica

Poglavlje 12 Šta čini igricu? 374
Poglavlje 13 Izrada pravila igranja 394
Poglavlje 14 Igrice i hijerarhija objekata 416
Poglavlje 15 JDSF lčsdkjf lačskjdf sdf

Sadržaj

Uvod xiv

Deo 1: Osnove programiranja

1 Početak 2

Priprema radnog mesta 4
Pribavljanje alatki i primera programa 4
Korišćenje alatki 5
Visual Studio projekti i rešenja 6
Pokretanje programa u okruženju Visual Studio 7
Zaustavljanje programa pokrenutih u prozoru Visual Studio 10
Aplikacija MyProgram 11
Šta ste naučili 15

2 Šta je programiranje? 18

Šta znači biti programer? 20
Programiranje i planiranje žurke 20
Programiranje i problemi 21
Programeri i ljudi 22
Računari kao obrađivači podataka 23
Mašine i računari i mi 23

Kako programi rade 26
Programi kao obrađivači podataka. 27
Podaci i informacije 35
Šta ste naučili 39

3 Pisanje programa 42

Struktura C# programa 44
Identifikovanje resursa 44
Početak definicije klase 45
Deklaracija metode StartProgram 46
Postavljanje naslova i prikazivanje poruke 47
Dodatne Snaps mogućnosti. 50
Metoda SpeakString 50
Izrada novih datoteka programa. 52
Dodatne Snaps mogućnosti. 61
Delay 61
SetTextColor 61
SetTitleColor 62
SetBackgroundColor 63
Pravljenje sopstvenih boja 63
Šta ste naučili 66

4 Rad sa podacima u programu 68

Promenljive za početak. 70
Promenljive i memorija računara 71
Deklarisanje promenljive 71
Jednostavni iskazi dodele 73
Korišćenje promenljive u programu 74
Dodeljivanje vrednosti u deklaraciji 76
Dodavanje stringova 77

Rad sa brojevima 80
Celi brojevi i realni brojevi 80
Obavljanje proračuna 83
Rad sa različitim tipovima podataka 85
Pretvaranje brojeva u tekst 86
Celi brojevi i relani brojevi u programima 89
Tipovi promenljivih i izrazi 89
Preciznost i tačnost 91
Pretvaranje tipova eksplicitnim dodeljivanjem tipa 92
Korišćenje eksplicitnog pretvaranja tipa na operandima u izrazima 93
Tipovi i greške 94
Dodatne Snaps mogućnosti. 95
Metode za vremenske prilike 95
Metoda ThrowDice 96
Šta ste naučili 97

5 Odlučivanje u programu 100

Upoznavanje logičkog tipa 102
Deklarisanje logičke promenljive 102
Logički izrazi 103
Korišćenje `if` konstrukcija i operatora. 104
Dodavanje dela else na konstrukciju if 105
Operatori poređenja 106
Operatori jednakosti 107
Poređenje stringova 109
Velika i mala slova 110
Pravljenje blokova iskaza 110
Lokalne promenljive u bloku koda 111
Pravljenje složenih uslova korišćenjem logičkih operatora . 113
Rad sa logikom 116
Dodavanje komentara kojim se pojašnjava program 117

Vožnje u luna parku i programi 119
Učitavanje brojeva 122
Logičko građenje korišćenjem if uslova 124
Dovršavanje programa 125
Rad sa pratećim elementima programa. 127
Upravljanje pratećim elementima programa u editoru Visual Studio 127
Emitovanje pratećih elemenata sa zvucima 128
Prikazivanje slika 129
Šta ste naučili 132

6 Ponavljanje postupaka sa petljama 134

Korišćenje petlje za pravljenje programa za naručivanje pice 136
Brojanje izbora 136
Prikazivanje ukupnih vrednosti 139
Preuzimanje izbora korisnika 139
Dodavanje petlje while 142
Proveravanje valjanosti ulaznih podataka petljom `while` 149
Korišćenje okruženja Visual Studio za praćenje izvršavanja programa 151
Brojanje u petlji kojim pravimo program za tablicu množenja 157
Korišćenje konstrukcije petlje for 160
Prekidanje petlje. 163
Prelazak na vrh petlje korišćenjem iskaza continue. 165
Dodatne Snaps mogućnosti. 168
Unos glasom 168
Tajni unos podataka 169
Šta ste naučili 170

7 Korišćenje nizova 172

Zaradite sladoled 174
Pamćenje podataka u jedinstvenim promenljivima . . . 175
Pravljenje niza 176
Korišćenje indeksa 177
Rad sa nizovima 179
Prikazivanje sadržaja niza korišćenjem petlje 184
Prikazivanje korisničkog menija 186
Sortiranje niza korišćenjem metode Bubble sort 187
Nalaženje najveće i najmanje vrednosti prodaje 194
Računanje vrednosti ukupne prodaje i prosečne prodaje 196
Dovršavanje programa 198
Više dimenzija u nizovima 199
Korišćenje ugnježđenih petlji for za rad sa dvodimenzionalnim nizovima 201
Pravljenje probnih verzija programa 203
Pronalaženje dužine dimenzije niza 204
Korišćenje niza kao tabele sa podacima 206
Šta ste naučili 208

Deo 2: Napredno programiranje

8 Korišćenje metoda za pojednostavljenje programa 212

Šta čini metodu? 214
Dodavanje metode u klasu 215
Unošenje informacija u metode korišćenjem parametara . 217
Vraćanje vrednosti iz metode 222
Izrada aplikacije za mini adresar 224

Učitavanje podataka o nekoj osobi 227
Čuvanje informacija o osobama iz adresara 228
Korišćenje lokalne memorije Windowsa za čuvanje podataka 229
Korišćenje reference parametara za preuzimanje rezultata iz pozivanja metode 231
Prikazivanje podataka o osobi iz adresara 237
Dodavanje IntelliSense komentara u vaše metode 241
Šta ste naučili 243

9 Izrada strukturnih tipova podataka 246

Čuvanje muzičkih nota korišćenjem strukture 248
Izrada i deklarisanje strukture 250
Pravljenje nizova od vrednosti strukture 252
Stukture i metode 253
Konstruisanje vrednosti strukture 256
Pravljenje programa za snimanje muzike 260
Izrada unapred definisanih nizova 262
Objekti i odgovornosti: omogućiti strukturi `SongNote` da svira samu sebe 263
Zaštita vrednosti koje se drže u strukturi. 264
Pravljenje programa za crtanje pomoću okruženja Snaps . . 267
Crtanje tačke na ekranu 268
Korišćenje Snap metode DrawDot za crtanje tačke na ekranu 269
Struktra SnapsCoordinate 270
Korišćenje Snaps metode GetDraggedCoordinate za otkrivanje položaja za crtanje. 272
Korišćenje Snaps metode SetDrawingColor za postavljanje boje za crtanje 274
Korišćenje Snaps metode ClearGraphics za brisanje ekrana 276
Struktura SnapsColor 277
Pravljenje nabrojivih tipova 278

Odlučivanje korišćenjem konstrukcije `switch` 280
Dodatne Snaps mogućnosti 282
Metoda GetTappedCoordinate 282
Metoda DrawLine 283
Metoda GetScreenSize 284
Metoda PickImage 285
Šta ste naučili 285

10 Klase i reference 288

Pravljenje programa za praćenje utrošenog vremena 290
Pravljenje strukture za držanje informacija o osobama iz adresara 290
Korišćenje reference this pri radu sa objektima 292
Upravljanje većim brojem podataka o osobama iz adresara 294
Pravljenje podataka za probu 296
Osmišljavanje korisničkog interfejsa programa za praćenje utrošenog vremena 297
Sastavljanje programa za praćenje utrošenog vremena 298
Pravljenje podataka o novoj osobi iz adresara 299
Nalaženje podataka o određenoj osobi 300
Dodavanje utrošenih minuta za osobe iz adresara 302
Prikazivanje sažetog pregleda 304
Strukture i klase 306
Sortiranje i strukture 306
Sortiranje i reference 307
Reference i tipovi vrednosti 308
Reference i dodeljivanje vrednosti 311
Klase i konstruktori 316
Nizovi od referenci klase 317
Od nizova do listi 319
Rad sa listama podataka 321
Liste i vrednost indeksa 322
Liste struktura 322

Čuvanje podataka korišćenjem formata JSON 323
Newtonsoft biblioteka JSON 324
Čuvanje i oporavljanje listi 326
Preuzimanje podataka jezikom XML 329
Šta ste naučili 334

11 Izrada rešenja sa objektima 336

Pravljenje objekata sa integritetom 338
Zaštita podataka koji se drže unutar nekog objekta . . 338
Pravljenje metoda za preuzimanje i postavljanje privatnih podataka. 341
Pravljenje metoda koje odražavaju korišćenje objekta 343
Korišćenje svojstava za upravljanje pristupom podacima . . 346
Korišćenje svojstava za primenjivanje pravila poslovanja. 349
Upravljanje procesom konstruisanja objekta 351
Hvatanje i obrada izuzetaka 353
Pravljenje upotrebljivih aplikacija 355
Čuvanje crteža u datotekama 356
Metoda SaveGraphicsImageToFileAsPNG 357
Metoda SaveGraphicsImageToLocalStoreAsPNG. 358
Metoda LoadGraphicsPNGImageFromLocalStore 358
Struktura DateTime 359
Dobijanje trenutnog datuma i vremena 360
Pretapanje prikaza datuma i vremena 360
Korišćenje datuma i vremena za davanje naziva datotekama 361
Pravljenje klase Drawing. 362
Pravljenje liste crteža. 364
Pravljenje metoda za zbirku crteža. 365
Šta ste naučili 368

Deo 3: Pravljenje igrica

12 Šta čini igricu? 374

Pravljenje video igrice 376
Igrice i mehanizmi igrice 376
Igrice i sprajtovi 378
Šta ste naučili 392

13 Izrada pravila igranja 394

Pravljenje reketa kojim upravlja igrač 396
Dodavanje zvukova u igricu 401
Prikazivanje teksta u igrici 403
Izrada čitave igrice 408
Šta ste naučili 414

14 Igrice i hijerarhija objekata 416

Igrice i objekti: Svemirske rakete u svemiru 418
Konstruisanje sprajta zvezde koja se kreće 419
Dozvoljavanje metodama da budu nadjačane 427
Pravljenje neba sa zvezdama koje se kreću 428
Pravljenje rakete koja se zasniva na klasi MovingSprite 430
Dodavanje nekoliko vanzemaljaca 432
Projektovanje hijerarhije klasa 440
Šta ste naučili 443

15 Igrice i softverske komponente 446

Igrice i objekti 448

Pravljenje objekata koji sarađuju 448

Objekti i stanje 456

Interfejsi i komponente 465

Šta ste naučili 471

Indeks 474

Uvod

Mislim da je programiranje nešto najkreativnije što možete da naučite. Ako naučite da slikate, možete da pravite slike. Ako naučite da svirate violinu, možete na njoj da izvodite muziku. Ali, ako naučite da programirate, možete da stvarate nešto do tada neviđeno (a takođe, možete i da slikate i da svirate, ako to želite). Pošto krenete putem programiranja, nema ograničenja dokle možete da stignete. Uvek će biti novih uređaja, tehnologija i tržišta gde možete da iskoristite svoje programerske veštine.

Ovu knjigu možete da zamislite kao vaš prvi korak na putovanju do programerskog prosvetljenja. Na najbolja putovanja se kreće sa odredištem u mislima, tako da ni ovo nije drugačije. Sviđa mi se da to odredište opisujem kao „korisnost". Na kraju ove knjige nećete biti najbolji programer na svetu (osim ako ja ne odem u penziju, naravno), ali imaćete dovoljno veština i znanja da pišete prilično korisne programe. A možda ćete imati bar jedan program dostupan svima širom sveta za preuzimanje sa lokacije Microsoft Store.

Međutim, pre nego što počnemo, želeo bih da vas na nešto upozorim. Slično kao što vodič želi da vam kaže nešto o lavovima, tigrovima i krokodilima na koje možete da naiđete pošto krećete na safari, osećam da moram da vas upoznam sa time da naše putovanje neće baš uvek teći glatko. Programeri moraju da nauče da o rešavanju problema razmišljaju pomalo drugačije, zato što računari ne rade na isti način kao i mi. Ljudi mogu da urade složene stvari, ali to rade sporo. Računari rade jednostavne stvari, ali to rade izuzetno brzo. Posao programera je da iskoristi te jednostavne mogućnosti računara za rešavanje složenih problema. To je ono čemu nameravamo da vas naučimo.

Ključ za uspeh programera je uglavnom potpuno isti kao i za većinu drugih poduhvata. Ako želite da postanete svetski poznati violinista, morate puno da vežbate. Isto važi i za programiranje. Morate da provedete mnogo vremena radeći na svojim programima da biste stekli veštinu pisanja programa. Dobra vest je da, kao što violonista uživa u sviranju na svom instrumentu, pošto naterate računar da uradi tačno ono što želite to stvara osećaj pravog zadovoljstva. A još je veće zadovoljstvo kada vidite druge ljude kako koriste programe koje ste napisali i misle da su korisni i zabavni za upotrebu.

Kako je ova knjiga spakovana

Ovu knjigu sam organizovao u četiri dela. Svaki deo nadograđuje prethodni sa ciljem da od vas napravi uspešnog programera. Počećemo razmatranjem programskih instrukcija nižeg nivoa koje programi koriste da bi računaru saopštili šta da radi, a završićemo pregledom profesionalnih softverskih rešenja.

Deo 1: Osnove programiranja

Prvi deo je odakle počinjete. Ukazuje vam se na to gde ćete instalirati i koristiti programerske alate koje su vam neophodni, a upoznaje vas i sa osnovnim elementima programskog jezika C# koje koriste svi programi.

Deo 2: Napredno programiranje

Drugi deo opisuje mogućnosti programskog jezika C# koje se koriste za pravljenje mnogo složenijih aplikacija. Pokazuje vam kako da velike programe razbijete u manje elemente i kako da napravite sopstvene tipove podataka koji odražavaju poseban problem koji rešavate. Saznaćete i kako programi mogu da sačuvaju podatke u memoriji kada sa ne izvršavaju.

Deo 3: Pravljenje računarskih igrica

Pravljenje računarskih igrica je prava zabava. Sa druge strane je i sjajan način da naučite kako se koriste tehnike objektno orijentisanog programiranja. U ovom delu pravićete igrice sa kojima je moguće da se zaista igra, a istovremeno ćete naučiti osnove o tome kako da ono što programirate proširite nasleđivanjem i osmišljavanjem softvera koje se zasniva na komponentama.

Deo 4: Pravljenje aplikacija

Četvrti deo je mesto gde ćete pronaći kako se prave složene aplikacije. Otkrićete kako se dizajnira grafički korisnički interfejs i kako da kôd programa povežete sa elementima na ekranu. Takođe, naučićete na koji način su uređene savremene aplikacije. Četvrti deo se ne nalazi u ovoj štampanoj knjizi, već je dostupan kao e-knjiga, koja je besplatna za preuzimanje sa veb stranice ove knjige na adresi https://aka.ms/BeginCodeCSharp/downloads.

Kako ćete učiti

U svakom poglavlju reći ću vam ponešto više o programiranju. Pokazaću vam kako se nešto radi, a zatim ću pozvati vas da uradite nešto sami, koristeći ono što ste naučili. Nikada nećete biti više od stranice ili dve udaljeni od nečega jedinstvenog i ličnog što možete da uradite ili napravite. U svakom poglavlju koristićemo biblioteku *Snaps* sa već ugrađenim funkcionalnostima, za koje ću vam pokazati kako se koriste. Posle toga, na vama je da napravite nešto čudesno!

Ovu knjigu možete da čitate kao i svaku drugu ako tako želite, ali naučićete mnogo više ako usporite i usput proradite praktične delove. Ova knjiga ne može da vas nauči kako se programira, kao što ni knjiga o biciklima ne može da vas nauči da vizite bicikl. Biće potrebno vreme i vežbanje da biste naučili kako se to radi. Ali, ova knjiga će vam dati znanje i smelost da se okušate u programiranju, a takođe, biće tu da vam pomogne ako vam programiranje ne ide onako kako očekujete. Ovo su ti elementi knjige koji će pomoći da, uz rad, zaista nešto naučite!



NEKA SE NEŠTO DEŠAVA

Da, najbolji način za učenje nečega je da se to uradi, tako da ćete elemente „Neka se nešto dešava" nalaziti u čitavom tekstu. Ovi elementi pomažu vežbanje vaših programerskih veština. Svaki od njih počinje primerom, a zatim predstavlja neke korake koje možete samostalno da isprobate. Sve što napravite izvršavaće se na Windows računarima, tabletima ili telefonima. Svoje kreacije možete i da objavite čitavom svetu putem lokacije Windows Store.



ANALIZA KODA

Sjajan način za učenje programiranja je pregledanje koda koji su napisali drugi ljudi i razmatranje šta radi (ponekad i zašto ne radi ono što bi trebalo). U tim isprobavanjima, u izdvojenim delovima „Analiza koda", koristićete svoje deduktivne veštine da biste shvatili ponašanje programa, ispravili greške i predložili poboljšanja.



ŠTA BI MOGLO DA POĐE NAOPAKO

Ako već ne znate da programi mogu da otkažu, naučićete ovu tešku lekciju vrlo brzo pošto počnete da pišete prvi program. Da bih vam pomogao da se sa time unapred izborite, dodao sam elemente „Šta bi moglo da pođe naopako?", kojima se predviđaju problemi koje biste možda mogli da imate i nude rešenja za te probleme. Na primer, kada uvodim nešto novo, ponekad ću deo vemena provesti razmatrajući kako to može da otkaže i o čemu je potrebno da brinete kada koristite tu novu mogućnost.

SAVETI PROGRAMERA

Dobar deo svog vremena proveo sam predajući programiranje. Ali sam napisao i mnogo programa, od kojih sam nekoliko prodao korisnicima koji su želeli to da plate. Za neke stvari koje sam naučio na teži način, zaista bih voleo da sam znao od samog početka. Cilj „Saveta programera" je da vam ukaže na te informacije tako da od početka steknete profesionalni pogled na razvoj softvera dok učite kako se to radi.

„Saveti programera" pokrivaju širok opseg pitanja, od programiranja do ljudi i filozofije. Preporučujem da pažljivo pročitate i prihvatite te savete – mogu vam ubuduće uštedeti puno vemena!

Programi i Snaps

Više niko ne pravi programe od samog početka. Svi programi se prave korišćenjem delova softvera koji su već napravljeni. Ako neki program treba da da prikazuje tekst, pušta muziku ili emituje neki video zapis, jednostavno zatraži od drugog programa da to uradi. Svi poznati računarski jezici poduprti su ogromnim bibliotekama postojećeg koda, a jedna od stvari koju je potrebno da programer razume je kako se koriste te biblioteke i softver koji su napisali drugi ljudi.

Biblioteku Snaps sam napravio posebno za ovu knjigu. Ona nudi skup funkcionalnih ponašanja koja se lako koriste i međusobno uklapaju. Biblioteku Snaps koristićete u svojim prvim programima. Kasnije u knjizi, otkrićete ostale biblioteke sa funkcionalnostima koje možete koristiti za izgradnju programa.

Programi koji koriste Snaps, izvršavaju se unutar mehanizma Snaps, koji predstavlja samostalno okruženje u kome programi mogu da izgovaraju poruke, prihvataju unose korisnika, crtaju slike, proizvode zvukove, pa čak i da odrede kakve su trenutne vremenske prilike.

Daću vam primere o tome kako Snaps radi i zatim prepustiti vama da vidite šta sve možete da uradite sa njim. Ideja koju sledimo je: „Ako programiranje ne možete da iskoristite da biste time zadivili prijatelje i porodicu, u čemu je onda njegova svrha?" Zaista se nadam da ćete se pojaviti sa nekim zadivljujućim programima, i možda ih čak i objavite, kako bi i drugi uživali u njima.

SAVET PROGRAMERA

Sve se pravi na kodu nekog drugog

Lepo se uklopilo da prvi savet programera bude baš o tome kako dobar programer može da bude „kreativno lenj". Nikada neće da piše program ako može da pronađe način da iskoristi neki koji je već napisan. (Zašto ponovo pronalaziti točak?) Bibliteka Snaps koju vam pružam je primer toga. Kasnije u ovoj knjizi, otkrićete da i ona sama koristi druge biblioteke.

Softver i hardver

Potreban vam je računar i neki softver da biste radili sa programima u ovoj knjizi. Bojim se da vam ne mogu pomoći što se tiče računara, ali ćete u prvom poglavlju pronaći gde možete da nabavite Visual Studio 2015 Community Edition, besplatni softver koji možete koristiti za pravljenje svojih programa. Takođe, naučićete gde možete da preuzmete biblioteku Snaps i primere koda koji ćemo istraživati i koristiti.

Računar koji koristite mora da pokreće 64-bitnu verziju opeartivnog sistema Windows 10. Ovo su ostali zahtevi:

- procesor brzine 1 GHz ili veće, poželjno Intel i5 ili bolji,
- najmanje 4 gigbajta (GB) memorije (RAM), ali je poželjno 8 GB ili više,
- za potpunu instalaciju Visual Studio 2015 Community potrebno je oko 8 GB prostora na hard disku.

Što se tiče grafičkog prikaza, nema posebnih zahteva, mada, na ekranima veće rezolucije moći ćete da vidite više dok pišete svoje programe. Biblioteka Snaps radi sa ekranima osetljivim na dodir, mišem, elektronskim olovkama za unos i Xbox One i Xbox 360 kontrolerima za igrice koje ćete razvijati u trećem delu.

Visual Studio 2015 Community Edition je besplatno dostupna aplikacija koju možete da koristite za pravljenje C# programa na računaru sa operativnim sistemom Windows 10. Ako na računaru već imate prethodnu verziju aplikacije Visual Studio (na primer, Visual Studio 2013), bojim se da nećete moći da je koristite uz ovu knjigu. Međutim, verzija 2015 aplikacije Visual Studio bez ikakvih problema radi uporedo sa postojećom instalacijom. U prvom poglavlju dajem vezu do detaljnih uputstava o tome kako da instalirate Visual Studio i pripremite ga za rad. Da biste što bolje iskoristili Visual Studio, najbolje je da imate Microsoftov nalog tako da ovlašćenje za programiranje bude dodeljeno vama.

Preuzimanja

U svakom poglavlju ove knjige prikazaću i objasniću programe koji vas uče o tome kako da započnete neki program – a koje možete iskoristiti da napravite sopstvene programe. Biblioteku Snaps, uzorke koda iz ove knjige, upustva za instaliranje i postavku aplikacije Visual Studio, kao i e-knjigu, odnosno četvrti deo pod nazivom „Pravljenje aplikacija" možete da preuzmete sa sledeće stranice na internetu:

https://aka.ms/BeginCodeCSharp/downloads

Sledite uputstva koja ćete pronaći u prvom poglavlju i u dokumentu za instaliranje da biste instalirali primere programa i kôd.

Zahvalnice

Zaista volim da pišem knjige! Veliku zahvalnost zaslužuju Devon Musgrave i ljudi u Microsoft Pressu koji su mi dali priliku da napišem još jednu, Rob Nance za čudesne crteže i John Pierce i Lance McCarthy za izuzetan posao koji su uradili sa tekstom. Desilo se da je zahvalnica jedini deo knjige koji oni nisu videli, a svima njima moram da se beskrajno zahvalim što su učinili sve da moj tekst bude lako čitljiv.

Greške, ispravke & podrška za knjigu

Učinili smo sve da ova knjiga i sadržaj koji ide uz nju budu bez grešaka. Možete da pronađete ispravke za ovu knjigu – u obliku spiska dostavljenih grešaka i njima odgovarajućih ispravki – na adresi:

https://aka.ms/BeginCodeCSharp/errata

Ako pronađete grešku koja još uvek nije navedena, molimo da je dostavite do navedene adrese.

Ako vam je neophodna dodatna podrška, pošaljite e-poruku timu Microsoft Pressa na adresu *mspinput@microsoft.com*.

Imajte na umu to da se tehnička podrška za Microsoftov softver i hardver ne nudi na prethodnim adresama. Za pomoć u vezi sa Microsoftim softverom i hardverom idite na adresu *http://support.microsoft.com*.

Možete takođe da pronađete „autorove beleške" o ovoj knjizi, uključujući i ostale projekte i informacije o biblioteci Snaps na adresi:

http://www.robmiles.com/begintocode

Besplatne e-knjige iz Microsof Pressa

Od tehničkih prikaza do podrobnih informacija o posebnim temama, besplatne e-knjige iz Microsoft Pressa obuhvataju širok spektar oblasti. Ove e-knjige dostupne su u formatima PDF, EPUB i Mobi za Kindle čitače i možete ih odmah preuzimati sa adrese:

http://aka.ms/mspressfree

Što češće proveravajte ovu adresu da biste videli šta ima novo!

Želimo da čujemo vaše mišljenje

Nama u Microsoft Pressu je vaše zadovoljstvo na prvom mestu, a povratne informacije koje dobijamo od vas su nešto najvrednije što imamo. Molimo vas, recite nam šta mislite o ovoj knjizi na:

http://aka.ms/tellpress.

Znamo da ste zauzeti, tako da vam nećemo oduzeti mnogo vremena, ako postavimo par kraćih pitanja. Vaši odgovori idu neposredno urednicima iz Microsoft Pressa. (Ne tražimo nikakve lične informacije.) Unapred zahvaljujemo na učešću!

Ostanimo u kontaktu

Nastavimo naš razgovor! Naša adresa na Tviteru je: *http://twiter.com/MicrosoftPress*.





Deo 1

Osnove programiranja

Krenimo na put programerskog prosvetljenja. Počećemo instaliranjem neophodnih programerskih alatki. Posle toga, otkrićete šta računari, u suštini, rade i šta su programski jezici. Takođe, napravićete prvi mali korak u korišćenju jezika C# da biste računaru saopštili da nešto uradi za vas, a videćete i kako se radi sa bibliotekom Snaps, malim pomoćnim programima koje sam napravio da biste ih iskoristili u svojim prvim programima.

Cilj prvog dela je da vas upozna sa osnovnim elementima programskog jezika C#, koje koriste svi programi. Zatim, u drugom delu, razmotrićemo kako savremeni programski jezici, kao što je C#, na osnovu tih osnovnih programskih elemenata olakšavaju pravljenje aplikacija.

1

Početak



Šta ćete naučiti

Programeri imaju skup alatki i tehnika koje koriste kada prave programe. U ovom poglavlju naučićete kakav računar vam je potreban za pisanje programa i kako da instalirate alatke koje ćete koristiti za građenje svog koda. Koristeći primere aplikacija iz ove knjige, napravićete prve korake u programiranju.

Priprema radnog mesta 4

Pribavljanje alatki i primera programa 4

Korišćenje alatki 5

Šta ste naučili 15



Priprema radnog mesta

Da ste vozač kamiona koji provodi mnogo vremena u kabini, prevozeći robu širom zemlje, voleli biste da ta kabina ima udobno sedište, dobar pregled puta i komande koje su lake za korišćenje. Nije beznačajno da taj kamion ima dovoljno snage da se pristojnom brzinom kreće uzbrdicama i da se njime lako upravlja na krivudavim planinskim putevima.

Slično tome, ako očekujete da provedete neko vreme za tastaturom pišući programe, potrebno je da imate pristojno mesto za rad. Ako možete, pronađite mesto gde ćete smestiti računar, tastaturu i ekran, a zatim postavite stolicu tako vam bude udobno da u njoj možete da provedete više sati.

Nije vam potreban posebno skup računar za pisanje programa, ali je neophodno da ima dovoljno memorije i brz procesor da bi radio sa alatkama koje ćete koristiti. Moj savet je da nabavite računar sa Windowsom 10, koji ima bar Intel i5 ili odgovarajući procesor, 4 gigabajta memorije i 256 gigabajta slobodnog prostora na hard disku. Možete da koristite i računar slabiji od toga, ali će uz njega postupak razvoja programa biti pomalo zamoran, pošto će mu biti potrebno više vremena da ažurira vaš program ako u njemu bilo šta promenite.

Ono što je najvažnije je da *morate da imate računar sa 64-bitnom verzijom Windowsa 10*. Veoma mali uređaji su sa 32-bitnom verzijom Windowsa 10. Ovo je sasvim pogodno za većinu aplikacija, ali ne može da se koristi za pravljenje Windows 10 aplikacija paketom Visual Studio, alatkom koju ćemo koristiti.

Pribavljanje alatki i primera programa

Sve alatke koje ćemo koristiti su besplatne za preuzimanje i instaliranje. Mislim da je čudesno i zadivljujuće da je tako moćan softver besplatno dostupan svima za korišćenje. Program Visual Studio olakšava pravljenje aplikacija i igrica. Čak pomaže da vaše programe dostavite na prodaju na lokaciji Microsoft Store.

Preporučujem da počnete program kao da je neki od onih koji biste vi želeli da kupite. Lepo je kad mislite da će softver koji pravite biti dostupan svima na lokaciji Store.

Međutim, pre nego što počnete sa deljenjem ili prodajom svojih programerskih rukotvorina, moraćete da preuzmete i instalirate alatke koje to omogućavaju. Instaliranje ne traje predugo, zavisno od brzine kojom ste povezani sa mrežom. U nekoliko slučajeva moraćete da sednete i sačekate dok se nešto preuzme sa interneta i instalira. Dok se Visual Studio preuzme i instalira, verovatno ćete imati dovoljno vremena da sredite neke druge stvari po kući. Uzgred, mala napomena: veoma je važno da

sve postupke obavite po redosledu koji navodim. Dobra vest je da ovo instaliranje treba da uradite samo jednom za svaki računar koji želite da koristite.

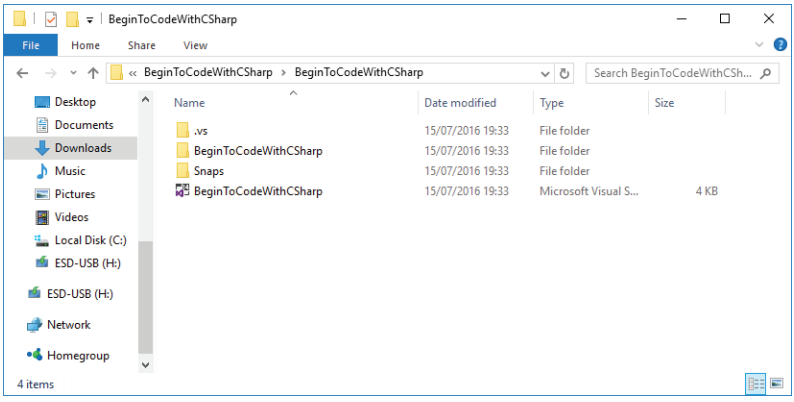
Postupak za instaliranje programa Visual Studio se povremeno menja, a može da zavisi i od toga da li imate Microsoftov nalog, a i od nekih drugih činilaca. Umesto da ovde navodim detaljni postupak, te informacije sam postavio onlajn (tako da mogu da ih ažuriram po potrebi).

Ako već niste preuzeli primere koda i preostali sadržaj opisan u odeljku „Preuzimanje" u uvodu ove knjige, idite sada na sledeću veb stranicu i preuzmite odgovarajuće datoteke:

https://aka.ms/BeginCodeCSharp/downloads

Otvorite datoteku pod nazivom GettingStarted.pdf i pratite uputstva iz nje da biste instalirali Visual Studio Community 2015 i raspakovali i podesili primere koda i aplikacija.

Pošto završite instaliranje, tek toliko da biste proverili da li ste spremni za početak, koristeći File Explorer, otvorite fasciklu sa primerima koda. Potrebno je da vidite datoteke i fascikle prikazane na **slici 1-1**. Sada ste spremni da otvorite Visual Studio i počnete pisanje koda jezikom C#, što ćemo i uraditi u narednom odeljku.



Slika 1-1 Sadržaj fascikle BeginToCodeWithCSharp. Odavde ćete uskoro početi.

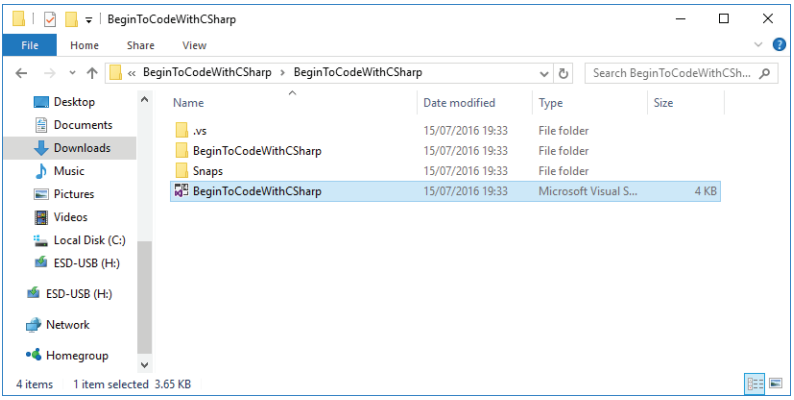
Korišćenje alatki

Došli ste do značajnog mesta u postupku učenja kako se programira. Spremni ste da otvorite Visual Studio i počnete rad sa primerima koda iz ove knjige. Ovo podeća na otvaranje ulaznih vrata novog stana ili kuće ili ulaska u novi sjajni auto koji ste upravo kupili.

Visual Studio projekti i rešenja

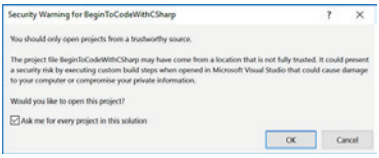
Kao što ćete uskoro saznati, Visual Studio organizuje svoje programerske radove kao *projekte* (engl. *projects*) i *rešenja* (engl. *solutions*). Kada Visual Studio koristite za razvijanje aplikacija, programa, stranica na internetu, veb aplikacija, komandnih datoteka, pomoćnih programa ili bilo čega drugog, pravite nov projekat. Projekat je skup resursa (datoteka sa kodom, slika i tako dalje) koji se koriste u programu koji razvijate. Kada pravite određeni projekat, Vusual Studio, takođe, pravi rešenje i taj projekat uključuje u to rešenje. Neko rešenje može da obuhvata samo jedan projekat, ali možete da uključite i druge druge projekate u to rešenje, onda kada su vašem programu neophodni resursi koje drugi projekti sadrže. Visual Studio udružuje sve projekte u nekom rešenju da bi to rešenje radilo.

Uz Visual Studio se automatski pridružuju datoteke sa rešenjem (datoteke sa oznakom tipa .sln) na sličan način na koji se programu za obradu teksta pridružuju datoteke dokumenata. To znači da kada otvorite rešenje **BeginToCodeWithCSharp**, koje je izabrano na **slici 1-2**, automatski se otvaraju okruženje Visual Studio i odgovarajuće rešenje. Samo napred, otvorite ovo rešenje sada (dvostrukim pritiskom mišem, na primer).



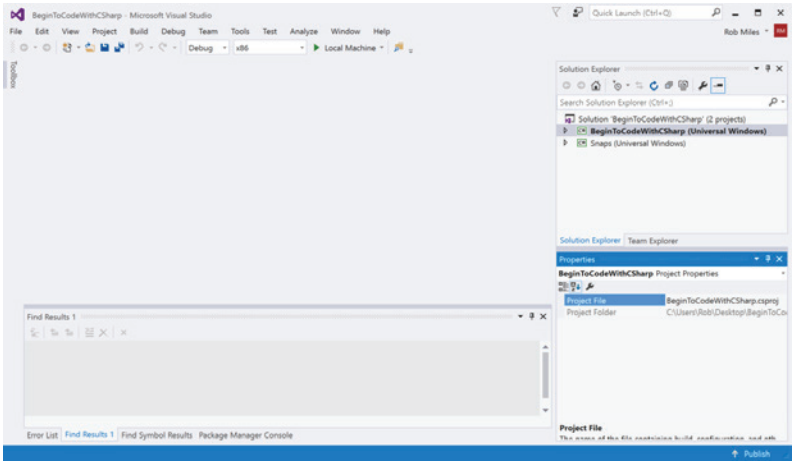
Slika 1-2 Otvaranje rešenja dvostrukim pritiskom miša na datoteku rešenja

Visual Studio je nepoverljiva vrsta programa koja baš potpuno ne veruje programima koji su preuzeti sa interneta, tako da traži da potvrdite da su projekti u određenom rešenju u redu, kao što možete da vidite na **slici 1-3**. U ovom slučaju, projekti su dobri (uostalom, ja sam ih napisao), tako da birate OK.



Slika 1-3 Visual Studio želi da bude siguran da verujete datoteci koju ste izabrali da otvorite

Visual Studio nudi *Solution Explorer*, alatku koju možete da koristite za pretraživanje projekata u nekom rešenju i pronalaženje datoteka u određenom rešenju i njegovim projektima. Solution Explorer pruža zgodan prikaz rešenja i projekata, pa počnimo da ga koristimo. Na **slici 1-4** prikazano je šta bi trebalo da vidite kada Visual Studio otvori rešenje **BeginToCodeWithCSharp**. (Prikaz na vašem ekranu može neznatno da se razlikuje zavisno od toga koje ste opcije instalirali.)



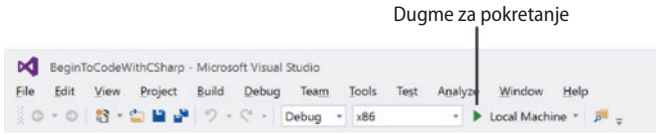
Slika 1-4 Glavna stranica u programu Visual Studio. Solution Explorer je na vrhu desno ove stranice.

Ovo rešenje sadrži dva projekta. Prvi projekat je **BeginToCodeWithCSharp**, a drugi projekat je **Snaps**, koji sadrži skup alatki koje se koriste za aplikacije koje služe kao primer. (Projekat **Snaps** nudi neke mogućnosti koje mogu da se koriste za bilo koji program, uključujući i one koje ćete kasnije praviti. Malo kasnije ćete saznati nešto više o biblioteci Snaps.)

Pokretanje programa u okruženju Visual Studio

Visual Studio se naziva *integrisano razvojno okruženje* (engl. *integrated development environment* ili IDE). To je mesto gde programeri ne samo da pišu svoje programe, već i posmatraju kako se njihovi programi izvršavaju. Počnimo sa upoznavanjem okruženja Visual Studio tako što ćemo pokrenuti neke od primera aplikacija koje su napravljene primerima koda koje ste preuzeli sa interneta.

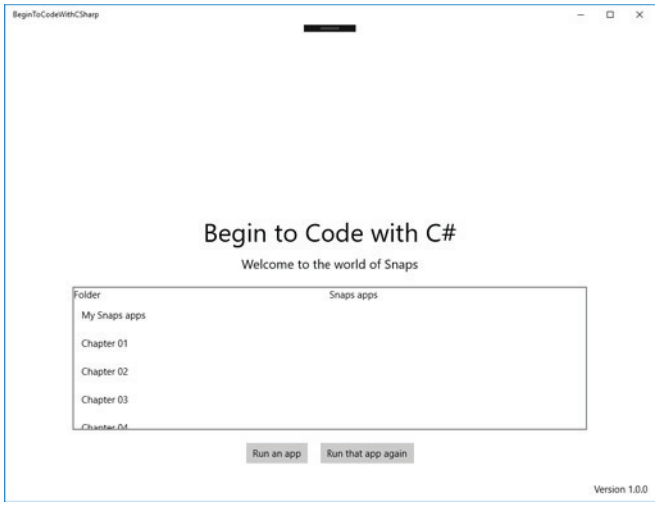
Kada želite da nekom programu prepustite upravljanje, kažete programu Visual Studio da ga pokrene koristeći dugme za pokretanje (engl. *run*), a to je dugme sa zelenom strelicom koje sam naznačio na **slici 1-5** na sledećoj stranici.



Slika 1-5 Dugme za pokretanje u prozoru Visual Studio pokreće programe na kojima radite

Kada za pokretanje neke aplikacije koristite ovo dugme, Visual Studio u suštini radi dve stvari. Prvo, pravi aplikaciju iz komponenata kojima se upravlja unutar odgovarajućeg rešenja. Ovaj postupak se naziva *građenje aplikacije*. Pošto se određena aplikacija izgradi, Visual Studio joj predaje kontrolu i pušta da se izvršava. Za naše rešenje, za postupak građenja aplikacije, potrebna je veza sa internetom kako bi radilo.

Samo napred i pritisnite dugme za pokretanje da biste pokrenuli aplikaciju. Visual Studio prikazuje prozor „Begin to Code with C#" („Početak programiranja jezikom C#") koji je prikazan na **slici 1-6**. Ako želite, možete da pomerate ovaj prozor po ekranu, umanjite ga ili prikažete na čitavom ekranu korišćenjem dugmeta za uvećanje (kvadratić) u gornjem desnom uglu.



Slika 1-6 Aplikacija BeginToCodeWithCSharp je pokrenuta

Rešenje **BeginToCodeWithCSharp** je aplikacija koju sam napravio za vas da biste mogli da se krećete između primera aplikacija iz ove knjige. Drugim rečima, svi programi iz knjige koji služe za pokazivanje su obuhvaćeni ovim rešenjem kao zasebne aplikacije, a pokretanjem ovog rešenja možete da birate poseban primer aplikacije koji želite da pokrenete. Neke od aplikacija su gotovi programi koje možete da koristite (ili čak prodate na lokaciji Windows Store), a druge koristimo za jednostavno prikazivanje izvesnih programerskih trikova koje ćemo zajedno kasnije razmatrati.

Dugmad pri dnu prozora su kontrole za pokretanje tih primera aplikacija. Da biste pokrenuli neku aplikaciju, prvo je izaberite korišćenjem okna za pronalaženje aplikacija iznad ovih dugmadi. U levom panelu, označenom sa **Folder** (fascikla), birate fasciklu odgovarajuće aplikacije, od kojih je većina nazvana po naslovima poglavlja iz knjige. U desnom panelu birate aplikaciju iz te fascikle. Kada izaberete dugme **Run an app** (pokreni aplikaciju), aplikacija koju ste izabrali se pokreće.

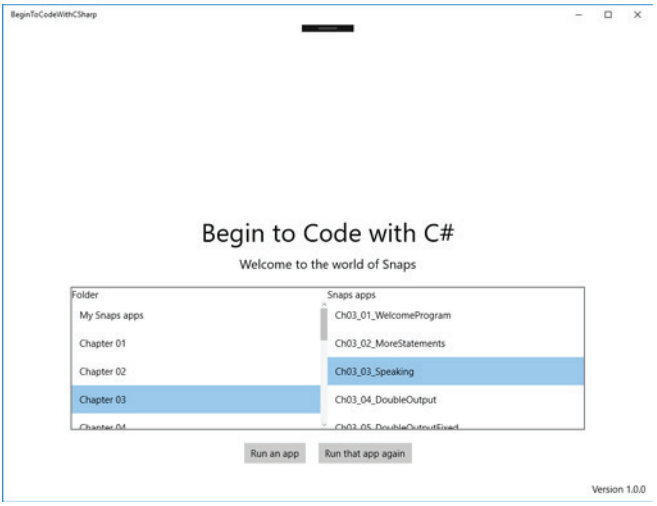


NEKA SE NEŠTO DEŠAVA

Biranje i pokretanje aplikacije

Ovo je prvi izdvojeni tekst „Neka se nešto dešava" – dobro došli! Ponekad ću u ovim izdvojenim tekstovima tražiti od vas da nešto uradite, a drugi put ću jednostavno tražiti da nešto isprobate. U svakom slučaju, radićete ono što programeri rade. Ovog puta želim samo da izaberete i pokrenete neku od aplikacija iz fascikle **BeginToCodeWithCSharp**. (Da, ovo je najjednostavniji „Neka se nešto dešava" u ovoj knjizi.)

Proverite da li je određeno rešenje pokrenuto. Izaberite neku fasciklu iz levog okna i primer aplikacije u desnom oknu. Postoji više aplikacija između kojih možete da birate. Možete da počnete pregledanjem fascikle **Chapter 03** za aplikaciju **Ch_03_03_Speaking**, koja omogućava da se vaš računar predstavi glasom.

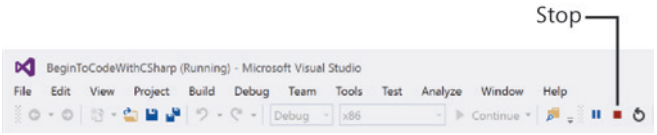


Izaberite **Run an app** i aplikacija koju ste izabrali se pokreće. Kada se aplikacija završi, okna za pretraživanje se ponovo pojavljuju. Izaberite i pokrenite još neku aplikaciju. Ako ste ljubitelj računarskih igrica, pogledajte aplikaciju **Ch_13_08_KeepUpGame** iz fascikle **Chapter 13**.

Zaustavljanje programa pokrenutih u prozoru Visual Studio

Pošto završite sa isprobavanjem nekih aplikacija, potrebno je da zaustavite program **BeginToCodeWithCSharp**. Ovo je neophodno pošto Visual Studio neće dozvoliti da menjate sadržaj programa dok je još uvek pokrenut. (Da je to moguće, ličilo bi na popravljanje aviona koji je još uvek u vazduhu.) I da, spremite se za pisanje koda kojim ćete menjati program!

Kada želite da zaustavite program, jednostavno zatvorite prozor koristeći dugme za zatvaranje u gornjem desnom uglu (znak X), na isti način na koji to radite sa bilo kojom aplikacijom. Međutim, Visual Studio, takođe, nudi dugme (naznačeno na **slici 1-7**) koje možete da koristite za zaustavljanje pokrenutog programa. Ovo koristite u neželjenim slučajevima kada se vaš program „zaglavi" zbog nečega.



Slika 1-7 Koristite ovo dugme za zaustavljanje pokrenutih programa u prozoru Visual Studio

Ponavljanje aplikacije

Dugme **Run that app again** (pokreni ovu aplikaciju ponovo) se koristi da biste ponovo pokrenuli poslednje pokretanu aplikaciju. Naziv poslednje aplikacije se pamti čak i pošto isključite računar i kasnije se vratite na rešenje **BeginToCodeWithCSharp**.



ŠTA BI MOGLO DA POĐE NAOPAKO

„Zaglavljivanje" aplikacije

Dobro došli u naš prvi izdvojeni tekst „Šta bi moglo da pođe naopako"! Ovde ćemo razmatrati moguće zamke na koje možete da naiđete dok radite sa kodom. U ovom izdvojenom tekstu namera nam je da razmotrimo način na koji možete da se krećete između primera aplikacija.

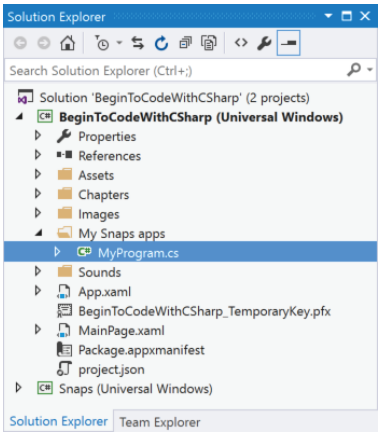
Neke od aplikacija su jednostavni prikazi kako se nešto radi i one se pokrenu i zatim završe, omogućavajući da izaberete i pokrenete drugu aplikaciju iz glavnih panela za pretraživanje aplikacija. Druge aplikacije su predviđene da rade stalno, baš kao i „prave" aplikacije. Na primer, možda ste već ustanovili da ne postoji način da zaustavite igricu Keep Up!; program je predviđen tako da radi neprestano.

Ako se „zaglavite“ u nekoj aplikaciji i želite da pokrenete neku drugu, jednostavno zaustavite rešenje **BeginToCodeWithCSharp** unutar prozora Visual Studio (kao što je pokazano ranije) ili izaberite X u gornjem desnom uglu aplikacije koja se izvršava, da biste je zaustavili. Mada sam vam malopre već pokazao kako da zaustavite neki program, želeo sam da ovde umetnem izdvojeni tekst „Šta bi moglo da pođe naopako“, kako bih vam pokazao da ću pratiti kako napredujete u učenju.

Aplikacija MyProgram

Kao što ste videli, kada prvi put pokrenete rešenje **BeginToCodeWithCSharp**, aplikacija prikazuje poruku dobrodošlice: „Welcome to the world of Snaps.“ („Dobro došli u Snaps svet.“) Ovu poruku prikazuje program koji je, naravno, ugrađen u rešenje. Pogledajmo C# kôd koji ovo radi.

Visual Studio omogućava da upravljate programima koje pišete na isti način na koji operativni sistem Windows omogućava da upravljate datotekama. Kada napravite neki program, obično zajedno skupite više različitih delova, kao što možete da vidite na **slici 1-8**. Na primer, savremene aplikacije sadrže zvukove i slike, a sve te stavke je potrebno čuvati zajedno tako da mogu da se iskoriste za građenje gotovog programa.

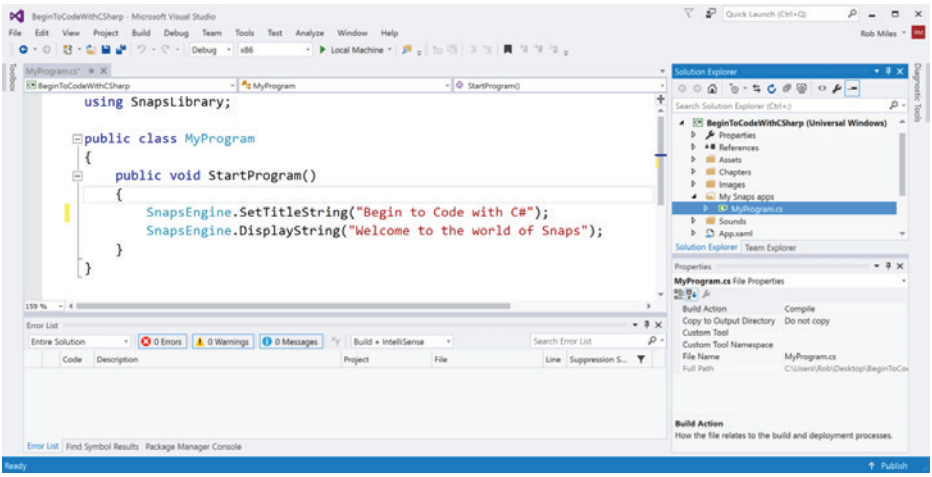


Slika 1-8 Izvorna datoteka MyProgram.cs i ostali resursi programa u prozoru Solution Explorer

U programu Visual Studio prozor Solution Explorer pomaže programerima da upravljaju raznim elementima koji sačinjavaju gotov program. Solution Explorer možete da zamislite kao posebnu vrstu pretraživača datoteka. On nudi prikaz fascikle koja sadrži sve datoteke koje se koriste kada se određena aplikacija izgrađuje i pokreće u programu Visual Studio. Projekat **BeginToCodeWithCSharp** sadrži više fascikli u kojima se nalaze različite datoteke. Kroz elemente i fascikle u nekom rešenju možete da se krećete pritiskom mišem na strelicu neposredno pored određene stavke.

Kasnije u ovoj knjizi, podrobnije ćemo zaviriti u sadržaj ovih fascikli, ali za sada vam skrećem pažnju na izvornu datoteku **MyProgram.cs**, koja se nalazi unutar fascikle **My Snaps apps** i koja je naznačena na **slici 1-8**. Ova datoteka sadrži kôd programa koji se izvršava kada se pokrene rešenje **BeginToCodeWithCSharp**.

Dvostruki pritisak mišem na tu datoteku u prozoru Solution Explorer dovodi do toga da se njen kôd pojavi u prozoru za obradu (editoru) programa Visual Studio, prikazanom na **slici 1-9**.



Slika 1-9 Visual Studio prikazuje sadržaj datoteke MyProgram.cs u prozoru za obradu

Ono što radite je slično otvaranju dokumenta u programu za obradu teksta, ali umesto da dobijete niz reči neke priče (ili pesme ili izveštaja), ugledaćete niz naredbi koje računar sledi kada izvršava određeni program – ili, recimo to drugim rečima, istražujete kôd programa. Ono što vidite na **slici 1-9** je pravi C#, stoga – čestitamo! – upravo gledate svoj prvi deo koda u jeziku C#.

SAVET PROGRAMERA

Programski jezici nisu ništa posebno

Ako ste mislili da su programski jezici nešto izuzetno složeno i teško za razumevanje, grešite. Mislim da bi većina ljudi (onih koji umeju da čitaju na engleskom jeziku) lako shvatila da program prikazan na **slici 1-9** prikazuje poruke „Begin to Code with C#" i „Welcome to the world of Snaps".

Deo jasnoće programa dolazi iz jezika C#, koji je napravljen tako da se lako prati, a ostalo potiče od pažljivog biranja naziva kojima se opisuju komponente programa. Mogao sam da koristim „xyzzy" umesto SetTitleString kao naziv za ponašanje koje prikazuje naslovnu poruku u prozoru. Računaru je sasvim svejedno kako nešto zovem sve dok sam dosledan i dok on može da ih jasno razlikuje.

Međutim, nisam pisao program samo za računar. Pisao sam ga i za početnike poput vas, koji će proučavati ovaj kôd da bi shvatili šta program radi i da bi potom počeli da pišu sopstvene programe.



NEKA SE NEŠTO DEŠAVA

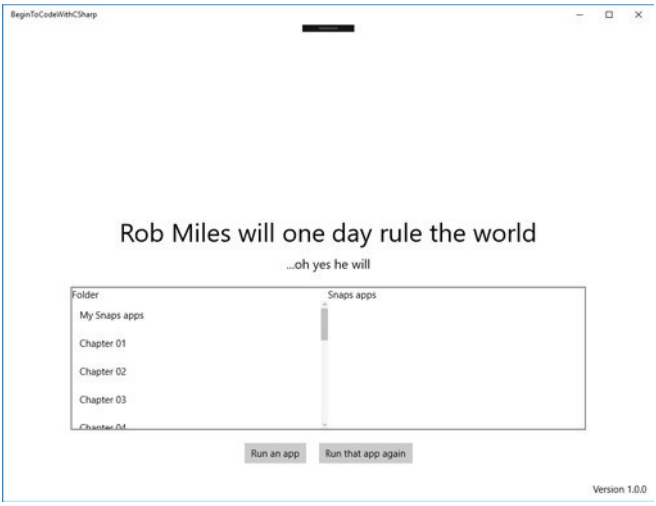
Menjanje poruka

Pritisnite mišem strelice u prozoru Solution Explorer da biste otvorili projekat **BeginToCodeWithCSharp** i fasciklu **My Snaps apps**. Dva puta brzo pritisnite mišem datoteku **MyProgram.cs** da biste je otvorili za menjanje (ako već nije otvorena). Možete da napravite svoj prvi program menjanjem ovog programa tako da radi neznatno drugačije. Kôd programa koji smo pokrenuli (koji, takođe, možete da vidite na **slici 1-9**) izgleda ovako:

```
public class MyProgram
{
  public void StartProgram()
  {
  SnapsEngine.SetTitleString("Begin to Code With C#");
  SnapsEngine.DisplayString("Welcome to the world of Snaps");
  }
}
```

Visual Studio je posebno napisan tako da različite delove koda prikazuje u različitim bojama. Uobičajeno je da se tekst koji će biti prikazivan na ekranu kada se program pokrene – ovaj tekst se označava kao *string* – pojavljuje kao crveni tekst u kodu programa. Tekst stringa ne morate da bojite u crveno; ovo se dešava automatski kada je odgovarajući string ispravno omeđen u kodu. (Više o tome malo kasnije.) Promenite nešto u ovim stringovima, kao što sam ja to uradio niže, ne menjajući ništa drugo u preostalom kodu. Zatim, ponovo pokrenite ovaj program koristeći dugme za pokretanje u prozoru Visual Studio. Poruke na ekranu će se promeniti tako da odražavaju izmene koje ste napravili u kodu. Snimak ekrana na sledećoj stranici, na primer, pokazuje rezultate mojih izmena.

```
public class MyProgram
{
  public void StartProgram()
  {
  SnapsEngine.SetTitleString("Rob Miles will one day rule the world");
  SnapsEngine.DisplayString("...oh yes he will");
  }
}
```



Budite pažljivi dok menjate tekst tako da ne uklonite znake navodnika (") koje označavaju početak i kraj stringa u programu. Ako ih uklonite, videćete da taj tekst više nema smisla unutar C# programa i dobićete grešku kada pokušate da pokrenete program. Ako se to desi, ne brinite: editor u programu Visual Studio ima izuzetne mogućnosti za poništavanje izmena koje ste napravili u nekoj datoteci. Jednostavno, držite pritisnut taster **Ctrl** i pritiskajte taster **Z** da biste jednu za drugom poništili izmene u datoteci. Ako pritisnete **Ctrl+Z** nekoliko puta, dok ste u editoru (prozor u kome se pojavljuje kôd), na kraju ćete se vratiti u prvobitno stanje programa.

Upravo ste napisali (ili bar izmenili) svoj prvi program. Sada, ako se od vas traži da napravite nešto što može da prikazuje poruku na ekranu, imate bar delimičnu ideju o tome kako da to uradite. Ja ću, naravno, objasniti ostale razne aspekte ovog koda koji ste upravo prepravljali, kako bi pomogao da ga potpuno razumete!

SAVET PROGRAMERA

Ne postoji nešto kao „profesionalni" program

Ono što ste ovde uradili – pošto ste promenili datoteku **MyProgram.cs** – je „pravi" program. Ako želite, mogli biste da iskoristite alatke ugrađene u Visual Studio da ovaj program pošaljete u Windows Store tako da bilo ko u svetu može da ga preuzme i koristi (mada, iskreno, nisam baš siguran da bi iko rekao za njega da je posebno koristan). Ljudi koji uče programiranje se ponekad pitaju kada će stići do tačke gde će biti dobri u programiranju kao „profesionalni" programeri. Odgovor je da čim neko plati za to što pišete programe, vi ste profesionalni programer.

Naravno, to što neko plaća da nešto uradite ne znači, automatski, da ste bolji u tome, ali u svakom slučaju bi to trebalo da ukaže na to da vaš trud nije uzaludan.

Ako želite da vam ljudi daju pare za vaše programe, neophodno je da budete sigurni u to da su vaši programi toga vredni. Kroz čitavu knjigu, razmatraćemo primere dobre progamerske prakse tako da, kada neko kaže: „Platio bih za to" kada vidi neki od vaših programa, daćete mu nešto što je kvalitetno i vredno.

Šta ste naučili

U ovom prvom poglavlju sagradili ste svoje mesto za rad, instalirali alatke programa Visual Studio koje nameravate da koristite za pisanje programa i pregledali neke od primera aplikacija koje se daju uz knjigu.

Otkrili ste da je Visual Studio u suštini „program za obradu teksta za programere", mesto gde programeri mogu da pišu i proveravaju svoj softver. Videli ste i da Visual Studio koristi rešenja i projekte za organizovanje resursa i kôd programa koji se skupljaju zajedno tako da čine savremenu aplikaciju. I napravili ste svoju prvu aplikaciju menjanjem poruke koju program prikazuje.

Da biste potvrdili da ste dobro razumeli ovo poglavlje, možda je potrebno da razmotrite sledeća pitanja o računarima, programima i programiranju.

U čemu je razlika između programa i aplikacije?

Kada ljudi pričaju o softveru, primetićete da se reči *program* i *aplikacija* koriste uporedo. Kada pričam o programu, opisujem kôd koji saopštava računaru šta da radi. Aplikaciju posmatram kao nešto veće i razvijenije. Aplikacija sa sobom nosi zajedno i kôd programa i prateće datoteke kao što su slike i zvuci, kako bi korisnik imao potpuni doživljaj. Program može da se sastoji od samo nekoliko redova u jeziku C#.

U čemu je razlika između projekta i rešenja u okruženju Visual Studio?

Rešenje je spoljašnji sadržalac. Rešenje može da sadrži projekte i često se koristi za pravljenje gotove aplikacije ili proizvoda. *Projekat* može da sadrži C# izvorne datoteke i obično je potpuna manja celina nekog rešenja. Na primer, sve C# datoteke programa koje sačinjavaju radno okruženje Snaps spakovane su kao projekat – projekat **Snaps** u rešenju **BeginToCodeWithCSharp** – tako da se mogu koristiti u bilo kom rešenju za koje je potrebno korišćenje Snaps resursa. Resurse u projektu **Snaps** koristićete u ovoj knjizi veoma često kada počnete da pišete kôd za svoje aplikacije.

Zašto nam je potreban poseban jezik kao što je C# da bismo programirali na računaru?

Moj omiljeni odgovor na ovo pitanje je par rečenica: „Vreme leti kao strela. Voćne mišice vole banane*". Ljudsko biće lako može da shvati da se prva odnosi na nešto

* Igra rečima ***flies*** (u značenju *leteti*, ali i *mušice*) i *like* (u značenju *kao*, ali i *voleti*) u engleskom jeziku. U originalu "Time flies like an arrow. Fruit flies like a banana." (nap.prev.)

što leti, dok je druga rečenica o insektima i o tome šta oni vole da jedu. Računar bi se grdno namučio dok bi shvatio ispravno značenje ove dve rečenice. Način na koji ljudi koriste jezik je nabijen dvosmislenostima i nejasnoćama. Srećom, imamo zaista moćan računar između naših ušiju koji je isprepletan nervima za jezik, a svoje prve godine proveli smo programirajući ga. Nasuprot tome, jadni računari imaju veoma jednostavnu mašinu za razmišljanje koja radi najbolje samo kad joj se zadaju stroga i čvrsta pravila. Programski jezik se sastoji od niza posebno određenih konstrukcija koje omogućavaju da ih računar razume tako da može da na pravi način sledi naša uputstva.

Da li je Visual Studio jedini način za pisanje programa?

Ne. Postoji više sjajnih alatki koje možete da koristite za pisanje softvera. Neke su usko vezane za određene programske jezike, a druge imaju opštiju namenu. Međutim, Visual Studio je jedan od najboljih.

Šta da radim ako upropastim neki program?

Neke ljude brine to da ono što rade sa nekim programom na računaru može da ga „pokvari" na neki način. I ja sam nekad o tome brinuo, ali sam prevladao taj strah time što sam se obezbedio da kad god da uradim nešto uvek imam način da to popravim. Vi ste, trenutno, u tom srećnom položaju. Znate kako da bez greške postavite Visual Studio na računar, a koristite kopiju primera koda koji ste raspakovali iz .zip fascikle koju ste preuzeli sa interneta. Čak i da nešto krene zaista loše i pokvarite neki program tako da više ne može da radi, sve što je potrebno jeste da raspakujete novu kopiju sa primerima koda iz .zip fasikle i počnete ispočetka.