

Michael Halvorson

Microsoft®

Visual Basic

**krok za
krokem**

- Základy programování ve Visual Basicu a Visual Studiu
- Názorný výklad na konkrétních příkladech bez zbytečné teorie
- Tvorba webových stránek a práce s databázemi
- Vývoj aplikací pro Windows Phone

computer
press®

Microsoft®

Michael Halvorson

Microsoft Visual Basic

Krok za krokem

Computer Press
Brno
2015

Microsoft Visual Basic

Krok za krokem

Michael Halvorson

Překlad: Milan Daněk

Obálka: Martin Sodomka

Odpočívající redaktor: Martin Herodek

Technický redaktor: Jiří Matoušek

Authorized Czech translation of the English edition of Microsoft Visual Basic 2013 Step by Step, ISBN 9780735667044 © 2013 Michael Halvorson.

This translation is published and sold by permission of O'Reilly Media, Inc., which owns or controls all rights to sell the same.

Translation © Milan Daněk, 2015

Objednávky knih:

<http://knihy.cpress.cz>

www.albatrosmedia.cz

eshop@albatrosmedia.cz

bezplatná linka 800 555 513

ISBN 978-80-251-4412-1

Vydalo nakladatelství Computer Press v Brně roku 2015 ve společnosti Albatros Media a. s. se sídlem Na Pankráci 30, Praha 4. Číslo publikace 19243.

© Albatros Media a. s., 2015. Všechna práva vyhrazena. Žádná část této publikace nesmí být kopírována a rozmnožována za účelem rozšiřování v jakékoli formě či jakýmkoli způsobem bez písemného souhlasu vydavatele.

1. vydání

 **ALBATROS** MEDIA a.s.

Stručný obsah

Část I

Úvod do vývoje v prostředí Visual Studio	25
1. Možnosti vývoje v jazyce Visual Basic a Windows Store	27
2. Integrované vývojové prostředí Visual Studio	41
3. Vytvoření první aplikace pro Windows Store	67
4. Aplikace pro Windows desktop: stručný úvod do používání Windows Forms	99

Část II

Návrh uživatelského rozhraní	127
5. Práce s ovládacími prvky aplikací pro Windows Store	129
6. Práce s ovládacími prvky pro Windows Forms	163
7. Značkový jazyk XAML krok za krokem	205
8. Používání stylů XAML	227
9. Seznámení se základními prvky návrhu pro Windows 8.1: řádek nabídek, informační rámeček, dlaždice a dotyk	245
10. Vytváření konzolových aplikací	277

Část III

Techniky programování ve Visual Basicu	299
11. Datové typy, operátory a práce s řetězci	301
12. Vytváření rozhodovacích struktur a smyček	349
13. Zachytávání chyb pomocí strukturovaného zpracování chyb	381
14. Používání polí, kolekcí a generických typů pro práci s daty	401
15. Inovativní práce s daty, založená na jazyce LINQ	435
16. Techniky objektově orientovaného programování	457

Část IV

Databázové a webové programování **483**

- | | | |
|-----|--|-----|
| 17. | Ovládací prvky databází pro aplikace, určené pro Windows desktop | 485 |
| 18. | Přístup k datům z aplikací, navržených pro Windows Store | 513 |
| 19. | Vývoj webových aplikací v prostředí Visual Studio s využitím ASP.NET | 541 |

Část V

Programování pro Windows Phone **581**

- | | | |
|-----|--|-----|
| 20. | Úvod do vývoje pro Windows Phone 8 | 583 |
| 21. | Vytvoření první aplikace pro Windows Phone 8 | 601 |

Rejstřík **635**

Obsah

O autorovi	15
Úvod	17
Komu je tato kniha určena	18
Doporučení	18
Komu tato kniha naopak určena není	19
Uspořádání knihy	19
Jak najít optimální místo pro zahájení studia této knihy	20
Konvence použité v této knize	20
Systémové požadavky	21
Ukázkové kódy	22
Instalace ukázkových kódů	22
Práce s ukázkovými kódy	22
Poděkování	22
Zpětná vazba od čtenářů	23
Errata	23

ČÁST I

Úvod do vývoje v prostředí Visual Studio

Kapitola 1

Možnosti vývoje v jazyce Visual Basic a Windows Store	27
Řada produktů Visual Basic 2013 a možnosti jejich využití	28
Obrovský rozsah příležitostí a platforem pro vývoj	29
Multiplatformní přístup k výuce Visual Basicu	31
Seznámení s Windows Store	32
Co je Windows Store?	32
Přístup na Windows Store	32
Prodejní informace a cenové hladiny	34
Anebo můžete svoji aplikaci nabízet zdarma...	34
Předem myslíte i na certifikaci	35
Kontrolní seznam požadavků Windows Store	36
Vše závisí na detailech	39
Shrnutí	39

Kapitola 2

Integrované vývojové prostředí Visual Studio	41
Zahájení práce	42
Vývojové prostředí sady Visual Studio	43
Důležité nástroje IDE	46
Uspořádání nástrojů v IDE	48
Návrhář a značkovací jazyk XAML	48
Spouštění a testování aplikací pro Windows Store	54
Práce s oknem Vlastnosti (Properties)	56
Uspořádání nástrojů pro programování	60
Přesun a ukotvení nástrojů	60
Skrývání oken nástrojů	62
Základní konfigurace IDE pro cvičení uvedená v této knize	62
Ukončení práce v sadě Visual Studio	65
Shrnutí	66

Kapitola 3

Vytvoření první aplikace pro Windows Store	67
Šťastná sedma: aplikace napsaná ve Visual Basicu pro Windows Store	68
Programování krok za krokem	68
Návrh uživatelského rozhraní	69
Konečná nastavení vlastností a další úpravy	82
Psaní kódu	84
Rozbor kódu zpracovatele události TlacitkoVytoc_Click	88
Spouštění aplikací navržených pro Windows Store	89
Vytvoření spouštěcí obrazovky aplikace	90
Sestavení spustitelného souboru	94
Shrnutí	97

Kapitola 4

Aplikace pro Windows desktop: stručný úvod do používání Windows Forms	99
Uvnitř aplikací pro Windows desktop	100
Visual Basic a aplikace pro Windows desktop	101
Vytvoření aplikace pro Windows desktop	103
Nastavování vlastností	112
Vlastnosti okénka pro zobrazování obrázku	116
Zvýšení srozumitelnosti díky pojmenovávání objektů	117
Příprava kódu	117

Rozbor kódu zpracovatele události TlacitkoVytoc_Click	120
Spuštění aplikace Šťastná sedma	121
Sestavení spustitelného souboru	122
Publikování aplikace pro Windows desktop	123
Shrnutí	125

ČÁST II

Návrh uživatelského rozhraní

Kapitola 5

Práce s ovládacími prvky aplikací pro Windows Store	129
Seznámení s ovládacími prvky aplikací pro Windows Store	129
Kořeny v technologii Windows Presentation Foundation a XAML	130
Základy návrhu pro systém Windows 8.1	131
Použití ovládacího prvku typu TextBox k načtení vstupu od uživatele	131
Přiřazení obsahu prvku TextBox proměnné	136
Víceřádkové ovládací prvky typu TextBox	138
Kontrola pravopisu v ovládacím prvku typu TextBox	142
Použití ovládacího prvku typu FlipView k zobrazení série obrázků	145
Použití ovládacího prvku typu MediaElement k přehrání multimediálního obsahu	150
Použití ovládacího prvku typu WebView k zobrazení živého webového obsahu	158
Shrnutí	162

Kapitola 6

Práce s ovládacími prvky pro Windows Forms	163
Použití ovládacího prvku typu DateTimePicker	164
Ovládací prvky pro zpracování vstupu uživatele	169
Použití ovládacího prvku typu CheckBox	169
Použití ovládacích prvků typu GroupBox a RadioButton	174
Použití seznamu ke zpracování vstupu	179
Vytváření nabídek pomocí ovládacího prvku typu MenuStrip	184
Práce s nabídkami	184
Přiřazení kláves rychlého přístupu příkazům nabídky	186
Zpracování výběrů provedených v nabídce	189
Vytváření panelů nástrojů pomocí ovládacího prvku typu ToolStrip	193
Práce s dialogovými okny	196
Zpracovatelé událostí pro práci s běžnými dialogovými okny	198
Shrnutí	203

Kapitola 7

Značkovací jazyk XAML krok za krokem	205
Seznámení s jazykem XAML	206
Jazyk XAML v IDE sady Visual Studio	207
XAML a Blend for Visual Studio	207
Prvky XAML	208
Jmenné prostory v jazyce XAML	210
XAML soubory projektu podrobně	210
Přidávání XAML prvků pomocí okna Editor kódu (Code Editor)	216
Shrnutí	226

Kapitola 8

Používání stylů XAML	227
Seznámení se styly XAML	227
Kam se poděl soubor StandardStyles.xaml?	228
Vytváření nových stylů XAML	229
Zvážení rozsahu platnosti stylů	230
Ukázka kódu nového stylu XAML	230
Vytvoření odkazu na styl	231
Používání explicitních a implicitních stylů	232
Práce se styly XAML	233
Vytváření nových stylů ze stylů existujících	238
Zkrácené postupy IDE pro používání stylů	241
Shrnutí	242

Kapitola 9

Seznámení se základními prvky návrhu pro Windows 8.1: řádek nabídek, informační rámeček, dlaždice a dotyk	245
Vytvoření řádku nabídek k provádění běžných úkonů	246
Funkce řádku nabídek	247
Návrh vlastního řádku nabídek	248
Řádek nabídek v praxi, krok za krokem	249
Použití ovládacího prvku typu Flyout k načtení vstupu a zobrazení informací	253
Návrh vlastních dlaždic pro aplikace	258
Složka Assets	258
Požadované dlaždice a jejich využití	259
Programování živých dlaždic	266
Plánování dotykového vstupu	268
Ovládací prvky XAML podporují dotykový vstup automaticky	269
Často používaná gesta	269
Úvahy týkající se použitelnosti	271

Nastavení zabezpečení a oprávnění	272
Shrnutí	275

Kapitola 10

Vytváření konzolových aplikací	277
Konzolové aplikace v sadě Visual Studio	278
Vytvoření konzolové aplikace	278
Moduly a procedury	280
Procedura Sub Main()	281
Interaktivní matematické hry	285
Nalezení čísla	285
Simulace házení kostek	290
Sestavení, publikování a spouštění konzolových aplikací	295
Shrnutí	298

ČÁST III

Techniky programování ve Visual Basicu

Kapitola 11

Datové typy, operátory a práce s řetězci	301
Strategie pro deklarace proměnných a konstant	301
Příkaz Dim	302
Definice konstant	305
Doporučení pro vytváření názvů proměnných a konstant	306
Datové typy a ovládací prvek typu ListBox	307
Operátory a vzorce	314
Aritmetické operátory	314
Pokročilé aritmetické operátory	318
Zkrácené operátory	322
Jak Visual Basic vypočítává vzorce	322
Převody datových typů	324
Metoda ToString	324
Metoda Parse	325
Třída Convert	327
Starší typy funkcí pro převody datových typů a jejich použití	327
Zpracování řetězců pomocí třídy String	328
Často prováděné úkoly	328
Řazení textu	331
Práce s ASCII znaky	332

Řazení řetězců v textovém poli	333
Seznámení se zdrojovým kódem programu Řazení Textu	336
Ochrana obsahu souboru základním šifrováním	338
Použití operátoru Xor	342
Podrobné seznámení s kódem programu Šifrování Xor	345
Shrnutí	348

Kapitola 12

Vytváření rozhodovacích struktur a smyček	349
Událostmi řízené programování	350
Používání podmínkových výrazů	351
Rozhodovací struktury If...Then	352
Testování několika podmínek v rozhodovací struktuře If...Then	352
Používání logických operátorů v podmínkových výrazech	357
Zkrácené vyhodnocování pomocí operátorů AndAlso a OrElse	359
Práce s rozhodovacími strukturami Select Case	361
Použití operátorů porovnání ve struktuře Select Case	362
Práce se smyčkami For...Next	368
Použití smyčky k naplnění objektu typu TextBox řetězcovými daty	369
Složitější smyčky For...Next	371
Příkaz Exit For	375
Práce se smyčkami Do	375
Předcházení nekonečným smyčkám	376
Převody teplot	377
Používání klíčového slova Until ve smyčkách Do	380
Shrnutí	380

Kapitola 13

Zachytávání chyb pomocí strukturovaného zpracování chyb	381
Zpracování chyb pomocí příkazu Try...Catch	382
Kdy je vhodné použít kód zpracovatele chyb	382
Příprava pasti: blok kódu Try...Catch	383
Chybná cesta k souboru a chyby při přístupu k mechanice	384
Aplikace pro Windows Store a vestavěné zpracování výjimek	388
Tvorba kódu zpracovatele chyb pro načítání souborů z USB paměti	389
Použití klauzule Finally k provedení závěrečného „úklidu“	390
Složitější zpracování chyb pomocí Try...Catch	392
Třída Exception	392
Určení počtu opakování	395
Používání vnořených bloků Try...Catch	397

Porovnání zpracovatelů chyb s technikami defenzivního programování	398
Příkaz Exit Try	399
Shrnutí	400

Kapitola 14

Používání polí, kolekcí a generických typů pro práci s daty	401
Práce s poli proměnných	402
Vytvoření pole	402
Deklarace pole s předem známým počtem prvků	403
Vyhrazení paměti	404
Práce s prvky polí	405
Deklarace pole a přiřazení počátečních hodnot	406
Vytvoření pole pro uložení teplot	407
Metody GetUpperBound a GetLowerBound	408
Nastavení velikosti pole v době běhu	413
Zachování obsahu pole pomocí příkazu ReDim Preserve	417
Použití příkazu ReDim pro trojrozměrná pole	418
Práce s velkými poli s využitím metod třídy Array	419
Třída Array	419
Další testy s řazením	425
Práce s kolekcemi	426
Vytváření kolekcí a generických seznamů	426
Deklarace generických kolekcí	427
Ukázková aplikace využívající generický seznam a obrázek pozadí	428
Shrnutí	434

Kapitola 15

Inovativní práce s daty založená na jazyce LINQ	435
Nástroje a techniky jazyka LINQ	435
Základy syntaxe dotazů	436
Načítání informací z polí	437
Používání jazyka LINQ pro práci s kolekcemi	448
Používání jazyka LINQ s XML dokumenty	451
Shrnutí	456

Kapitola 16

Techniky objektově orientovaného programování	457
Zdědění formuláře pomocí nástroje Výběr dědičnosti (Inheritance Picker)	458
Vytváření vlastních bazových tříd	464
Přidání nové třídy do projektu	465

Dědění базové třídy	474
Polymorfismus	476
Syntaxe překrývání metod a vlastností	477
Odkazování na базovou třídu pomocí MyBase	478
Experimentování s polymorfismem	478
Shrnutí	482

ČÁST IV

Databázové a webové programování

Kapitola 17

Ovládací prvky databází pro aplikace určené pro Windows desktop 485

Databázové programování s využitím ADO.NET	486
Základní databázová terminologie	487
Práce s databází Microsoft Access	488
Okno Zdroje dat (Data Sources)	497
Použití ovládacích prvků panelu nástrojů k zobrazení informací z databáze	503
Příkazy jazyka SQL a filtrování dat	506
Shrnutí	511

Kapitola 18

Přístup k datům z aplikací navržených pro Windows Store 513

Vázání dat v XAML	514
Různé zdroje dat	514
Prvky vázání	514
Vázání ovládacího prvku s třídou	515
Použití kolekce jako zdroje dat	520
Přístup k datům uloženým v XML dokumentech	524
Čtení XML dokumentu	524
Vyhledávání prvků v XML dokumentu	531
Zápis dat do XML dokumentu	534
Uživatelské rozhraní pro zadávání dat	538
Shrnutí	539

Kapitola 19

Vývoj webových aplikací v prostředí Visual Studio s využitím ASP.NET 541

Seznámení s technologií ASP.NET	541
Webové formuláře	543
ASP.NET MVC	544

Stránky Web Pages se syntaxí Razor	545
HTML5 a JavaScript	547
Tvorba webu s využitím technologie ASP.NET a webových formulářů	549
Softwarové požadavky pro vývoj aplikací s využitím ASP.NET	549
Základní kroky	550
Webové stránky a formuláře Windows	550
Práce s oknem Návrhář (Designer)	555
Přidání serverových ovládacích prvků na webovou stránku	560
Tvorba kódu zpracovatelů událostí ovládacích prvků webových stránek	562
Úpravy šablony webu	568
Zobrazení záznamů z databáze na webové stránce	570
Úprava vlastností dokumentu a vzorové stránky	577
Shrnutí	579

ČÁST V

Programování pro Windows Phone

Kapitola 20

Úvod do vývoje pro Windows Phone 8	583
Příležitosti platformy Windows Phone 8	584
Klíčové funkce systému Windows Phone 8	585
Hardwarové požadavky	586
Integrace a spolupráce	586
Windows Phone Store	587
Co je Windows Phone Store?	587
Přístup k Windows Phone Store	588
Kolik si vývojáři vydělávají?	590
Předem uvažujte o získání certifikace	591
Práce se sadou Windows Phone SDK 8.0	592
Stažení balíčku Windows Phone SDK	593
Srovnání platforem Windows Phone 8 a Windows Store	595
Rozdíly	595
Podobnosti	598
Shrnutí	600

Kapitola 21

Vytvoření první aplikace pro Windows Phone 8	601
Vytvoření nového projektu aplikace pro Windows Phone	602
Návrh uživatelského rozhraní aplikace Golfový vozík	608

Příprava kódu	611
Testování aplikací pro Windows Phone	615
Životní cyklus aplikace	621
Uzavření či deaktivace?	621
Třída PhoneApplicationService	623
Řízení životního cyklu aplikace pomocí třídy IsolatedStorageSettings	630
Nastavování možností v souboru manifestu aplikace pro systém Windows Phone	631
Shrnutí	632

Rejstřík	635
-----------------	------------

O autorovi

Michael Halvorson je autorem či spoluautorem více než 35 různých knih, mezi něž patří například *Start Here! Learn Microsoft Visual Basic 2012*, *Microsoft Visual Basic 2010 Step By Step*, *Microsoft Office XP Inside Out* a *Microsoft Visual Basic 6.0 Step By Step*. Za svoji tvůrčí činnost v oblasti populárně naučné literatury získal mnoho ocenění, včetně Computer Press Best How-to Book Award (v kategorii Software) či Society for Technical Communication Excellence Award (v kategorii Writing). Michael Halvorson získal titul bakaláře v oboru počítačových věd na univerzitě Pacific Lutheran University v městě Tacoma, ležícím ve státě Washington. Svoje magisterská a posléze i doktorandská studia v oboru historie dokončil na University of Washington v městě Seattle. V letech 1985 až 1993 pracoval ve společnosti Microsoft a od prvního uvedení jazyka Visual Basic na výstavě Windows World v roce 1991 patří k velkým zastáncům programování v tomto jazyce. V současné době Michael Halvorson působí jako docent na univerzitě Pacific Lutheran University. Více informací o jeho knihách a myšlenkách najdete na webu <http://michaelhalvorsonbooks.com>.

Úvod

Microsoft Visual Basic 2013 je důležitým upgradem a rozšířením oblíbeného programovacího jazyka Visual Basic a jeho kompilátoru, představujících technologii, již se zabývají miliony programátorů po celém světě. Visual Basic 2013 není samostatným produktem, nýbrž klíčovou komponentou sady Microsoft Visual Studio 2013 – komplexního vývojového systému, umožňujícího vývoj výkonných aplikací pro Microsoft Windows 8.1, Windows desktop, web, Windows Phone 8 a celou řadu dalších prostředí.

Ať už si zakoupíte jednu z placených edicí sady Visual Studio 2013 anebo si zdarma stáhnete Visual Studio Express 2013 za účelem jeho vyzkoušení, čeká vás vynikající zkušenost. Poslední funkce jazyka Visual Basic zvýší vaši produktivitu a programátorskou zručnost, především pak tehdy, pokud rádi využíváte a integrujete informace z databází, různých médií pro zábavu či webových stránek. Kromě toho důležitým přínosem ovládnutí jazyka Visual Basic a integrovaného vývojového prostředí (*Integrated Development Environment – IDE*) sady Visual Studio je to, že většinu nástrojů budete moci využít i při psaní programů v jazycích Microsoft Visual C# 2013, Microsoft Visual C++ 2013, HTML5, JavaScript či dalších oblíbených jazycích.

Microsoft Visual Basic Krok za krokem představuje komplexní úvod do programování v jazyce Visual Basic, a to s využitím verze Visual Basic 2013 a operačního systému Windows 8.1. Při navrhování této praktické příručky jsem měl na mysli programátory s různými úrovněmi znalostí a zkušeností. Domnívám se totiž, že nejlepším způsobem pro ovládnutí tak složité technologie, jakou Visual Basic bezesporu je, je vycházet z předpokladu, že programátoři se nejlépe učí vlastní prací. Čtením této knihy a současným procházením jednotlivých příkladů si tedy osvojíte základní programátorské techniky, popsané v pečlivě připravených tutoriálech, které můžete dokončovat dle svých vlastních časových možností.

Ačkoliv mám značné zkušenosti s výukou na vysokých školách a řízením projektů ve velkých podnicích, tato kniha rozhodně není jen suchou učebnicí či referenční příručkou pro programátory; namísto toho se jedná o praktický programátorský tutoriál, který vás činí odpovědnými za vaše vlastní studium, milníky vašeho vývoje a následné úspěchy. Čtením této knihy se programátoři, kteří se s jazykem Visual Basic teprve seznamují, naučí základy vývoje v tomto jazyce v kontextu užitečných, reálných aplikací, zatímco středně pokročilí programátoři si rychle osvojí základní nástroje a techniky, které jim nabízí Visual Basic 2013 a systém Windows 8.1.

Při psaní této knihy jsem vycházel z multiplatformního přístupu, takže kromě toho, že se naučíte programovat v jazyce Visual Basic, se také dozvíte, jak se vytváří široká škála aplikací, počínaje aplikacemi pro Windows Store přes aplikace pro Windows desktop, využívající Windows Forms, konzolové aplikace, webové aplikace (ASP.NET) až po aplikace pro Windows Phone 8. Každý z těchto typů aplikací má v reálném vývoji a v reálném světě svoje místo a využití.

Tento komplexní přístup je doprovázen rozdělením do 5 tematicky uspořádaných částí, 21 kapitol a mnoha cvičeními a ukázkovými programy. Čtením této knihy se velmi rychle naučíte vytvářet v jazyce Visual Basic aplikace profesionální kvality, určené pro operační systém Windows, platformu Windows Phone 8 a řadu webových prohlížečů. A navíc se při tom všem budete ještě bavit!

Komu je tato kniha určena

Toto je příručka, která vás naučí programovat v jazyce Visual Basic krok za krokem: je tedy určena pro takové čtenáře, kteří se rádi učí nové věci tím, že je rovnou dělají. Při psaní knihy jsem vycházel z předpokladu, že již máte nějaké programátorské zkušenosti, v ideálním případě pak s nějakou předcházející verzí jazyka Visual Basic, a že chcete zvládnout práci se sadou Visual Studio 2013 v kontextu vytváření aplikací, které můžete nabízet na Windows Store, aplikací založených na Windows Forms a určených pro Windows desktop jak v domácnostech, tak v podnicích, webových aplikací (ASP.NET), které můžete spouštět ve webových prohlížečích, či aplikací určených pro platformu Windows Phone 8.

V této knize najdete nejen konkrétní techniky programování v jazyce Visual Basic, ale i rozsáhlý přehled strategií programování vhodných pro vývoj v jazyce Visual Basic. Rozsáhlá kolekce příkladů, která je součástí knihy, má široký záběr; tyto příklady jsou napsány spíše pro technicky zaměřené osoby, které rozumí programování, než pro úplné začátečníky či osoby, pro něž je programování pouze koníčkem. Kromě toho vás v knize seznámím i s možnostmi operačního systému Windows 8.1 a konkrétními návrhovými směrnicemi, jejichž dodržování při vývoji aplikací pro Windows 8.1 a Windows Phone 8 společnost Microsoft doporučuje.

Doporučení

Tato kniha je navržena tak, aby poučila čtenáře používající programovací jazyk Visual Basic. Navíc se dozvíte, jak se pracuje s IDE a vývojovými nástroji sady Visual Studio 2013. Tato kniha sice nepředpokládá žádné předcházející zkušenosti se sadou Visual Studio 2013, nicméně je napsána pro čtenáře, kteří rozumí programování a nejsou úplnými začátečníky. Předpokládám, že znáte alespoň základy programování či jste někdy v minulosti studovali nějakou verzi jazyka BASIC nebo Visual Basic a jste připraveni přejít od úplně základních poznatků k technikám specifickým pro konkrétní platformy.

Microsoft Visual Basic Krok za krokem také předpokládá, že máte nainstalovaný operační systém Windows 8.1 a že se chcete naučit vytvářet aplikace pro platformu Windows Store a další prostředí. Plné porozumění obsahu knihy také vyžaduje, abyste měli alespoň základní zkušenosti s prováděním běžných úkonů v systému Windows 8.1, s úpravami obrazovky Start a uživatelského rozhraní tohoto systému, abyste dokázali hledat informace na webu a upravovat základní nastavení systému. Ještě lepší ovšem je, máte-li systém Windows 8.1 nainstalovaný i na tabletu či jiném zařízení s dotykovým ovládním, neboť při návrhu systému Windows 8.1 byl kladen značný důraz právě na problematiku manipulace s digitálním obsahem pomocí dotyků a gest. Svoje aplikace pak můžete vytvářet na stolním počítači či notebooku s nainstalovanou sadou Visual Studio 2013 a systémem Windows 8.1 a následně je zkoušet na svém tabletu či jiném zařízení s dotykovým ovládním.

Pokud se týká samotné sady Visual Studio, předpokládám, že pracujete s jednou z plných, placených verzí sady Visual Studio 2013, mezi něž patří například Visual Studio 2013 Professional, Premium či Ultimate. Díky tomu budete moci vytvářet všechny typy aplikací popisovaných v této knize, včetně aplikací pro Windows Store, aplikací pro Windows desktop, založených na Windows Forms, konzolových aplikací, webových formulářů (ASP.NET) či aplikací pro Windows Phone 8.

Nemáte-li přístup k plné, placené verzi sady Visual Studio 2013, pak můžete s touto sadou experimentovat díky možnosti stažení volně šiřitelných verzí, určených pro konkrétní platformy. Tyto funkčně omezené či „Express“ verze sady Visual Studio 2013 se nazývají Express for Windows, Express for Windows Desktop, Express for Windows Phone a Express for Web.

Na webových stránkách sady Visual Studio (<http://www.microsoft.com/visualstudio>) najdete přístup jak k placeným, tak i k Express verzím sady Visual Studio. Současně zde najdete i popis rozdílů mezi všemi dostupnými verzemi.

Komu tato kniha naopak určena není

Tato kniha bude pravděpodobně zklamáním pro každého zkušeného programátora, pracujícího v jazyce Visual Basic a hledajícího pouze informace o nových funkcích sady Visual Studio 2013. Řada *Krok za krokem* je totiž určena těm čtenářům, kteří sice jsou profesionálními vývojáři, ale mají žádné či pouze velice malé zkušenosti s tématem knihy. Jste-li pokročilým vývojářem v jazyce Visual Basic, pak vás pravděpodobně brzy unaví cvičení, krok za krokem popisující základní funkce, jako například rozhodovací struktury, značkovací jazyk XAML, strategie přístupu k datům či používání platformy .NET Framework.

Vývojáři mající již značné zkušenosti s vývojem v jazyce Visual Basic budou mít pocit, že objevují již dávno objevené – nicméně to, co je jasné a zřejmé zkušeným programátorům, mnohdy nebývá až tak zřejmé těm, kteří se teprve učí pracovat s novou platformou pro vývoj. Představuje-li tedy programování pro Windows Store či pro Windows Phone pro vás něco nového, pak svoje objevování můžete zahájit právě studiem této knihy.

Uspořádání knihy

Celá kniha je rozdělena do pěti částí, z nichž každá se zaměřuje na nějaký jiný aspekt či jinou technologii, tvořící součást softwaru Visual Studio a jazyka Visual Basic. V části I „Úvod do vývoje v prostředí Visual Studio“ najdete přehled informací o vývojovém prostředí Visual Studio 2013 a jeho zásadní roli při vytváření aplikací pro platformu .NET Framework. Poté přejdete k příkladům aplikací pro platformy Windows Store a Windows Forms (tyto aplikace jsou určeny pro Windows desktop).

Druhá část, nazvaná „Návrh uživatelského rozhraní“, se zaměřuje na vytváření aplikací ve vývojovém prostředí Visual Studio, a to především aplikací pro Windows Store, Windows Forms a konzolových aplikací. Kromě jiného se dozvíte, jak se pracuje se značkovacím jazykem XAML, styly XAML, důležitými ovládacími prvky a novými návrhovými prvky systému Windows 8.1, mezi něž patří řádek nabídek, informační rámeček, dlaždice na obrazovce Start či dotykové ovládní.

Poté přejdete k části III „Techniky programování ve Visual Basicu“, zabývající se základními dovednostmi při programování v jazyce Visual Basic, včetně práce s datovými typy, používání platformy .NET Framework, strukturovaného zpracování chyb, práce s kolekcemi a generickými datovými typy, práce s daty pomocí jazyka LINQ a základů objektově orientovaného programování.

V části IV „Databázové a webové programování“ se dozvíte, jak se pracuje s daty v aplikacích pro Windows desktop a Windows Phone. Mimo jiné vám ukážu, jak se propojují data s ovládacími prvky a jak se pracuje s XML dokumenty a datovými zdroji Microsoft Access. Dále v této části najdete nejen přehled základních strategií pro vývoj webových aplikací s využitím technologie ASP.NET, ale i podrobnou ukázkou vývoje aplikace pro tuto platformu.

Poslední částí je část V, nazvaná „Programování pro Windows Phone“, v níž najdete přehled funkcí a vlastností, které vám nabízí platforma Windows Phone 8. Seznámíte se se základními hardwarovými charakteristikami této platformy, marketingovými možnostmi Windows Store a nakonec si krok za krokem zkusíte vytvořit svoji první aplikaci pro Windows Phone 8.

Jak najít optimální místo pro zahájení studia této knihy

Tato kniha je navržena tak, aby vám pomohla získat základní dovednosti v několika základních oblastech. Můžete ji použít, pokud jste nováčkem v oboru programování, pokud přecházíte z jiného programovacího jazyka či pokud přecházíte z používání sady Visual Studio 2010 nebo Visual Studio 2012 na Visual Studio 2013. Při hledání optimálního místa pro zahájení studia této knihy můžete využít následující tabulku:

Jste-li ...	Pak postupujte takto:
Programátorem, který začíná pracovat v jazyce Visual Basic	Podle postupu uvedeného v části „Instalace ukázkových kódů“, kterou najdete dále v této kapitole, si nainstalujte ukázkové projekty. Postupným studiem kapitol 1 až 21 této knihy se naučte základní dovednosti pro práci v prostředí Visual Studio 2013.
Programátorem přecházejícím ze sady Visual Studio 2010 či Visual Studio 2012	Podle postupu uvedeného v části „Instalace ukázkových kódů“, kterou najdete dále v této kapitole, si nainstalujte ukázkové projekty. Přečtete si kapitolu 1, pouze zběžně si projděte kapitoly 2 až 4 a poté si podrobně prostudujete kapitoly 5 až 21.
Programátorem zajímavícím se především o vytváření aplikací pro Windows Store a systém Windows 8.1	Podle postupu uvedeného v části „Instalace ukázkových kódů“, kterou najdete dále v této kapitole, si nainstalujte ukázkové projekty. Přečtete si kapitolu 1 až 3, poté rovnou přejděte ke kapitole 5, dále si prostudujete kapitoly 7 až 16 a nakonec se věnujte kapitole 18.
Programátorem zajímavícím se především o vytváření aplikací pro Windows desktop (technologie Windows Forms), tj. pro systémy Windows 8.1, Windows 8 či Windows 7	Podle postupu uvedeného v části „Instalace ukázkových kódů“, kterou najdete dále v této kapitole, si nainstalujte ukázkové projekty. Přečtete si kapitolu 1 až 2, poté rovnou přejděte ke kapitolám 4, 6 a 10 a nakonec si prostudujete kapitoly 11 až 17.

Konvence použité v této knize

Při sazbě této knihy byly použity následující konvence, navržené tak, aby byly informace prezentovány jasně a srozumitelně:

- Každé cvičení sestává z řady úkolů, uvedených formou číslovaných kroků (1, 2 atd.). Jednotlivé kroky popisují ty akce, které musíte provést, chcete-li dokončit celé cvičení.
- Názvy všech prvků programů – ovládacích prvků, objektů, metod, funkcí, vlastností, tříd, jednotlivých proměnných apod. – jsou vždy zdůrazněny *kurzivou*.
- Při provádění jednotlivých kroků občas narazíte i na tabulky, v nichž budou uvedeny seznamy vlastností, jejichž nastavení v sadě Visual Studio bude nezbytné. Textové vlastnosti budou vždy uvedeny v uvozovkách, přičemž při vlastním zápisu těchto vlastností nebudete muset uvozovky použít.
- Rámečky, označené „Poznámka“ či podobně, obsahují dodatečné informace týkající se popisovaných kroků anebo alternativní metody vedoucí k jejich dokončení.
- Text, který budete zapisovat (a to včetně některých bloků kódu), bude vždy uveden **tučným** písmem.
- Znaménko plus (+) mezi označeními dvou kláves znamená, že obě uvedené klávesy musíte stisknout současně. Tak například pokyn „Stiskněte Alt+Tab“ říká, že máte stisknout

klávesu Alt, držet ji stisknutou a současně stisknout klávesu Tab. Teprve poté máte obě klávesy uvolnit.

- Svislá čára mezi názvy dvou či více položek nabídky (například Soubor | Zavřít) znamená, že nejprve máte z nabídky vybrat první nabídku či položku nabídky, poté druhou atd.

Systémové požadavky

Budete-li si chtít vyzkoušet jednotlivé příklady uvedené v této knize, budete potřebovat následující hardware a software:

- Operační systém Windows 8.1. (V závislosti na konfiguraci operačního systému vašeho počítače se může stát, že budete potřebovat oprávnění administrátora lokálního počítače k instalaci či konfiguraci sady Visual Studio 2013.) V této souvislosti je nutné zdůraznit, že byť plné verze sady Visual Studio 2013 podporují starší verze operačních systémů Windows, jako například Windows 8 či Windows 7 SP1, funkce, jimiž se budeme v této knize zabývat, vyžadují Windows 8.1. Kromě toho i veškeré snímky obrazovek uvedené v této knize pocházejí z prostředí tohoto systému.
- Plnou verzi sady Visual Studio 2013, která je nezbytná k dokončení všech cvičení uvedených v této knize (přitom plnou verzí míním verzi Visual Studio 2013 Professional, Premium či Ultimate). Na webu sady Visual Studio (<http://www.microsoft.com/visualstudio>) najdete přehled rozdílů mezi těmito třemi verzemi. Alternativně můžete se sadou Visual Studio 2013 experimentovat i tak, že si stáhnete volně šiřitelné verze (tzv. Express verze), určené pro jednotlivé platformy. Tyto funkčně omezené verze sady Visual Studio 2013 se nazývají Express for Windows, Express for Windows Desktop, Express for Windows Phone a Express for Web. Je nutné zdůraznit, že pro dokončení všech cvičení uvedených v této knize si budete muset stáhnout všechny čtyři zmíněné verze. (Nicméně i v případě jejich použití narazíte na několik mezer: například nebudete schopni dokončit cvičení, která jsou součástí kapitoly 10 „Vytváření konzolových aplikací“.)
- Připojení k internetu, umožňující studium nápovědy sady Visual Studio 2013, vyzkoušení obchodu Windows Store a Windows Phone Store, a nakonec i stažení veškerých ukázkových kódů použitých v této knize.
- Počítač s procesorem o frekvenci minimálně 1,6 GHz.
- Paměť 1 GB (v případě 32bitového operačního systému) anebo 2 GB (v případě 64bitového operačního systému).
- Přinejmenším 16 GB (v případě 32bitového operačního systému) či 20 GB (v případě 64bitového operačního systému) volného místa na disku pro instalaci Windows 8.1.
- Grafický adaptér podporující technologii DirectX 9 či novější a využívající ovladač modelu WDDM 1.0 či vyššího.
- Monitor s rozlišením minimálně 1 024 x 768 bodů.

Budete-li chtít využívat i dotykové ovládání, budete potřebovat notebook s vícedotykovým ovládáním, tablet anebo displej podporující takové ovládání. Platí však, že zařízení podporující vícedotykové ovládání není pro úspěšné provedení příkladů nezbytné; je však výhodné ve chvíli, pokud chcete porozumět tomu, co vše vám taková zařízení mohou nabídnout. Lze říci, že programátor obvykle vyvíjí software na stolním či přenosném počítači a teprve následně testuje jeho vícedotykové funkce na zařízení podporujícím vícedotykové ovládání.

Ačkoliv vás v této knize seznámím i s vývojem aplikací pro Windows Phone 8, telefon s tímto systémem není pro dokončení jednotlivých cvičení nezbytný.

Ukázkové kódy

Součástí většiny kapitol této knihy jsou i cvičení, krok za krokem popisující tvorbu nějakého softwaru. Ta vám umožňují interaktivní vyzkoušení všech nových poznatků, které jste získali během studia dané kapitoly. Kód všech ukázkových projektů si můžete stáhnout z webové adresy:

<http://knihy.cpress.cz/K2161>

Instalace ukázkových kódů

Pokud chcete tyto ukázkové kódy využít během práce na jednotlivých cvičeních uvedených v této knize, pak při jejich instalaci na svém počítači postupujte takto – rozbalte soubor, který jste si stáhli z výše uvedené webové adresy. V případě potřeby si pro rozbalené soubory vytvořte samostatnou složku. Osobně vám doporučuji použití složky Dokumenty\Visual_Basic_KzK.

Práce s ukázkovými kódy

Rozbalením archivu s kódy vytvoříte složku s názvem Visual_Basic_KzK a v ní 19 podsložek – jednu pro každou kapitolu této knihy, obsahující nějaká cvičení. Chcete-li tedy nalézt příklady související s nějakou kapitolou, otevřete si složku s příslušným názvem. Příklady vztahující se k dané kapitole pak najdete v dalších podsložkách. Přitom názvy podsložek jsou shodné s názvy příkladů v knize. Tak například ve složce Dokumenty\Visual_Basic_KzK\Kapitola 02 naleznete příklad nazvaný Základy hudby. Máte-li systém nastaven tak, aby zobrazoval i přípony názvů souborů, obsahujících projekty sady Visual Studio, pak hledejte soubory s příponou .sln.

Poděkování

Tato kniha vznikla jako velmi zásadní přepracování staršího vydání Visual Basic Step by Step, vydaného nakladatelstvím Microsoft Press. Ve skutečnosti lze říci, že v mnoha – ne-li ve všech – ohledech se jedná o zcela novou knihu, a navíc je to i první kniha o programování, kterou jsem připravoval jako příručku pro více platforem, neboť se zabývá vývojem v jazyce Visual Basic jak pro Windows Store, tak i vývojem s využitím Windows Forms, Windows Web Forms či vývojem pro platformu Windows Phone. Jsem vděčný mnoha talentovaným programátorům a redaktorům za myšlenky a jiné příspěvky, jimiž se podíleli na přípravě této knihy.

Chtěl bych touto cestou poděkovat Devonu Musgravovi z nakladatelství Microsoft Press za jeho počáteční nadšení pro tento projekt a za to, že mi zprostředkoval kontakty na členy týmu produktové skupiny Visual Studio. V nakladatelství O'Reilly Media bych chtěl opět poděkovat Russellu Jonesovi, který se mnou prodiskutoval mnohá z témat probíraných v této knize a přišel s různými technickými a praktickými nápady, díky nimž jsem knihu dokončil včas. Jsem také vděčný Timu Patrickovi, technickému recenzentovi a zkušenému autorovi a vývojáři, který se mnou pracoval jak na této knize, tak i na mé další knize *Start Here! Learn Microsoft Visual Basic 2012*. (Time, možná spolu někdy napíšeme i nějakou knihu o historii!)

V redakční skupině nakladatelství O'Reilly Media bych chtěl poděkovat Kristen Brownové za naplánování redakčních recenzí a zodpovězení všech otázek týkajících se grafického návrhu; a Richardu Careymu za jeho zručné úpravy a vyřízení veškerých otázek týkajících se stylů či lokalizace a objevivších se v průběhu finálních příprav knihy. (Richarde, rád jsem s tebou

opět pracoval!) Také bych chtěl poděkovat Rebecce Demarestové, Kim Burton-Weismanové a Lindě Weidemannové za jejich důležité umělecké, redakční a technické příspěvky.

Jsem také velmi vděčný celému týmu vývojářů sady Visual Studio 2013, především za to, že mi poskytli jak zkušební, tak i release candidate verze této sady, a já jsem je mohl využít pro svoji práci. Kromě toho bych chtěl poděkovat i celému týmu vývojářů systému Windows 8.1 za jejich podporu. Své díky posílám i mnoha přispěvatelům diskuzních fór MSDN, kteří nejen kladli, ale i odpovídali otázky týkající se programování v jazyce Visual Basic pro systém Windows.

A nakonec bych chtěl předat svoje díky a obdiv i své nejbližší rodině za jejich neutuchající podporu při realizaci mých knižních projektů a různých akademických aktivit. Na přípravách této knihy se opět podílel můj syn, Henry Halvorson, který vytvořil elektronickou hudbu a grafiku. Jeho příspěvky najdete v kapitolách 3, 4, 5 a 9.

Zpětná vazba od čtenářů

Nakladatelství a vydavatelství Computer Press, které pro vás tuto knihu přeložilo, stojí o zpětnou vazbu a bude na vaše podněty a dotazy reagovat. Můžete se obrátit na následující adresy:

Computer Press

Albatros Media a.s., pobočka Brno

IBC

Příkop 4

602 00 Brno

nebo

sefredaktor.pc@albatrosmedia.cz

Computer Press neposkytuje rady ani jakýkoli servis pro aplikace třetích stran. Pokud budete mít dotaz k programu, obraťte se prosím na jeho tvůrce.

Errata

Přestože jsme udělali maximum pro to, abychom zajistili přesnost a správnost obsahu, chybám se úplně vyhnout nelze. Pokud v některé z našich knih najdete chybu, ať už chybu v textu nebo v kódu, budeme rádi, pokud nám ji oznámíte. Ostatní uživatelé tak můžete ušetřit frustrace a pomoci nám zlepšit následující vydání této knihy.

Veškerá existující errata zobrazíte na adrese <http://knihy.cpress.cz/K2161> po klepnutí na odkaz Soubory ke stažení.

Úvod do vývoje v prostředí Visual Studio

V této části:

- Kapitola 1: Možnosti vývoje v jazyce Visual Basic a Windows Store
- Kapitola 2: Integrované vývojové prostředí Visual Studio
- Kapitola 3: Vytvoření první aplikace pro Windows Store
- Kapitola 4: Aplikace pro Windows desktop: stručný úvod do používání Windows Forms

Kapitola 1

Možnosti vývoje v jazyce Visual Basic a Windows Store

Po dokončení studia této kapitoly budete schopni

- Popsat možnosti vývoje, které vám nabízí jazyk Microsoft Visual Basic 2013
- Porozumět požadavkům na distribuci aplikací prostřednictvím Windows Store

Jste připraveni začít pracovat s jazykem Microsoft Visual Basic? V této kapitole najdete přehled funkcí a vlastností vývojového prostředí Microsoft Visual Studio 2013. Seznámíte se také s různými vydáními sady Visual Studio, které si můžete zakoupit anebo stáhnout *zdarma*. Dalším tématem budou nové hardwarové a softwarové platformy a jejich využití a obrovský rozsah aplikací, které můžete pro tyto platformy vytvářet. Jak sami uvidíte, k platformám, jimiž se budeme zabývat, patří Windows Store a aplikace pro systém Windows 8.1; desktopové aplikace pro systémy Windows 7, Windows 8 a Windows 8.1; aplikace pro platformu Windows Phone 8; webové aplikace; konzolové aplikace a mnohé další.

Seznámíte se také s Windows Store, úžasným novým distribučním místem aplikací, navržených speciálně pro systém Windows 8.1. Ukážeme vám kontrolní seznam kroků, které musíte zvážit či provést ještě dříve, než začnete vytvářet aplikace pro Windows Store. A samozřejmě se dozvíte i to, jaké postupy musíte dodržet, chcete-li distribuovat a prodávat aplikace prostřednictvím Windows Store. Jakmile budete mít připravený jasný a přehledný seznam jak požadavků Windows Store, tak i funkcí zamýšleného programu, budete připraveni začít s jeho tvorbou. A po dokončení jej budete moci distribuovat milionům potenciálních zákazníků po celém světě.

Avšak ještě než začneme, musíme se chvíli věnovat použité terminologii. Tato kniha byla navržena a testována v prostředí operačního systému Windows 8.1. Aplikace pro Windows Store, které vás naučíme vytvářet, bude možné spouštět v prostředí operačního systému Windows 8.1 s nainstalovanou platformou .NET Framework 4.5.1. Řekneme vám také, jak se v jazyce Visual Basic programují aplikace využívající Windows Forms a modely konzolových aplikací. Tyto aplikace lze spouštět v prostředí známém jako „Windows desktop“. To znamená, že vámi vytvořené aplikace bude možné spustit v prostředí operačních systémů Windows 8.1, Windows 8, Windows 7 či starších, avšak pouze za podmínky, že na daných počítačích bude nainstalována správná verze platformy .NET Framework.

V této knize vám však ukážeme, jak se pomocí sady Visual Studio a technologie zvané ASP.NET vytváří ještě další typ aplikací. Jedná se o aplikace, jimž se říká Web Forms. Ty jsou určeny ke spuštění v prostředí webového prohlížeče, jímž je například Internet Explorer. A nakonec se v průběhu studia této knihy naučíte vytvářet i aplikace pro mobilní telefony, a to pomocí sady Visual Studio a Windows Phone SDK 8.0. Takové aplikace pak lze spouštět v prostředí platformy Windows Phone 8.

Řada produktů Visual Basic 2013 a možnosti jejich využití

Předpokládám, že jste si tuto knihu koupili proto, že se chcete naučit programování v jazyce Visual Basic. Ve skutečnosti je můj výchozí předpoklad ale takový, že už máte nějaké předcházející vývojářské zkušenosti – možná dokonce i s nějakou předcházející verzí Visual Basicu – a že se chcete seznámit s produktem Visual Studio 2013 v kontextu platform Windows Store, Windows Forms, Windows Phone či Web Forms. Další rozšiřování vašich zkušeností s vývojem v jazyce Visual Basic je vynikající volbou: na celém světě jsou více než 4 miliony programátorů pracujících v tomto jazyce a vyvíjejících řadu inovativních řešení, a nejnovější operační systém společnosti Microsoft, tj. systém Windows 8.1, přináší pro programátory, pracujících ve Visual Basicu, mnoho úžasných příležitostí.

Pojem „Visual Basic“ má na trhu s prostředky pro vývoj softwaru v podstatě dva významy. Z užšího technologického pohledu Visual Basic je názvem programovacího jazyka, majícího specifická syntaktická pravidla a logické procedury, které musí být dodrženy, chce-li vývojář připravit nějaký kód pro kompilátor s úmyslem vytvoření spustitelného programu neboli *aplikace*. Avšak název Visual Basic je používán i v širším slova smyslu, a to pro popis sady nástrojů a technik, využívaných vývojáři při vytváření aplikací pro operační systémy Windows pomocí určitého softwarového balíku. Dříve si vývojáři mohli koupit samostatnou verzi Visual Basicu, jako například Visual Basic .NET 2003 Professional Edition, avšak v současnosti je Visual Basic prodáván pouze jako jedna z komponent sady Visual Studio. K jejím dalším součástem patří Microsoft Visual C#, Microsoft Visual C++ a další nástroje pro vývoj.

Sada Visual Studio 2013 je distribuována v několika různých produktových konfiguracích, mezi něž patří Professional, Premium či Ultimate. Doprovázena je podмноžinou nástrojů sady Visual Studio, navrženou pro testery a známou pod označením Visual Studio 2013 Test Professional. Kromě těchto placených produktů můžete k experimentování se sadou Visual Studio 2013 využít i jednu ze zdarma šiřitelných verzí této sady, určených pro konkrétní vývojové platformy. Tyto funkčně omezené či „Express“ verze sady Visual Studio 2013 se nazývají Express for Windows, Express for Windows Desktop, Express for Windows Phone a Express for Web.

Plně placené verze sady Visual Studio 2013 se liší nejen svými cenami, ale i nabízenými funkcemi. Přitom verze Ultimate představuje nejuplněnější (a také nejdražší) sadu nástrojů pro vývoj. Na webu sady Visual Studio (<http://www.microsoft.com/visualstudio>) najdete vysvětlení rozdílů mezi výše zmíněnými verzemi této sady. Obvykle platí, že plně placené verze jsou dostupné i v podobě zkušebních verzí s licencí platnou po dobu 30 dnů, přičemž tuto dobu lze prodloužit na maximálně 90 dnů. Tyto zkušební verze nabízejí více funkcí než verze Express. A navíc pedagogové, zaměstnanci a studenti uznávaných vzdělávacích institucí si mohou stáhnout plně placené verze sady Visual Studio 2013 prostřednictvím programu Microsoft DreamSpark, přičemž platnost licencí takto stažených programů nevyprší nikdy.

Tuto knihu jsem napsal především proto, abych zdůraznil funkce a možnosti vývoje, které vám nabízejí sady Visual Studio 2013 Professional a Visual Studio 2013 Premium. Používáte-li verzi Visual Studio 2013 Ultimate, pak máte k dispozici nejen vše, co potřebujete k dokončení jednotlivých cvičení uvedených v této knize, ale i něco navíc. Avšak tyto dodatečné funkce verze Visual Studio 2013 Ultimate se vztahují především k problematice vývoje softwaru ve větších týmech a k softwarovým řešením pro velké podniky, což jsou témata, která již přesahují rámec této knihy.

Téměř všechna cvičení uvedená v této knize budete schopni dokončit i tehdy, nainstalujete-li si *všechna čtyři* Express vydání sady Visual Studio 2013 a posléze budete mezi nimi přecházet dle pokynů. (To znamená, že budete moci dokončit většinu cvičení v případě, pokud si nainstalujete Visual Studio 2013 Express for Windows, Visual Studio 2013 Express for Windows Desktop, Visual Studio 2013 Express for Web a Visual Studio 2013 Express for Windows Phone 8.) V každé kapitole vám řeknu, který produkt z řady Express je právě potřebný. Kromě toho se dozvíte i to, zda náhodou doporučená Express verze nemá taková omezení, která by vám bránila v dokončení uvedených cvičení. V některých případech však bude možné cvičení dokončit jenom s použitím plně komerční verze sady Visual Studio 2013. Toto platí například pro kapitolu 10 „Vytváření konzolových aplikací“.

Souhrnně lze říci, že jednotlivé kapitoly této knihy jsou navrženy tak, aby programátorům pracujícím v jazyce Visual Basic otevřely svět mnoha dalších – zcela jistě zajímavých – technických a obchodních příležitostí. Rozsáhlá kolekce příkladů, řešených krok za krokem, má poměrně široké zaměření, přičemž tyto příklady jsou napsány pro techniky rozumějící programování, a nikoliv pro úplné začátečníky či ty, pro něž je programování pouze příležitostným koníčkem. Stručně řečeno, cvičení, která najdete v této knize, vám zpřístupní skutečné programátorské postupy a zkušenosti.

Obrovský rozsah příležitostí a platform pro vývoj

Jak se programování ve Visual Basicu vyvíjelo v průběhu času a jaké příležitosti jsou nyní dostupné programátorům pracujícím v jeho nejnovější verzi, tj. verzi Visual Basic 2013? Ještě než začneme s vlastním psaním kódu, měli bychom si něco říci o současných trendech ve vývoji softwaru a programování pro systémy Windows.

První verze jazyka Visual Basic, označená Visual Basic 1.0, byla společností Microsoft uvedena na trh roku 1991. Již od svého počátečního představení na konferenci Windows World tento jazyk přitahoval spousty softwarových vývojářů, neboť inovativním způsobem kombinoval pokročilý kompilátor jazyka Visual Basic s integrovaným vývojovým prostředím (IDE), umožňujícím programátorům vytváření aplikací pro Windows umístováním ovládacích prvků na formuláře (tj. vlastně okna) a následným upravováním vlastností těchto prvků a souvisejícího kódu ve Visual Basicu. Od těchto skromných počátků se Visual Basic postupně vyvinul ve výkonný vývojářský nástroj, těsně spjatý s programováním pro systémy Windows a umožňující vytváření rychlých a poměrně efektivních aplikací pro tyto systémy, které bylo možné provozovat na mnoha různých hardwarových platformách.

Po roce 2000 se programátoři pracující v jazyce Visual Basic zajímali především o vytváření takových aplikací pro systémy Windows, které by podnikům umožňovaly efektivní správu dat. Díky schopnosti graficky zobrazovat informace a umožňovat k nim přístup pomocí výkonných ovládacích prvků uživatelského rozhraní si Visual Basic získal mnoho podporovatelů, v důsledku čehož se počet jeho uživatelů brzy začal počítat v milionech. V průběhu posledního desetiletí se nejdůležitější aplikace vyvinuté ve Visual Basicu řadily do kategorií různých

frontendů pro databáze, systémů pro řízení zásob, webových aplikací a pomocných programů, nástrojů pro podporu nákupu, CAD programů, vědeckých aplikací a her.

Avšak po roce 2010 doslovná exploze internetové konektivity a online obchodování zásadně změnila půdu pro softwarové vývojáře. V minulosti platilo, že většina aplikací pro systémy Windows byla spouštěna na serveru či na stolním počítači. V současné době se však prakticky všude setkáte s notebooky, tablety a chytrými telefony. Mnohdy je tomu dokonce tak, že jedna a táž osoba vlastní tři či čtyři různé typy zařízení. Uživatelé potřebují přenášet aplikace a informace jednoduchým a transparentním způsobem mezi všemi těmito zařízeními a vývojáři softwaru proto potřebují nástroje, které jim umožní vytváření aplikací, fungujících na více různých platformách či umožňujících snadný přesun z jednoho zařízení na druhé.

Vývojový tým sady Visual Studio 2013 se této výzvy nezalekl a připravil nástroj, který vývojářům umožní využití jejich dosavadní práce a současně jim umožní tvorbu různých modelů aplikací. V následujícím seznamu najdete přehled základních vývojových platform a příležitostí pro programátory pracující v jazyce Visual Basic (připomeňme si, že některé z uvedených platform jsou podporovány pouze plnou komerční verzí sady Visual Studio 2013):

- **Windows 8.1.** Vývojáři pracující ve Visual Basicu mohou vytvářet aplikace pro Windows Store, určené pro systém Windows 8.1. Tyto aplikace je možné provozovat na celé řadě různých zařízení, mezi něž patří například klasické stolní počítače, notebooky či tablety Microsoft Surface. (Poznámka: Chcete-li vytvářet nové aplikace pro Windows Store, určené pro systém Windows 8, pak musíte použít sadu Microsoft Visual Studio 2012.)
- **Windows 8, Windows 7 a Windows Server.** Vývojáři pracující ve Visual Basicu však mohou vytvářet i desktopové aplikace pro předcházející verze systému Windows a distribuovat je různými způsoby. Při tvorbě desktopových aplikací můžete například využít model Windows Forms („Win forms“) či model Windows Presentation Foundation (WPF).
- **Windows Phone 8.** Pomocí sady Visual Studio 2013 mohou programátoři vytvářet i aplikace spustitelné na platformě Windows Phone 8. Díky tomu mohou využít mnoha výhod této platformy. Aplikace pro chytré telefony se systémem Windows Phone se naučíte psát v kapitole 20 „Úvod do vývoje pro Windows Phone 8“ a kapitole 21 „Vytvoření první aplikace pro Windows Phone 8“.
- **Vývoj pro web.** Vývojáři mohou využít jazyky Visual Basic, HTML5, CSS3 či JavaScript k vytváření aplikací, které budou spustitelné na webu a budou dobře vypadat v mnoha různých webových prohlížečích. Technologie známá pod označením ASP.NET umožňuje programátorům pracujícím v jazyce Visual Basic rychlé vytváření webů, webových aplikací a webových služeb, aniž by museli znát veškeré podrobnosti týkající se uložení příslušných informací na webu. Všechny tyto možnosti si podrobněji ukážeme v kapitole 19 „Vývoj webových aplikací v prostředí Visual Studio s využitím ASP.NET“.
- **Konzolové aplikace a ovladače zařízení.** Programátoři pracující ve Visual Basicu však mohou psát i aplikace běžící v režimu příkazového řádku, kterému se také někdy říká „textová konzola Windows“ či „okno DOS“. A i když konzolové aplikace bývají primárně určeny k provádění nějakých akcí „v pozadí“, mohou také využívat knihovny platformy .NET Framework. S programováním konzolových aplikací vás seznámím v kapitole 10 „Vytváření konzolových aplikací“.
- **Kancelářské aplikace.** Vývojáři pracující ve Visual Basicu mohou i nadále připravovat makra a další nástroje rozšiřující funkcionalitu kancelářských aplikací sady Microsoft Office, k nimž patří například Word, Excel, Access či PowerPoint.

- **Xbox 360.** Další platformou, pro kterou můžete v jazyce Visual Basic programovat, je herní konzola Xbox 360. Kromě sady Visual Studio k tomu potřebujete ještě sadu Microsoft XNA Game Studio, a to ve verzi 4.0 či novější.
- **Aplikace pro Windows Azure, tj. aplikace pro webové servery a cloud.** Visual Basic je natolik výkonný, že umožňuje psaní aplikací určených pro složité webové servery, distribuovaná datová centra a verzi Windows navrženou pro cloud computing a známou pod označením Windows Azure.

To je až neuvěřitelný seznam typů aplikací! Ačkoliv na první pohled vás tento seznam může až zstrašovat, dobrou a bezesporu důležitou zprávou je, že základní programátorské poznatky či zkušenosti, které se v této knize naučíte, zůstávají platné pro všechny platformy. Kromě toho existuje celá řada nástrojů a technik, s jejichž pomocí můžete poměrně snadno přenášet práci mezi jednotlivými platformami. Tato kniha vás seznámí s mnoha základními dovednostmi, které budete používat; především se ale zaměřuje na nové nástroje, které jsou součástí sady Visual Studio 2013 a pomáhají vám při programování vašich aplikací pro systém Windows 8.1, Windows desktop a Windows Phone 8. Jakmile si však osvojíte základní dovednosti práce v jazyce Visual Basic, pak vám k většímu zaměření na jednu ze zmíněných platform bude stačit pouhé získání podrobnějších informací specifických pro daný trh.

Multiplatformní přístup k výuce Visual Basicu

Jak jste již asi stačili zjistit, aplikace pro systém Windows 8.1 se často nazývají aplikacemi pro Windows Store. Ano, propojení mezi systémem Windows 8.1 a Windows Store je opravdu *takto přímé*. Nicméně společnosti Microsoft je zřejmé, že zdaleka ne všichni vývojáři jsou připraveni psát aplikace *pouze* pro systém Windows 8.1, neboť mnozí vývojáři musí i nadále podporovat předcházející verze systémů Windows a jiní zase navrhují aplikace pro webové prohlížeče, které musí být možné spustit na mnoha různých platformách. Z tohoto důvodu v této knize popisují techniky programování ve Visual Basicu z hlediska vývoje pro mnoho různých platform. Dozvíte se, jak se vytvářejí aplikace pro Windows Store, aplikace pro Windows desktop, konzolové aplikace, webové aplikace či aplikace pro Windows Phone.

V některých případech se budu v jedné kapitole zabývat technikami programování ve Visual Basicu z pohledu pouze jedné platformy. Typickým příkladem může být kapitola 3 „Vytvoření první aplikace pro Windows Store“. V jiných případech budu zase přecházet z jedné platformy na druhou, přičemž se vám budu snažit ukázat, jakým způsobem lze funkce či vlastnosti jazyka Visual Basic a sady Visual Studio přizpůsobit rozdílným situacím. Příkladem takového přístupu může být kapitola 14 „Používání polí, kolekcí a generických typů pro práci s daty“, v níž najdete rady týkající se práce s daty, ilustrované příklady jak z platformy Windows Store, tak i platformy Windows desktop (Windows Forms).

Pro takto komplexní přístup při psaní knihy *Microsoft Visual Basic Krok za krokem* jsem se rozhodl proto, že sada Visual Studio 2013 Professional byla navržena tak, aby podporovala všechny tyto typy aplikací. Současná realita je taková, že programování ve Visual Basicu znamená vynaložení námahy na více platformách a středně pokročilí programátoři ve Visual Basicu potřebují pracovat v mnoha různých prostředích, aby rozšiřovali a dále doplňovali svoje programátorské zkušenosti a dovednosti. Protože ale programování pro Windows Store je poměrně nové, budu se této platformě věnovat o něco více než těm ostatním.

Seznámení s Windows Store

Protože Windows Store představuje nový a potenciálně profitabilní způsob prodeje a distribuce aplikací velkému množství uživatelů, chtěl bych tuto knihu začít popisem toho, co vlastně Windows Store je a jak jej můžete využít při vyhledávání a získávání potenciálních zákazníků. Z tohoto důvodu také všechny kapitoly o vývoji pro Windows Store budou mít silné *obchodní* podbarvení. Avšak ještě než se příliš „ponoříte“ do nějakého velkého projektu pro Windows Store, chtěl bych, abyste se důkladně seznámili s technickými požadavky Windows Store a předem tak věděli, co vše budete muset udělat. Společnost Microsoft také doporučuje toto „vzdělání předem“, neboť týmy vytvářející aplikace pro Windows Store budou nejvíce produktivní tehdy, budou-li předem znát všechny požadavky na certifikaci.

Co je Windows Store?

Zjednodušeně řečeno, Windows Store je elektronickým tržištěm, umožňujícím zákazníkům vyhledávání a získávání aplikací pro systém Windows. Windows Store je navržen tak, aby umožňoval distribuci aplikací pro systémy Windows 8 a Windows 8.1, podobně jako App Store společnosti Apple umožňuje zákazníkům stahování aplikací pro počítače této značky či jako Windows Phone Store umožňuje zákazníkům stahování produktů pro zařízení se systémem Windows Phone 8.



Poznámka: Windows Phone Store je podrobněji popsán v kapitole 20.

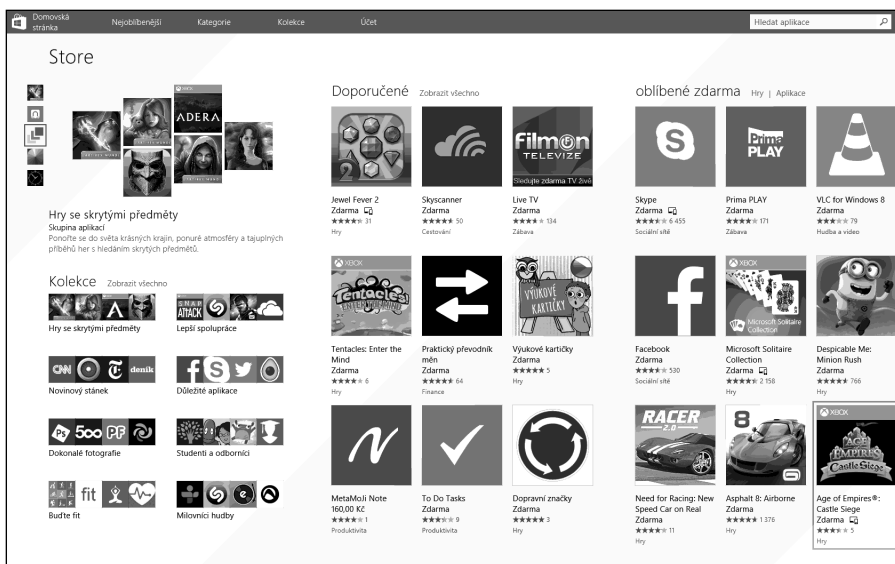
Windows Store vývojářům zpřístupňuje celý globální trh způsoby, jejichž použití by bylo v minulosti velice obtížné, ne-li přímo nemožné. Prostřednictvím Windows Store lze aplikace pro systémy Windows zpeněžit, a to buď formou prodeje aplikace za nějakou stanovenou cenu, anebo formou vložení reklamy do aplikace. Programy stažené z Windows Store jsou certifikovány a jsou připravené ke spuštění; jakmile splníte předpoklady pro přípravu nějaké aplikace pro toto tržiště, veškeré podrobnosti týkající se stahování a šíření aplikace již budou řízeny samotným Windows Store.

V průběhu čtení této knihy se naučíte vytvářet v jazyce Visual Basic a sadě Visual Studio 2013 aplikace, které je možné spustit v systému Windows 8.1. V tuto chvíli se potřebujete pouze dozvědět, jak se aplikace na Windows Store prodávají a nakupují. Kromě toho musíme společně projít kontrolní seznam, který říká, jaké funkce či vlastnosti jsou nezbytné pro certifikaci a následnou distribuci na tomto globálním tržišti.

Přístup na Windows Store

Pracujete-li na počítači se systémem Windows 8.1, pak na obrazovce Start budete mít dlaždici Store, představující bránu pro přístup k Windows Store. Pokud však na svém počítači nemáte systém Windows 8.1, pak se na adrese <http://www.windowsstore.com> můžete o Windows Store *něco dozvědět*; v žádném případě jej však nemůžete *použít*, neboť Windows Store je navržen tak, aby k němu mohli přistupovat pouze uživatelé systémů Windows 8 a Windows 8.1.

Na následující obrázku vidíte, jak Windows Store vypadá, když do něj poprvé vstoupíte. A jelikož se seznam představovaných produktů neustále mění, vaše obrazovka se bude od této ukázky zcela jistě lišit.



Na horním okraji obrazovky Windows Store najdete lištu umožňující vyhledávání aplikací podle různých kritérií. V nabídce Nejoblíbenější budete moci například hledat ty oblíbené aplikace, které jsou zdarma, oblíbené placené aplikace anebo nejlépe hodnocené aplikace. Samozřejmostí je i možnost hledání v různých kategoriích, které se vám zobrazí po rozbalení nabídky Kategorie. Najdete zde například kategorie Hry, Hudba a video, Zprávy a počasí, Finance a mnoho dalších. Pokud si vyberete nějakou kategorii a v ní pak některou ze zobrazených aplikací, uvidíte stránku podobající se této:



A právě stránka s popisem aplikace je tím místem, na němž mohou dodavatelé aplikací propagovat svoje produkty a popisovat jejich výhody či přínosy. Je velice důležité, abyste právě na tomto místě prezentovali svoje aplikace v co nejlepším možném světle. Název aplikace, její popis, seznam jejích vlastností, přehled hodnocení, cena aplikace a ukázkové obrazovky: to vše jsou zásadní faktory ovlivňující dojem, jaký vaše aplikace zanechá v návštěvnících Windows Store. Jakmile si lidé začnou vaši aplikaci stahovat či kupovat, začne rozhodování dalších návštěvníků Windows Store čím dál více ovlivňovat hodnocení vaší aplikace (to je založeno na stupnici mající maximálně 5 hvězdiček, vyjadřujících nejvyšší stupeň spokojenosti).

Instalace aplikace z Windows Store je extrémně jednoduchá: stačí pouze klepnout na tlačítko Instalovat. Během krátké chvíle se vybraná aplikace objeví na obrazovce Start vašeho počítače a budete ji moci začít používat. Stažení aplikace však vyžaduje spolehlivé připojení k internetu, které je mnohdy potřebné i k načítání dat v době běhu aplikace.

Prodejní informace a cenové hladiny

Aplikace pro Windows lze prostřednictvím Windows Store distribuovat jak zdarma, tak je také lze prodávat za předem určené ceny. Nastavení, kterému se říká *cenová hladina*, nastavuje poplatek za tu aplikaci, kterou chcete prodávat. Přitom si můžete vybrat tu cenovou hladinu, která se vám líbí; cenové hladiny začínají na 20,- Kč a v přírůstcích po 5,- Kč pokračují až do 100,- Kč; samozřejmě ale existují i vyšší cenové hladiny.

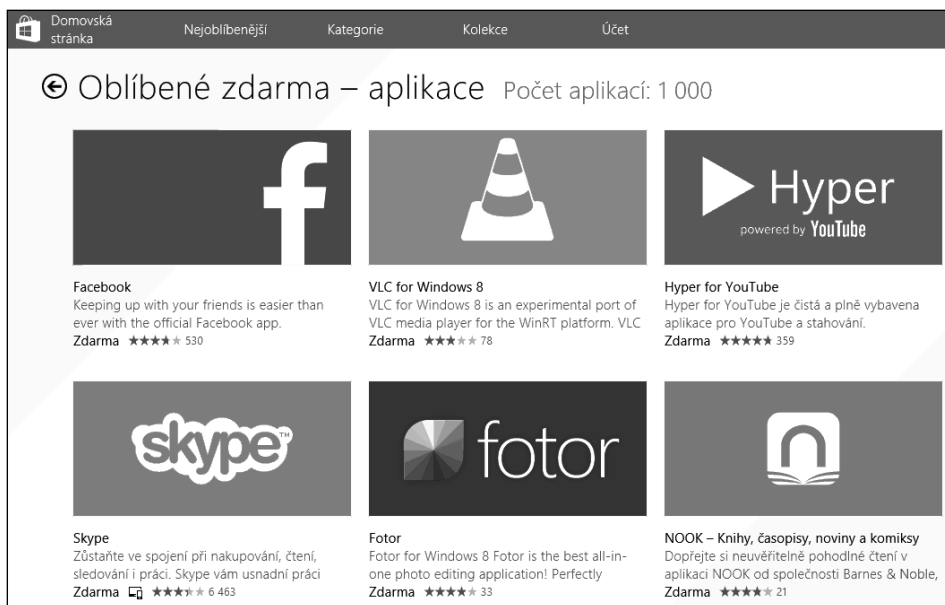
Uvažujete-li o prodeji aplikací prostřednictvím Windows Store, je nezbytné, abyste celému procesu prodeje porozuměli ještě dříve, než začnete s vlastním vývojem. Dokud celkový příjem z vaší aplikace nepřesáhne 25 000 USD, budete dostávat 70 % z toho, co za prodej vaší aplikace obdrží společnost Microsoft. Jakmile celkový příjem z vaší aplikace překročí uvedenou hranici, začnete dostávat 80 % z částky přesahující 25 000 USD. Mějte však na paměti, že vaše aplikace bude prodávána na mezinárodním trhu a v některých zemích budou tržby společnosti Microsoft sníženy o částky vyjadřující daně, určené příslušnými předpisy dané země.

Současně také platí, že ještě než začnete prodávat svoje aplikace prostřednictvím nového elektronického tržiště společnosti Microsoft, musíte se zaregistrovat jako vývojář aplikací pro Windows Store. Původní roční poplatek za účet vývojáře činil ve Spojených státech 49 USD za jednotlivce a 99 USD za společnost. Kromě toho však budete muset vyplnit ještě nějaké papírové registrační formuláře, obsahující mimo jiné kontaktní informace a další údaje.

Anebo můžete svoji aplikaci nabízet zdarma...

Samozřejmě není nutné, abyste svoje aplikace prodávali: uživatelům na celém světě můžete také nabídnout jejich stažení zdarma. Tato možnost může být zajímavá ve chvíli, kdy chcete nabídnout nějakou obecnou informaci, veřejnou službu anebo chcete pouze přitáhnout pozornost návštěvníků Windows Store k vaší společnosti či potřebujete zvýšit využívání produktů a služeb vaší společnosti. Například byste tedy mohli vyvinout aplikaci pro Windows Store, prezentující denní menu a další služby nějaké restaurace, anebo byste mohli publikovat hlavní zprávy a nejdůležitější fotografie z nějaké informační služby.

Při tvorbě aplikací nabízených zdarma se pak můžete rozhodnout generovat nějaké příjmy tím, že do aplikace zahrnete nástroje pro online reklamu, anebo se můžete rozhodnout pouze distribuovat informace s vědomím, že jste informace o svém produktu rozšířili po celém světě. Windows Store má pro aplikace nabízené zdarma speciální kategorii. To ostatně dokazuje i následující obrázek:

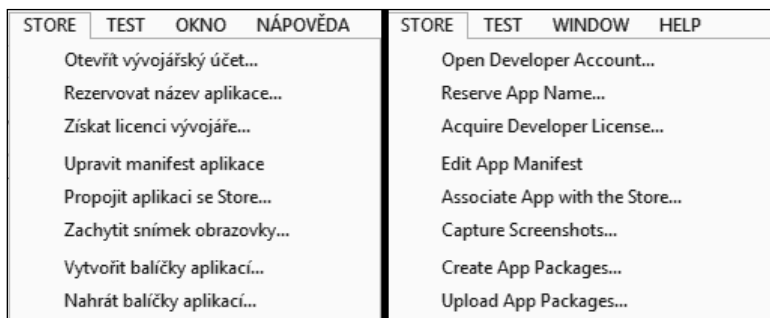


V každém případě však platí, že rozhodnutí, zda aplikaci prodávat, či naopak nabízet zdarma, závisí pouze na vás a potřebách vaší společnosti a vašich zákazníků!

Předem myslíte i na certifikaci

Avšak ještě než zahájíte jakékoliv vážnější kroky na vývoji své aplikace, měli byste se zamyslet nad doporučením společnosti Microsoft, které říká, že byste si měli pozorně prostudovat požadavky na certifikaci aplikací pro Windows Store, abyste později nebyli překvapeni některými nezbytnými kroky. Z větší části lze říci, že tyto kroky pouze vyjadřují doporučené vývojářské postupy, díky jejichž respektování se vaše aplikace stanou robustnějšími a kvalitnějšími. Společnost Microsoft se tímto snaží donutit vývojáře k dodržování vysokých standardů, neboť jejím přáním je, aby zákazníci začali důvěřovat nejen samotnému Windows Store, ale i všem na něm nabízeným aplikacím. My všichni jsme na úspěch Windows Store vsadili poměrně dost.

Součástí nabídky Projekt (Project) sady Visual Studio 2013 Professional je i podnabídka Store, obsahující 8 příkazů týkajících se Windows Store. Analogickou nabídku najdete i ve Visual Studio 2013 for Windows Express, kde vypadá takto:



Než začnete pracovat na projektu, jehož výsledek chcete zpřístupnit na Windows Store, měli byste spustit první 3 příkazy z nabídky Store. Příkazem Otevřít vývojářský účet (Open Developer Account) se přihlásíte jako jednotlivec či jako společnost. Díky tomu umožníte následné odeslání aplikace a budete moci získávat další informace. Příkazem Rezervovat název aplikace (Reserve App Name) si můžete rezervovat název své aplikace v celém Windows Store. Toto byste rozhodně měli udělat dříve, než se svým vývojem postoupíte příliš daleko a pak s překvapením zjistíte, že musíte změnit název aplikace. Poslední příkaz, jímž je Získat licenci vývojáře (Acquire Developer License), vám umožňuje získání dočasné vývojářské licence. Tu jste však již možná dostali díky své předcházející práci v sadě Visual Studio.

Poměrně dobrý blog pro vývojáře připravující aplikace pro Windows Store najdete na adrese <http://blogs.windows.com/buildingapps/>. Zde najdete články zaměstnanců společnosti Microsoft a dalších specialistů, vysvětlující základní koncepty aplikací pro Windows Store a odpovídající na dotazy, které se těchto aplikací týkají. Například kromě vyplnění kontrolního seznamu Windows Store, jehož ukázkou najdete v tabulce 1.1 níže, budete muset vyplnit úplný manifest své aplikace a používat další praktiky bezpečného programování. Další užitečné informace najdete také v článku „Publish your Store app“, který najdete na adrese <http://dev.windows.com/en-us/publish/>.

Kontrolní seznam požadavků Windows Store

Formální proces certifikace začíná ve chvíli, kdy svoji aplikaci nahrajete na Windows Store. V tabulce 1.1 najdete ukázkou kontrolního seznamu, jehož vyplnění společnost Microsoft doporučuje všem vývojářům vytvářejícím aplikace pro Windows Store. Většina položek z této tabulky je pro certifikaci nezbytná a bude vyhodnocována ve chvíli, kdy se zaregistrujete u společnosti Microsoft a vyplníte požadované online formuláře pro odeslání aplikace. Platí, že požadavky na certifikaci mohou být měněny, nicméně následující tabulka vám umožní alespoň získání představ o potřebných údajích. Musíte si však zapamatovat, že ještě před přechodem do režimu online a před odesláním své aplikace k certifikaci byste měli provést nějaké přípravné kroky. Všechny potřebné informace byste si měli připravit předem a měli byste si je důkladně zkontrolovat.

Tabulka 1.1 Kontrolní seznam pro odeslání aplikace na Windows Store

Stránka odeslání	Název pole	Popis
Název (Name)	Název aplikace (App Name)	Zadejte název aplikace mající nejvýše 256 znaků. Zvolte takový název, který bude přitahovat pozornost návštěvníků Windows Store. Je vhodné používat spíše krátké názvy.
Podrobnosti o prodeji (Selling Details)	Cenová hladina (Price tier)	Zde zadejte prodejní cenu své aplikace (anebo cenu nastavte na „zdarma“).
	Bezplatné zkušební období (Free Trial Period)	Umožněte svým zákazníkům stažení a nainstalování aplikace na bezplatnou zkušební dobu. Pokud si zákazník vaši aplikaci během zadané doby nekoupí, aplikace přestane fungovat.
	Země/regiony (Countries/Regions)	Zadejte zamýšlený trh pro svoji aplikaci.
	Datum uvolnění (Release Date)	Zadejte datum uvolnění aplikace.

Stránka odeslání	Název pole	Popis
	Kategorie (Category)	Svoji aplikaci přiřadte k některé z předdefinovaných kategorií a usnadněte tak zákazníkům její vyhledání ve Windows Store.
	Aplikace, podporující usnadnění (Accessible App)	Pokud jste svoji aplikaci navrhli tak, aby splňovala směrnice společnosti Microsoft pro usnadnění, pak to vyjádřete v tomto poli.
	Minimální úroveň funkcí DirectX (Minimum DirectX Feature Level)	Zde vyjádřete minimální požadavky na video a hardware, které vaše aplikace má.
	Minimální systémová paměť (Minimum System RAM)	Zde zadejte velikost paměti, kterou vaše aplikace vyžaduje. Základní systémové požadavky na zařízení, na nichž bude vaše aplikace spouštěna, byste měli zkontrolovat opravdu důkladně.
Pokročilé funkce (Advanced Features)	Nabídky v aplikaci (In-App Offers)	Do tohoto pole vyplňte informace o produktech, které si uživatelé budou moci koupit přímo z vaší aplikace. Nezapomeňte zadat ceny těchto produktů a také délky období, po jaké je uživatelé budou moci používat.
Hodnocení (Ratings)	Věkové hodnocení (Age Rating)	Zadejte vhodné hodnocení obsahu vaší aplikace. Při zadávání použijte některou z nabízených úrovní.
	Certifikáty hodnocení (Rating Certificates)	Prodáváte-li hru, můžete zde zadat certifikát hodnocení, vydaný nějakou organizací. Vyplnění tohoto pole závisí především na tom, kde chcete svoji aplikaci nabízet.
Šifrování (Cryptography)	Otázka 1 (Question 1)	V tomto poli můžete říci, zda vaše aplikace využívá nějaké šifrování či nikoliv.
	Otázka 2 (Question 2)	Zde určete, zda jakékoliv použití šifrování ve vaší aplikaci vyhovuje limitům daným úřadem Bureau of Industry and Commerce, který je součástí amerického Ministerstva obchodu (United States Department of Commerce).
Balíčky (Packages)	Umístění nahrávané aplikace (Package Upload Control)	Zadejte cestu k dokončenému balíčku vaší aplikace.
Popis (Description)	Popis (Description)	Zadejte jasný a výstižný marketingový popis své aplikace, jejích funkcí a jejich výhod. Obsah tohoto pole si před odesláním formuláře ještě jednou důkladně zkontrolujte. Platí, že popis smí mít délku nejvýše 10 000 znaků.
	Vlastnosti (App Features)	(Volitelné) Zadejte až 20 vlastností své aplikace. (Každá vlastnost smí být popsána nejvýše 200 znaků.)
	Klíčová slova (Keywords)	(Volitelné) Zadejte maximálně 7 klíčových slov, výstižně popisujících vaši aplikaci.

Stránka odeslání	Název pole	Popis
	Popis aktualizace (Description of Update)	Napište, jakým způsobem tato nová verze vaší aplikace aktualizuje verzi předcházející. (V případě vydání první verze své aplikace ponechte toto pole prázdné.)
	Informace o ochraně autorských práv (Copyright and Trademark Info)	Stručně – pomocí maximálně 200 znaků – popište copyright své aplikace.
	Další informace o licenci (Additional License Info)	(Volitelné) Do tohoto pole smíte zadat nejvýše 10 000 znaků.
	Ukázky obrazovek (Screenshots)	Do tohoto pole můžete vložit až 8 kvalitních obrazovek, ukazujících běh vaší aplikace. Přitom každá obrazovka smí být doprovázena popisem o maximální délce 200 slov. Dále platí, že minimální rozlišení obrázku musí být 1 366 x 766 pixelů. K přípravě těchto obrázků můžete použít příkaz Store (Store) Zachytit snímek obrazovky (Capture Screenshots), který je součástí sady Visual Studio.
	Reklamní obrázky (Promotional Images)	(Volitelné) Do tohoto pole můžete vložit další reklamní obrázky, týkající se vaší aplikace (nejvýše však 4).
	Doporučený hardware (Recommended Hardware)	(Volitelné) Zadejte maximálně 11 poznámek týkajících se hardwarových požadavků vaší aplikace.
	Web aplikace (App Website)	Zadejte URL webu vaší aplikace.
	Kontakt podpory (Support Contact)	Zadejte URL adresu podpory, kterou budou moci zákazníci použít v případě, budou-li potřebovat nějaké rady či budou-li se vás chtít na něco zeptat. Počítejte ale s tím, že na dotazy a poznatky zákazníků byste měli reagovat velmi rychle.
	Ochrana soukromí (Privacy Policy)	Do tohoto pole napište vše, co se vztahuje k tématu ochrany osobních dat, která shromáždíte o uživatelích.
	Popis nabídek v aplikaci (In-App Offer Description)	Toto pole je určeno pro informace popisující produkty, které si uživatelé mohou zakoupit z prostředí vaší aplikace. Součástí těchto údajů by měly být i ceny a informace o době, po kterou je možné zakoupené produkty používat. (Obdobné pole je uvedeno již výše. Zadejte proto shodné informace.)
Poznámky pro testery (Notes to Testers)	Poznámky (Notes)	Poskytněte hodnotitelům ve společnosti Microsoft dodatečné informace týkající se vaší aplikace a umožňující jim plně otestování jejích funkcí. Například popište případné skryté funkce či poskytněte nějaké uživatelské jméno a heslo pro testování.

Vše závisí na detailech

Význam předcházejícího kontrolního seznamu plně oceníte až poté, kdy se znovu podíváte na obsah jednotlivých nabízených aplikací na Windows Store. Čím více budete ještě před zahájením vývoje vědět o svých potenciálních zákaznících a základních funkcích, které by vaše aplikace měla mít, tím snazší pro vás bude rozhodování o návrhu a rozvržení aplikace v průběhu jejího vývoje. Na následujícím obrázku si všimněte, jak důležité jsou pro zobrazenou aplikaci její hodnocení, popis a kategorie funkcí. Stejně důležité jsou však i ukázky obrazovek, které produkt popisují vizuálně.

Na stránce Podrobnosti (ta na obrázku není vidět, neboť se jedná o třetí stránku) najdete další informace, mezi něž patří například verze, podporované procesory, podporované jazyky a případná oprávnění k aplikaci. Stránka Hodnocení a revize (ta také není ukázána) obsahuje postřehy skutečných zákazníků.

V tuto chvíli již tedy znáte základní marketingové a distribuční mechanismy aplikací nabízených na Windows Store. Nastal proto čas přejít k vlastnímu vývoji aplikací v jazyce Visual Basic pomocí sady Visual Studio. Ačkoliv mnohé z programů, které během čtení této knihy vytvoříte, budou spíše ukázkovými programy, navrženými tak, abyste se na nich naučili jednotlivé základní prvky programovacího jazyka Visual Basic, vždy byste měli mít na mysli konečný cíl svého snažení – a tím je tvorba softwaru, který ostatní lidé mohou *používat*.

Shrnutí

V závěru každé kapitoly této knihy najdete část Shrnutí, obsahující přehled toho, co jste se v dané kapitole dozvěděli. Tyto části tedy můžete využít k rychlé rekapitulaci získaných poznatků ještě před přechodem k následující kapitole.

V této kapitole jste se seznámili s možnostmi vývoje pro programátory pracující v jazyce Visual Basic. Kromě toho jste se ale dozvěděli i to, jaké možnosti vám nabízí sada Visual Studio 2013. Řekli jsme vám, jaké typy aplikací můžete vytvářet v sadě Visual Studio 2013 a jaké nástroje a platformy jsou popsány v této knize. Představili jsme vám i Windows Store, který je úžasným distribučním místem a obrovskou marketingovou příležitostí pro ty vývojáře softwaru, kteří chtějí svoje produkty buď distribuovat zdarma, anebo prodávat. V tuto chvíli již víte, jak Windows Store funguje a jaké požadavky budete muset splnit, budete-li chtít jeho prostřednictvím distribuovat svoje aplikace pro systém Windows 8.1. Ačkoliv celý proces vyžaduje nejen nějaké předběžné plánování, ale i technické a marketingové znalosti, budoucí možný přínos je zcela zásadní: Windows Store totiž může zpřístupnit vaše aplikace milionům zákazníků po celém světě.

V kapitole 2 „Integrované vývojové prostředí Visual Studio“ budete podrobně zkoumat IDE sady Visual Studio 2013, a to včetně funkcí pro spouštění a testování programů napsaných v jazyce Visual Basic, způsobů používání vývojových nástrojů v IDE a důležitých nastavení kompilátoru.

Integrované vývojové prostředí Visual Studio

Po dokončení studia této kapitoly budete schopni

- Používat integrované vývojové prostředí sady Visual Studio 2013
- Načítat a spouštět aplikace z Windows Store
- V okně Návrháře (Designer) se naučíte pracovat se soubory XAML
- Pomocí okna Vlastností (Properties) měnit nastavení vlastností
- Uspořádat si jednotlivá okna sady Visual Studio a panely nástrojů dle svých potřeb
- Nastavit si integrované vývojové prostředí pro cvičení, která vás krok za krokem budou provádět jazykem Visual Basic

V této kapitole získáte dovednosti, které potřebujete nejen k pouhému spuštění, ale i k následné práci v integrovaném vývojovém prostředí (Integrated Development Environment – IDE) sady Visual Studio. Toto prostředí je totiž tím místem, v němž budete psát svoje programy ve Visual Basicu. Při psaní této kapitoly jsem vycházel z předpokladu, že jste již někdy dříve programovali v některé z předcházejících verzí Visual Basicu, a že tedy potřebujete pouze zopakovat postupy pro práci se základními nástroji IDE. Nicméně i přesto se budeme v této kapitole podrobněji věnovat i některým základním dovednostem potřebným pro práci v IDE, a to včetně informací o tom, jak Visual Studio 2013 pracuje se značkovacím jazykem XAML a Windows Store.

Celá kapitola začíná opakováním základních příkazů nabídky sady Visual Studio a postupů pro programování. Otevřete si a spustíte jednoduchou aplikaci pro Windows Store, nazvanou Základy hudby; změníte nastavení vybrané vlastnosti; a poté si vyzkoušíte přesun, změnu velikosti, ukotvení a naopak skrytí oken nástrojů. Poté si řekneme, jak si máte nastavit IDE, aby odpovídalo cvičením uvedeným v této knize. Toto závěrečné cvičení je velice důležité především proto, že veškerá následující cvičení budou z těchto nastavení vycházet.

Před dalším pokračováním ve studiu této kapitoly byste si měli nainstalovat sadu Visual Studio 2013. Více informací o tomto procesu a o možnostech, které máte, najdete v úvodu této knihy a poté i v kapitole 1 „Možnosti vývoje v jazyce Visual Basic a Windows Store“.

Nevidíte zde dostatek informací pro začátečníky? Mějte na paměti, že toto je kniha pro ty programátory, kteří již někdy dříve pracovali s nějakou předcházející verzí jazyka Visual Basic či sady Visual Studio. Tato kniha byla v nakladatelství Microsoft Press zařazena do řady Krok za krokem, a proto byla navržena jako učebnice pro programátory, kteří začínají pracovat s jazykem Visual Basic 2013, a pak také jako učebnice základních postupů a dovedností. Ačkoliv i v této knize najdete přehled základních technik, mezi něž patří například nastavo-

vání vlastností a přesouvání oken nástrojů, podrobný popis základních postupů musíte hledat v jiné literatuře.

Zahájení práce

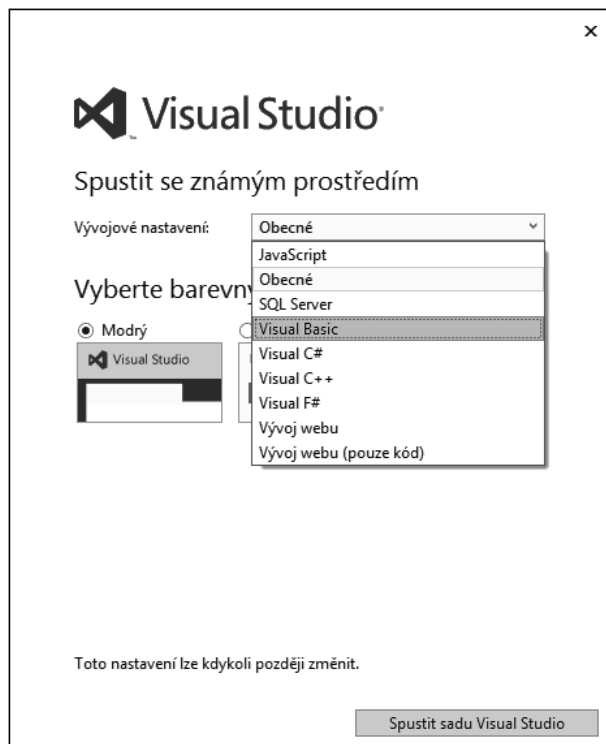
Chcete-li spustit sadu Visual Studio 2013 a začít pracovat v jejím IDE, pak postupně proveďte následující kroky. V závislosti na verzi sady Visual Studio 2013, kterou máte k dispozici, se může stát, že budete používat poněkud odlišné příkazy a uvidíte jiné obrazovky, nicméně rozdíly nebudou nikterak zásadní. (Následující ukázky obrazovek odpovídají sadě Visual Studio 2013 Professional.)

Spuštění sady Visual Studio 2013

1. Na obrazovce Start systému Windows 8.1 klepněte na dlaždici Visual Studio 2013.

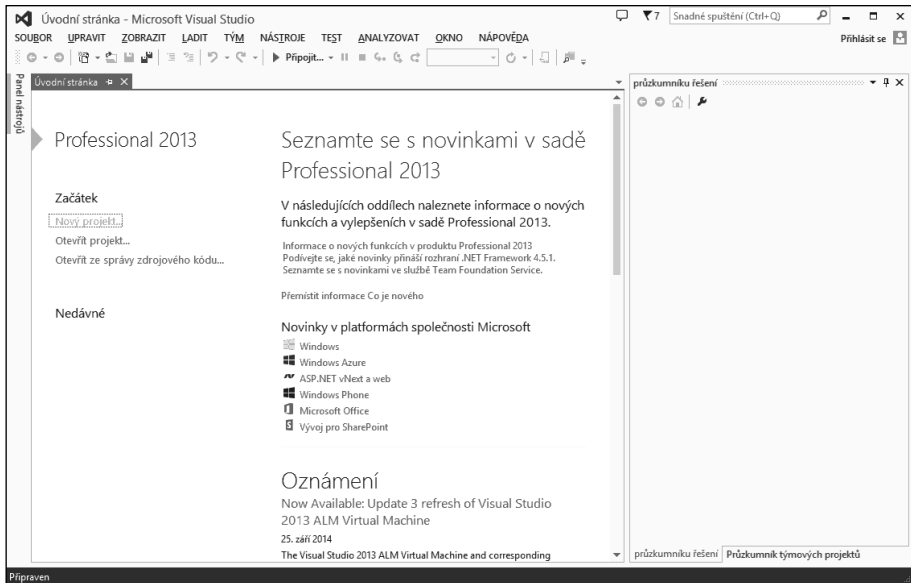
Spouštíte-li tuto sadu poprvé, počítejte s tím, že program bude potřebovat určitou dobu na konfiguraci celého prostředí. Je možné, že budete vyzváni k získání vývojářské licence pro Windows, což obvykle znamená vytvoření účtu Windows Live anebo zadání informací o již existujícím účtu. V době psaní této knihy byly vývojářské licence poskytovány zdarma a byly platné 1 měsíc. Poté bylo nutné je vždy obnovit. Lze předpokládat, že podobný postup budete muset provést i vy.

2. Jste-li v následující obrazovce požádáni o zadání svých programátorských preferencí, vyberte Visual Basic.



3. Na závěr v tomto dialogu klepněte na tlačítko Spustit sadu Visual Studio (Start Visual Studio).

Jakmile se sada Visual Studio spustí, uvidíte na obrazovce její IDE se známými nabídkami, nástroji a okny komponent. Kromě toho byste také měli uvidět úvodní stránku, obsahující sadu karet s odkazy, zdroji pro další studium, novinkami a možnostmi projektu.



Obrazovka, kterou vidíte výše, ukazuje typické nastavení sady Visual Studio 2013 Professional. Ukázka obrazovky byla vytvořena s menším rozlišením, než jaké pravděpodobně používáte na svém počítači, nicméně cílem bylo ukázat vám obsah této obrazovky v co nejuplnějši a nejzřetelnější podobě.

Při práci se sadou Visual Studio 2013 si můžete vybrat ze tří různých barevných motivů. Kvůli vyšší zřetelnosti jsem si vybral motiv Světlý (Light), a to proto, že takto připravené ukázky obrazovek jsou pro konečnou tištěnou podobu knihy nevhodnější. Nicméně při prvním spuštění sady Visual Studio si můžete vybrat barevný motiv Tmavý (Dark), zobrazující bílý text na černém pozadí. Ačkoliv tento barevný motiv je vcelku uklidňující a při práci zdůrazňuje samotný kód programu spolu s prvky uživatelského rozhraní, je možné, že si kvůli dosažení shody se zde uvedenými ukázkami obrazovek budete chtít nastavit motiv Světlý (Light).

Pokud po spuštění sady Visual Studio uvidíte tmavý barevný motiv a budete jej chtít změnit, pak z nabídky Nástroje (Tools) zvolte Možnosti (Options), vlevo v kategorii Prostředí (Environment) klepněte na Obecné (General) a z rozbalovacího seznamu Barevný motiv (Color Theme) vyberte Světlý (Light). Celý postup ukončete klepnutím na tlačítko OK. Jak jsme si již řekli, předcházející obrázek ukazuje právě barevný motiv Světlý (Light).

Vývojové prostředí sady Visual Studio

V IDE sady Visual Studio můžete otevřít nový či nějaký již existující projekt Visual Studio anebo můžete začít studovat mnoho online zdrojů, které máte při programování v jazyce Visual Basic k dispozici. Jednou z novinek sady Visual Studio 2013, které si možná všimnete hned napoprvé, je vaše přihlašovací jméno, uvedené v pravém horním rohu (pokud se nepři-

hlásíte, uvidíte zde odkaz Přihlásit se /Sign in/). Tento odkaz umožňuje přihlášení ke cloudu, díky čemuž budou všechna vaše nastavení IDE ukládána právě na tento cloud a vy si je tak budete moci snadno přenášet z jednoho počítače na druhý. Toto oceníte zejména při práci na projektech v různých lokalitách.

Pro začátek si zkusíme otevřít jeden z těch projektů, které jsem vytvořil pro vás. Tento projekt se jmenuje Základy hudby a jeho podstatou je kladení základních otázek týkajících se hudebních nástrojů, a následné zobrazování nejen správných odpovědí, ale i fotografií těchto nástrojů. Znovu připomínám, že všechny aplikace pro Windows Store, které v této knize najdete, jsou vytvořeny pro operační systém Windows 8.1. (Aplikace Windows Forms a konzolové aplikace je však možné provozovat i na starších verzích systému Windows, tj. na Windows 7, Windows 8 či Windows 8.1.)



Poznámka: V následujících krocích budete v IDE požádáni o otevření a stažení aplikace Windows Store. Pokud jste si dosud nestáhli soubory s ukázkovými příklady, měli byste tak učinit právě teď: budete totiž potřebovat otevřít určitý program na pevném disku svého počítače. Informace o tom, kde se tyto soubory nacházejí a jak si je můžete stáhnout, najdete v úvodní části této knihy.

Otevření již existujícího projektu sady Visual Studio

1. Klepněte na odkaz Otevřít projekt (Open Project), nacházející se v levé části Úvodní stránky (Start).

Zobrazí se vám dialog Otevřít projekt (Open Project). (Tento dialog si můžete spustit také tak, že v nabídce Soubor /File/ klepnete na příkaz Otevřít projekt /Open Project/ anebo stisknete klávesovou zkratku Ctrl+O.) Jak již asi sami vidíte, práce s tímto dialogem je poměrně jednoduchá, neboť dialog do značné míry připomíná dialog Otevřít soubor, používaný mnoha jinými aplikacemi pro systémy Windows.

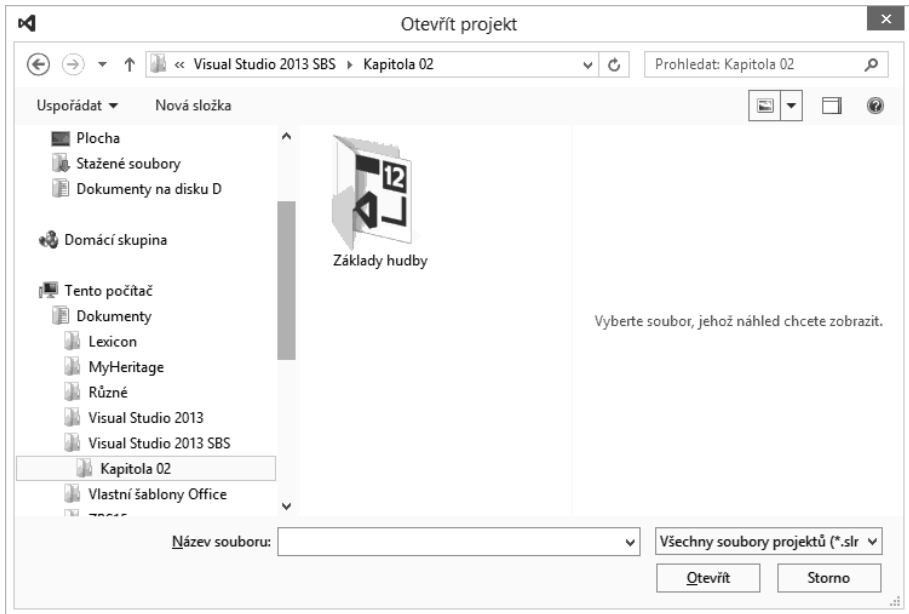


Tip: Na levé straně dialogu Otevřít projekt (Open Project) uvidíte celou řadu umístění, z nichž nejzajímavější je pro vás složka Projects, nacházející se v kategorii Microsoft Visual Studio 2013. Standardně totiž platí, že sada Visual Studio ukládá vaše projekty právě do této složky, přičemž pro každý projekt vytváří vlastní podsložku. Nicméně v této knize budeme pro práci s ukázkovými příklady používat jinou složku, a to složku Dokumenty\Visual_Basic_KzK.

2. V dialogu Otevřít projekt (Open Project) přejděte do složky Dokumenty\Visual_Basic_KzK.

Tato složka bude výchozím úložištěm pro rozsáhlou sbírku ukázkových příkladů, doprovázejících tuto knihu. Pokud jste postupovali podle instrukcí uvedených v části „Ukázkové kódy“ úvodní kapitoly, pak všechny soubory již v této složce najdete. Opět připomínám, že pokud jste si ukázkové příklady dosud nestáhli, máte nejvyšší čas.

Klepnete-li pak na složku Kapitola 02, bude dialog Otevřít projekt (Open Project) vypadat tak, jak vidíte na následující ukázce. Číslo „12“ v ikoně složky Základy hudby naznačuje, že projekt byl vytvořen pomocí 12. verze jazyka Visual Basic neboli pomocí jazyka Visual Basic 2013.



3. Otevřete složku Kapitola 02\Základy hudby a poté poklepejte na soubor řešení Základy hudby. (Máte-li systém nastaven tak, aby zobrazoval i přípony souborů, pak tento soubor bude mít příponu .sln.)
4. Visual Studio otevře stránku Základy hudby, vlastnosti a zdrojový kód řešení, jímž je aplikace pro Windows Store, navržená pro systém Windows 8.1. Průzkumník řešení (Solution Explorer), jehož okno najdete v pravé části obrazovky, zobrazuje některé ze souborů tvořících celé řešení.

Součástí sady Visual Studio je speciální volba, nazvaná Vždy zobrazit řešení (Always Show Solution) a určená k řízení několika možností souvisejících s řešeními otevřenými v IDE. Příslušné zaškrtačkové políčko najdete na kartě Projekty a řešení | Obecné (Projects And Solutions | General) dialogu Možnosti (Options), který si můžete otevřít klepnutím na příkaz Možnosti (Options) v nabídce Nástroje (Tools). Je-li tato volba zaškrtnuta (což je výchozí nastavení), bude pro každé nové řešení vytvořena nová podsložka, což povede k uložení projektu a jeho souborů do samostatné složky, nacházející se pod úrovní složky řešení.

Zachováte-li výchozí nastavení volby Vždy zobrazit řešení (Always Show Solution), pak se v IDE objeví několik dalších možností, mezi něž patří například příkazy v nabídce Soubor (File) a záznam pro řešení v Průzkumníku řešení (Solution Explorer). Pokud tedy upřednostňujete vytváření samostatných složek pro jednotlivá řešení a možnost práce s příkazy a nastaveními, týkajícími se řešení, pak vám doporučuji, abyste výchozí nastavení popisované volby ponechali beze změny. Více informací o těchto volbách se dozvíte v závěru této kapitoly.

Základní terminologie týkající se projektů a řešení

V sadě Visual Studio se vyvíjené programy běžně nazývají projekty či řešeními, neboť jsou tvořeny mnoha komponentami, nikoliv pouze jedním souborem. Součástí programů vytvořených v jazyce Visual Basic je soubor projektu (přípona .vbproj), soubor řešení (přípona .sln) a několik pomocných souborů, uspořádaných do různých podsložek. Aplikace navržená pro Windows Store bude navíc obsahovat i označovací soubor (přípona .xaml) a složku Assets.

Přitom *projekt* obsahuje veškeré soubory a další informace, vztahující se k jedné ucelené činnosti. Naopak *řešení* obsahuje veškeré informace, týkající se jednoho či více projektů. Lze tedy říci, že řešení představují vcelku užitečný mechanismus pro řízení více vzájemně souvisejících projektů. Ukázkové příklady, které jsou součástí této knihy, obvykle obsahují pouze jeden projekt v každém řešení, což znamená, že otevření souboru projektu (souboru typu .vbproj) má zcela stejný výsledek jako otevření souboru řešení (souboru typu .sln). V případě řešení obsahujících více projektů však budete obvykle otevírat soubor řešení.

Důležité nástroje IDE

Nyní bychom si měli vyhradit dostatek času k seznámení se s nástroji pro programování a okny vývojového prostředí sady Visual Studio 2013. Pokud jste již dříve programovali v jazyce Visual Basic, pak vám většina z těchto nástrojů bude připadat povědomá. Souhrnně lze říci, že tyto funkce a vlastnosti jsou těmi komponentami, které budete používat pro tvorbu svých programů ve Visual Basicu, jejich uspořádání a i testování. Několik nástrojů pro programování vám také umožní bližší seznámení s prostředky vašeho systému, a to včetně rozsáhlejšího světa databází a webových připojení.

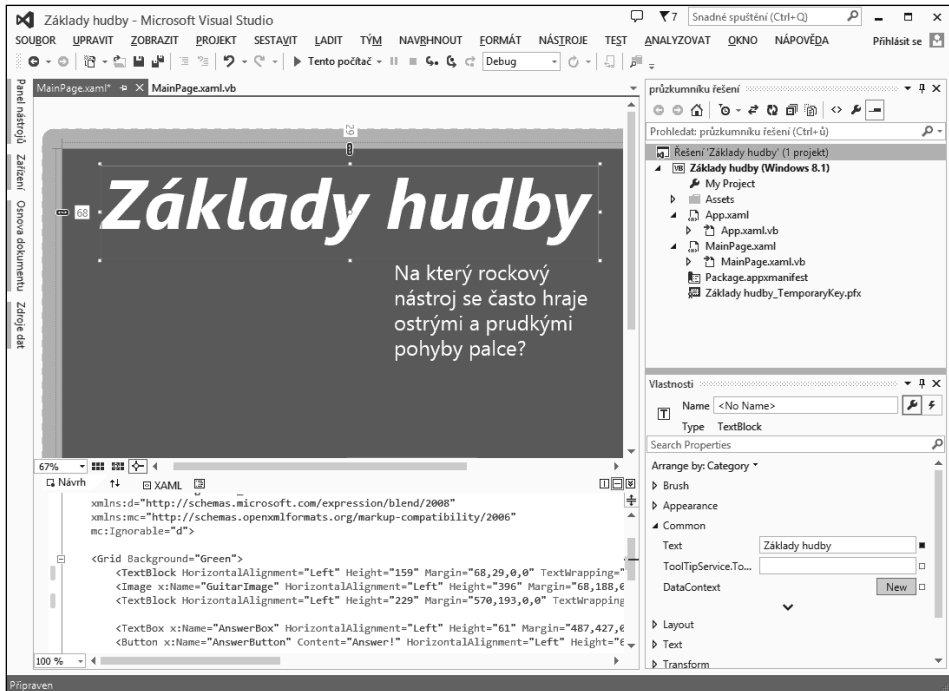
Předpokládám, že jste již pracovali s některou ze starších verzí sady Visual Studio, editorem Microsoft Word či jinými aplikacemi pro Windows, a již tedy znáte základní nabídky, panely nástrojů, systém nápovědy a často používané příkazy typu Nový projekt (New Project), Zavřít (Close), Spustit ladění (Start Debugging) a Uložit vše (Save All). Úplný seznam všech panelů nástrojů si můžete kdykoliv zobrazit tak, že pravým tlačítkem myši klepnete na libovolný panel nástrojů, zobrazený v IDE.

Nabídka Nápověda (Help) je v sadě Visual Studio velice užitečná. Rozsáhlou kolekci zdrojů, týkajících se práce v sadě Visual Studio a programování v jazyce Visual Basic, však navíc najdete i na webové adrese <http://msdn.microsoft.com>, což je web Microsoft Developer Network (MSDN). Během studia této knihy vás mnohdy vyzvu k tomu, abyste se navíc podívali i na související témata právě na tomto webu.

Následující obrázek ukazuje některé z nástrojů a oken IDE sady Visual Studio Professional. Nelekejte se toho, že tento obrázek vypadá jinak než to vývojové prostředí, které právě vidíte na svém počítači. V dalších částech této kapitoly vás podrobněji seznámím s mnoha pro vás dosud neznámými prvky IDE a možnostmi přizpůsobení jednotlivých obrazovek.

Hlavními nástroji, které v IDE sady Visual Studio uvidíte, jsou Návrhář (Designer), Průzkumník řešení (Solution Explorer), okno Vlastnosti (Properties) a karta jazyka XAML (eXtensible Application Markup Language) editoru kódu. Je však možné, že se vám zobrazí i některé specializované nástroje, jako například okna Panel nástrojů (Toolbox), Osnova dokumentu (Document Outline), Zařízení (Device), Zdroje dat (Data Sources), Průzkumník serveru (Server Explorer) či Prohlížeč objektů (Object Browser); případně se tyto nástroje mohou v IDE objevit i jako karty. Jelikož preference každého vývojáře jsou jiné, je velice obtížné předvídat, co uvidíte, pokud jste svoji instalaci sady Visual Studio 2013 použili již někdy

dříve. (To, co vidíte na obrázku výše, odpovídá prvnímu spuštění právě nainstalované sady Visual Studio, v níž je otevřena nějaká ukázková aplikace v režimu vývoje.)



Poznámka: V této souvislosti je nutné zdůraznit, že nabídky IDE se mohou měnit v závislosti na tom, co právě v sadě Visual Studio děláte.

Novou funkcí sady Visual Studio 2013 je tlačítko Zpětná vazba (Feedback), které je zobrazeno v podobě ikony „myšlenkové bubliny“ či ikony úsměvu v horní liště obrazovky, a to vlevo od políčka Snadné spuštění (Quick Launch). Tlačítko Zpětná vazba (Feedback) můžete využít k předání informací společnosti Microsoft o tom, které funkce či vlastnosti sady Visual Studio se vám líbí anebo naopak nelíbí. Kromě toho můžete toto tlačítko využít i k přístupu na fóra MSDN a k nahlášení chyby v softwaru.

Není-li některý z uvedených nástrojů zobrazený a vy si jej chcete zpřístupnit, pak klepněte na nabídku Zobrazit (View) a následně si v ní vyberte požadovaný nástroj. Jelikož nabídka Zobrazit (View) se v průběhu posledních let stále rozšiřovala, společnost Microsoft se rozhodla některé méně často používané volby z této nabídky přesunout do podnabídky nazvané Ostatní okna (Other Windows). Pokud tedy přímo v nabídce Zobrazit (View) nevidíte to, co hledáte, zkuste se podívat sem.

K několika dalším novinkám sady Visual Studio 2013, které vidíte již nyní, patří barevnější ikony v panelech nástrojů a v Průzkumníku řešení (Solution Explorer): například ikony složek jsou nyní žluté. Kromě toho se součástí posuvníku v editoru kódu stal ukazatel pozice, znázorňující relativní pozici místa, do něž chcete vkládat kód, vůči začátku dokumentu.

Uspořádání nástrojů v IDE

Vaše IDE možná nevypadá přesně tak jako předcházející ukázka, neboť přesná velikost a tvar jednotlivých nástrojů a oken v IDE závisí na tom, jak přesně jste si svoje IDE nakonfigurovali. V případě sady Visual Studio si můžete okna ve vývojovém prostředí zarovnat a připojit (neboli *ukotvit*) tak, abyste viděli pouze ty prvky, které opravdu chcete vidět. Kromě toho si můžete nástroje i částečně skrýt formou jednotlivých karet, nacházejících se na levém okraji vývojového prostředí, což vám následně dává možnost rychlého přecházení z jednoho nástroje do druhého. Tak například pokud klepnete na popisek Panel nástrojů (Toolbox) na levém okraji okna, příslušný panel nástrojů vyskočí a budete jej moci ihned použít. Pokud poté klepnete na jiný nástroj či na jiné okno v IDE, původní panel nástrojů se opět skryje.

Vaše vývojové prostředí bude pravděpodobně vypadat nejlépe tehdy, pokud upravíte nastavení monitoru a plochy Windows tak, abyste maximalizovali prostor na obrazovce; nicméně i v tomto případě může dojít k tomu, že obrazovka bude poněkud přeplněná. V praxi je tomu tak, že mnozí profesionálové pracující v sadě Visual Studio používají dva monitory, zobrazující různé pohledy na vyvíjený software.

Hlavním smyslem celé této složitosti související s nástroji je přidat do IDE mnoho nových a užitečných funkcí a současně vám nabídnout nějaké chytré mechanismy pro jejich rozumné uspořádání. K těmto mechanismům patří takové funkce, jako například možnost ukotvení, automatického skrývání, možnost používání plovoucích panelů nástrojů a několik dalších stavů oken, jejichž popis najdete dále. Visual Studio 2013 také automaticky skrývá zřídka používané funkce IDE až do doby, dokud s nimi nezačnete skutečně pracovat. I to je jedna z cest vedoucích k „vyčištění“ pracovní plochy IDE.

Pokud v sadě Visual Studio právě píšete či se chystáte psát svoji první aplikaci pro Windows Store, pak nejlepším způsobem řešení „nadbytku“ funkcí je skrytí těch nástrojů, jejichž používání neplánujete; vytvoříte si tak dostatek místa pro důležité nástroje. Mezi zásadní okna a nástroje, postačující pro vývoj středně složitých úloh ve Visual Basicu – což jsou právě ty, jimiž se budeme v této knize zabývat –, patří okna Návrhář (Designer), Vlastností (Properties), Průzkumník řešení (Solution Explorer) a Panel nástrojů (Toolbox). Zcela určitě nebudete zpočátku potřebovat panely nástrojů Osnova dokumentu (Document Outline), Průzkumník serveru (Server Explorer), Zdroje dat (Data Sources), Zobrazení tříd (Class View), Prohlížeč objektů (Object Browser), Zařízení (Device) či Ladit (Debug); s některými z těchto panelů nástrojů budeme pracovat až v pozdějších kapitolách této knihy. Pokud tedy některé z uvedených oken nechcete mít zobrazené, klepnutím na tlačítko Zavřít (Close) v jeho pravém horním rohu si jej zavřete.

V následujících cvičeních si zopakujeme chování základních nástrojů IDE. Kromě toho si řekneme něco i o značkovacím jazyku XAML, což je návrhový jazyk, používaný k definici uživatelského rozhraní aplikací určených pro Windows Store.

Návrhář a značkovací jazyk XAML

Pokud jste dokončili předcházející cvičení („Otevření již existujícího projektu sady Visual Studio“), pak ve vývojovém prostředí sady Visual Studio máte načtený projekt Základy hudby. Nicméně uživatelské rozhraní neboli stránka projektu nemusí být v sadě Visual Studio zatím zobrazená. (Složitější projekty mohou obsahovat více stránek, avšak tento první ukázkový program využívá pouze jednu.) Chcete-li si stránku projektu Základy hudby v IDE zobrazit, budete k tomu potřebovat Průzkumník řešení (Solution Explorer).



Poznámka: Nemáte-li projekt Zásady hudby dosud načtený, vraťte se v této kapitole o několik stran zpět a proveďte všechny úkony cvičení nazvaného „Otevření již existujícího projektu sady Visual Studio“.

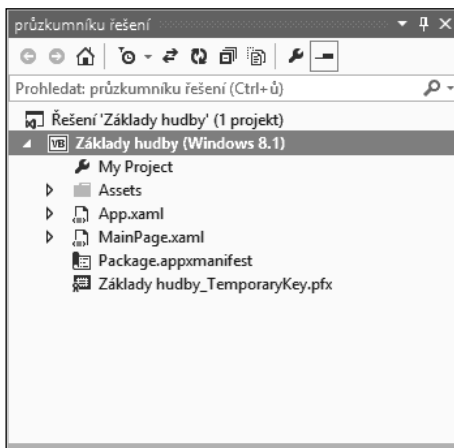
Zobrazení okna Návrháře (Designer)

1. V okně sady Visual Studio vyhledejte okno Průzkumníku řešení (Solution Explorer). To se obvykle nachází v pravé horní části celého prostředí. Pokud toto okno nevidíte, pak si klepnutím zobrazte nabídku Zobrazit (View) a následně si volbou Průzkumník řešení (Solution Explorer) požadované okno otevřete.



Poznámka: Od tohoto okamžiku se v knize budete občas setkávat se zkráceným způsobem popisu voleb z různých nabídek. Tak například slova „vyberte Zobrazit (View) | Průzkumník řešení (Solution Explorer)“ znamenají „klepněte na nabídku Zobrazit (View) a poté si zvolte Průzkumník řešení (Solution Explorer)“.

Jakmile se vám projekt Zásady hudby načte, bude okno Průzkumník řešení (Solution Explorer) vypadat takto:



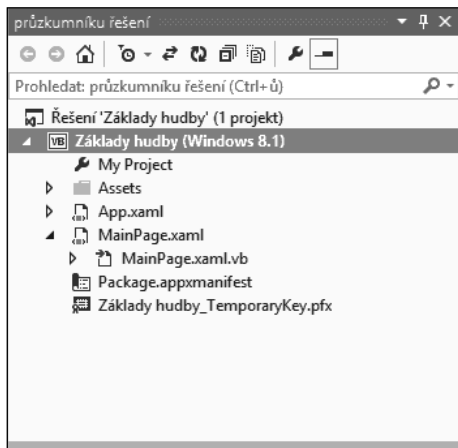
Podobně jako většina základních aplikací pro Windows Store i toto řešení obsahuje soubor App.xaml, v němž jsou uložena globální nastavení projektu a informace o prostředcích; dále obsahuje složku Assets, v níž najdete kolekci různých souborů s logy a spouštěcí obrazovku; soubor certifikátu, obsahující dočasné klíče; *manifest balíčku nasazení*, což je soubor, v němž jsou uloženy informace o sestavení a nastavení pro distribuci; a dále jedno či více oken uživatelského rozhraní neboli *stránek*, které lze snadno identifikovat díky jejich příponě .xaml.

Poblíž horního okraje okna Průzkumníku řešení (Solution Explorer) je program Zásady hudby identifikován jako projekt pro systém Windows 8.1. To znamená, že celá aplikace je navržena pro platformu Windows 8.1, a využívá tedy funkce, které jsou součástí tohoto operačního systému.

V sadě Visual Studio 2012 jste zde v případě projektů pro Windows Store viděli ještě složku Common, v níž byly uloženy společné třídy a styly XAML. Tyto prvky jsou však nyní skryty.

2. Klepněte na šipku, která se v okně Průzkumníku řešení (Solution Explorer) nachází vlevo vedle souboru MainPage.xaml.

Jakmile tento soubor rozbalíte, bude okno Průzkumníku řešení (Solution Explorer) vypadat takto:



V případě tohoto projektu pro Windows Store je hlavní stránka programu Základy hudby definována souborem MainPage.xaml.

Soubor MainPage.xaml si můžete otevřít buď v pohledu Návrh (Design), v němž budete moci ke kontrole a úpravám grafického uživatelského rozhraní využít grafické nástroje, anebo v editoru kódu, v němž můžete uživatelské rozhraní upravovat pomocí jazyka XAML, což je jazyk pro definici uživatelského rozhraní navržený pro aplikace určené pro Windows Store a další počítačové programy.



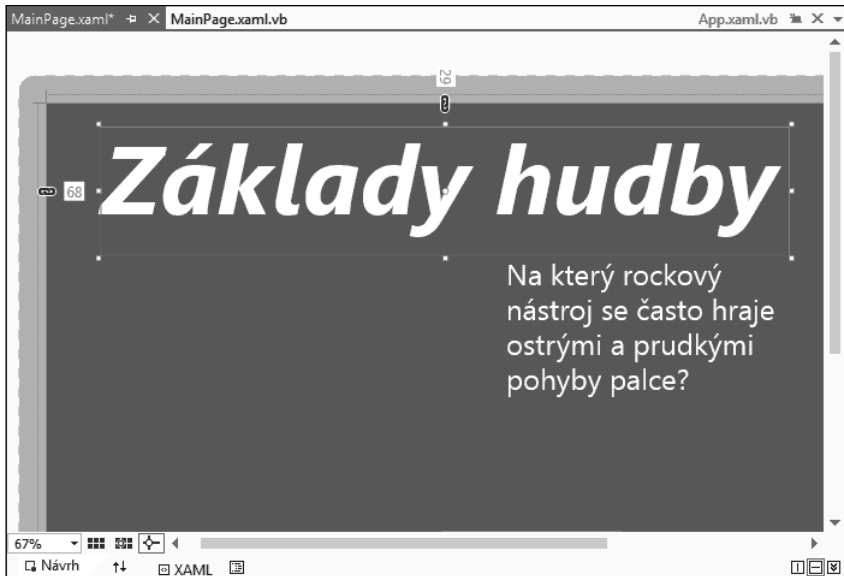
Tip: Pokud jste v některé z předcházejících verzí sady Visual Studio vytvářeli aplikace založené na technologii Windows Presentation Foundation (WPF), pak vězte, že toto je ten samý značkovací jazyk XAML (pouze rozšířený o několik důležitých aktualizací), s nímž jste možná již pracovali při vytváření uživatelského rozhraní pro aplikace napsané ve Visual Basicu. V podstatě lze říci, že aplikace pro Windows Store jsou nástupcem aplikací využívajících technologii WPF.

Pod souborem MainPage.xaml uvidíte druhý soubor, nazvaný MainPage.xaml.vb. Tento soubor také souvisí s uživatelským rozhraním projektu Základy hudby. Jedná se totiž o tzv. *soubor kódu na pozadí (code-behind file)*, neboť obsahuje programový kód v jazyce Visual Basic, vztahující se k uživatelskému rozhraní, definovanému souborem MainPage.xaml. Jakmile se naučíte vytvářet aplikace pro Windows Store v jazyce Visual Basic pomocí sady Visual Studio, budete schopni začít si tento soubor upravovat.

Průzkumník řešení (Solution Explorer) je bránou k práci s různými soubory vašeho projektu – lze tedy říci, že se jedná o základní nástroj. Poklepete-li na kterýkoliv soubor v Průzkumníku řešení (Solution Explorer), daný soubor se vždy otevře v příslušném editoru. Toto platí za podmínky, že jsou povoleny přímé úpravy vybraného souboru.

3. V Průzkumníku řešení (Solution Explorer) poklepejte na soubor MainPage.xaml a zobrazte si tak uživatelské rozhraní projektu v okně Návrháře (Designer). V případě potřeby si zobrazení uživatelského rozhraní můžete upravit pomocí svislého posuvníku.

V okně Návrháře (Designer) byste nyní měli vidět základní stránku projektu Základy hudby. Ukázku vidíte níže:



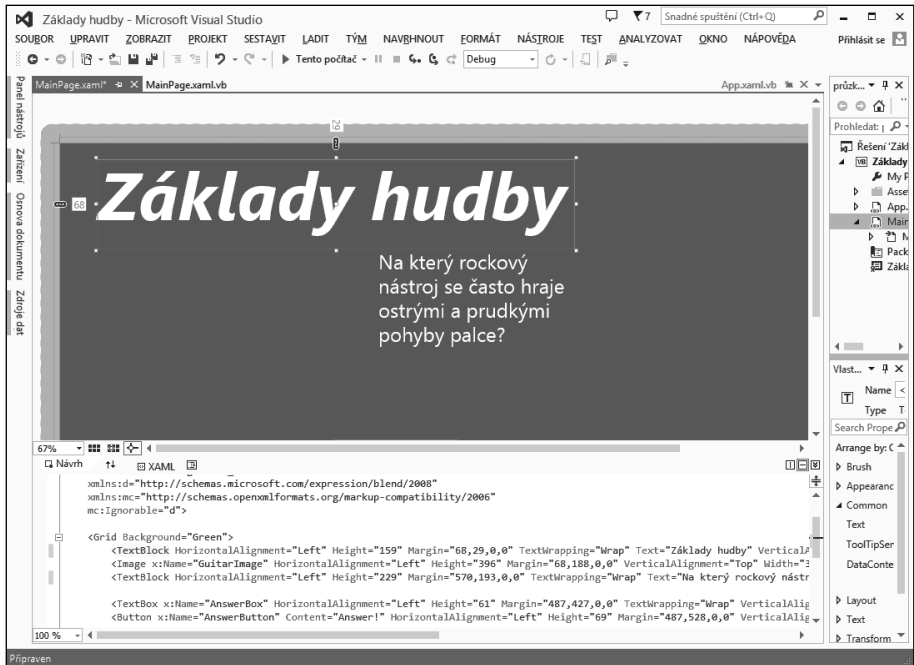
Všimněte si, že karta s názvem souboru MainPage.xaml je zobrazena poblíž horního okraje okna Návrháře (Designer), a to spolu s dalšími kartami. Platí, že kdykoliv můžete klepnout na kteroukoliv jinou kartu a zobrazit si tak obsah příslušného souboru.

Jak jsme si řekli již dříve, soubor MainPage.xaml představuje vizuální podobu uživatelského rozhraní programu. Avšak pokud chcete, můžete si poměrně snadno zobrazit kód značkovacího jazyka XAML, definujícího toto uživatelské rozhraní: stačí, když poklepete na kartu XAML editoru kódu, nacházejícího se ve spodní části okna Návrháře (Designer).

Jelikož v tuto chvíli pravděpodobně nevidíte celé uživatelské rozhraní, budete si zřejmě chtít upravit velikost okna Návrháře (Designer) tak, abyste viděli co největší část uživatelského rozhraní.

4. Ukazatel myši přesuňte na pravou hranu okna Návrháře (Designer), která je totožná s pravým okrajem svislého posuvníku. Ukazatel myši by se měl změnit na oboustrannou šipku, což je ukazatel pro změnu velikosti. Stiskněte levé tlačítko a hranu okna přesuňte směrem doprava. Tímto způsobem si okno zvětšete dle potřeby.

Okno Návrháře (Designer) se zvětší, avšak současně se zmenší okna Průzkumníku řešení (Solution Explorer) a Vlastnosti (Properties). Po této úpravě by vaše obrazovka mohla vypadat například takto:



Nyní si prostudujeme kód ve značkovacím jazyce XAML, definující ty prvky uživatelského rozhraní, na něž se v okně Návrhář (Designer) právě díváte.

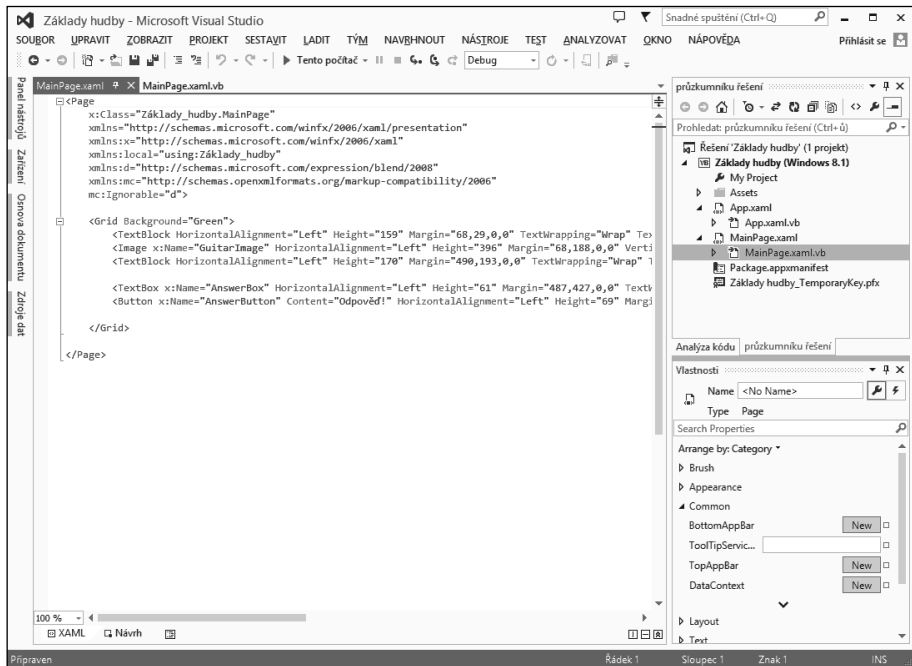
5. Okno Návrhář (Designer) zmenšíte na původní velikost a poté poklepete na kartu XAML. V editoru kódu se zobrazí XAML kód uživatelského rozhraní projektu Základy hudby.
6. Přejděte do horní části okna a zobrazte si tak celý dokument.

Uvidíte zhruba to co na obrázku na protější stránce.

V editoru kódu se zobrazí obsah souboru MainPage.xaml, napsaný pomocí značkovacího jazyka XAML. Jsou to právě tyto strukturované informace, které říkají, jakým způsobem má Visual Studio – a následně i systém Windows – zobrazovat uživatelské rozhraní aplikace a její grafické části. Máte-li alespoň základní znalosti HTML či WPF, měl by vám tento kód připadat poněkud povědomý. Součástí souboru, zapsaného pomocí jazyka XAML, jsou značky – tj. instrukce, jejichž hlavním smyslem je říci programu, jak má na obrazovce zobrazovat jednotlivé věci. Značky jazyka XAML jsou v našem příkladu zapsány mezi návěštímí `<Page>` a `</Page>`. Z důvodu zvýšení srozumitelnosti zapsaného kódu jsou tyto značky navíc odsazeny. (Aplikace pro systém Windows Phone také využívají pro definici uživatelského rozhraní značky jazyka XAML.)

Prvních sedm řádků, následujících za návěštímí `<Page>`, definuje zdroje sloužící k vytvoření uživatelského rozhraní. Za těmito řádky následuje sekce uvozená návěštímí `<Grid>` a `</Grid>`. Ta popisuje objekty uživatelského rozhraní. Jak vidíte, tato část kódu definuje dva ovládací prvky `TextBlock`, jeden ovládací prvek `ImageControl`, další ovládací prvek `TextBlock` a na závěr ovládací prvek `Button`. Vráťte-li se zpět do okna Návrhář, pak uvidíte, kolik z těchto prvků je viditelných. (Dva z nich jsou v tuto chvíli skryté, a proto nejsou viditelné.) Kromě toho si v editoru kódu můžete prohlédnout i nastavení jednotlivých

vlastností těchto objektů. Jak vidíte, vlastnosti jsou popsány svými názvy (například *HorizontalAlignment*) a hodnotami (například *Left*).



Značkovacímu jazyku XAML se budeme podstatně podrobněji věnovat v následujících kapitolách. V tuto chvíli byste si měli zapamatovat to, že okno Návrhář (Designer) vám umožňuje jak zobrazení *náhledu* uživatelského rozhraní, tak i kódu v jazyce XAML, popisujícího konkrétní charakteristiky objektů zobrazených v náhledu rozhraní. Programátoři pracující se sadou Visual Studio si během práce na nějakém programu často chtějí zobrazit tyto dva panely vedle sebe. *Lze říci, že pokud jste někdy v minulosti vytvářeli HTML aplikace pro web, pak by vám tento koncept měl připadat povědomý, neboť řada nástrojů pro vývoj webových stránek zobrazuje v horní části obrazovky náhled stránky, zatímco ve spodní části zobrazuje jejich HTML kód.*



Tip: Na spodním okraji okna Návrhář (Designer) a editoru kódu se nachází řada užitečných tlačítek, nabízejících vám poměrně značné možnosti pro řízení rozdělování obrazovky s těmito prvky. V levém spodním rohu okna Návrhář (Designer) jsou nejen karty XAML a Návrh (Design), ale i tlačítko Osnova dokumentu (Document Outline), umožňující otevření samostatného okna, v němž jsou zobrazené objekty uživatelského rozhraní, uspořádané podle typu. V pravém spodním rohu okna Návrhář (Designer) pak najdete tlačítka Svislé rozdělení (Vertical Split), Vodorovné rozdělení (Horizontal Split) a Sbalit / Rozbalit panel (Collapse / Expand Pane). Pomocí těchto tlačítek lze určit uspořádání oken Návrhář (Designer) a editoru kódu. Velmi užitečné je především tlačítko Sbalit / Rozbalit panel (Collapse / Expand Pane): jedná se vlastně o přepínač, s jehož pomocí můžete okna zobrazovat buď jednotlivě, anebo vedle sebe (či nad sebou).

7. Klepnutím na kartu Návrh (Design) si znovu zobrazte hlavní stránku projektu v okně Návrháře (Designer).
8. V pravém spodním rohu klepněte na tlačítko Rozbalit panel (Expand Pane). Pod oknem Návrháře (Designer) se zobrazí druhé okno, obsahující ten kód v jazyce XAML, který hlavní stránku projektu vykresluje.

A nyní si zkusíme tento jednoduchý program v sadě Visual Studio spustit.

Spouštění a testování aplikací pro Windows Store

Projekt Základy hudby je jednoduchou aplikací pro Windows Store, napsanou v jazyce Visual Basic. Tuto aplikaci jsem vytvořil proto, abych vás seznámil s nástroji pro programování obsaženými v sadě Visual Studio. Při úpravách stránky, kterou nyní vidíte, jsem vytvořil celkem 5 objektů, a do souboru kódu na pozadí jsem přidal celkem tři řádky kódu, díky nimž vám program položí jednoduchou otázku a následně zobrazí správnou odpověď. Více informací o vytváření objektů tohoto typu a přidávání kódu v jazyce Visual Basic do souboru kódu na pozadí najdete v kapitole 3 „Vytvoření první aplikace pro Windows Store“. V tuto chvíli si pouze vyzkoušíme spuštění ukázkového programu ve vývojovém prostředí sady Visual Studio.

Spuštění programu Základy hudby

1. Klepněte na tlačítko Tento počítač (Local Machine). Tlačítko poznáte podle toho, že před uvedeným názvem se nachází ještě zelený trojúhelníček směřující vpravo. Pokud byste toto tlačítko na panelu nástrojů neviděli, pak si zkontrolujte, zda máte zobrazený panel nástrojů Standardní (Standard).



Tip: Chcete-li ve vývojovém prostředí sady Visual Studio spustit nějaký program, pak kromě výše uvedeného postupu můžete též z nabídky Ladit (Debug) zvolit volbu Spustit ladění (Start Debugging) anebo můžete jednoduše stisknout klávesu F5.

Visual Studio projekt načte a zkompiluje jej do *sestavení*, což je soubor typu EXE, obsahující data a kód v takové podobě, v jaké je dokáže počítač zpracovat. Toto konkrétní sestavení však obsahuje i informace užitečné pro testování neboli *ladění*, které je jednou ze základních součástí celého procesu vývoje nějakého softwaru. Je-li proces kompilace úspěšný, Visual Studio spustí program ve vývojovém prostředí. (Tomuto procesu se říká spuštění programu na *lokálním počítači*, což je něco jiného než spuštění programu na *vzdáleném počítači*, připojeném kdesi k internetu, anebo spuštění programu v *softwarovém simulátoru* nějakého typu.)

V době běhu programu je v hlavním panelu systému Windows zobrazena jeho ikona. Po chvíli se vám uživatelské rozhraní programu Základy hudby spustí v systému Windows 8.1 zcela stejně jako jakákoliv jiná aplikace. (Je také možné, že si na horním okraji obrazovky všimnete jakýchsi čísel, která slouží pouze pro účely ladění; z důvodu zvýšení srozumitelnosti jsem je z následujícího obrázku odstranil.)

Jinak však program vypadá úplně stejně jako jeho náhled, který jsme si před chvílí ukázali v okně Návrháře (Designer) sady Visual Studio:

Základy hudby

Na který rockový nástroj se často hraje ostrými a prudkými pohyby palce?

Odpověď!

Jak vidíte, program Základy hudby vám položí jednoduchou otázku: „Na který rockový nástroj se často hraje ostrými a prudkými pohyby palce?“

2. Správnou odpověď si zobrazíte klepnutím na tlačítko Odpověď! Text odpovědi („Na basovou kytaru“) se zobrazí pod tímto tlačítkem a současně se vlevo objeví fotografie hráče na tento nástroj.

Základy hudby



Na který rockový nástroj se často hraje ostrými a prudkými pohyby palce?

Odpověď!

Na basovou kytaru

3. Běh aplikace můžete ukončit stažením záhlaví okna (nebo horní části okna) ke spodnímu okraji obrazovky. Samozřejmě ale můžete použít i jakýkoliv jiný způsob, na který jste zvyklí.

Jakmile přesunete ukazatel myši k hornímu okraji okna, změní se jeho tvar na ruku, čímž vám program dává jistou zpětnou vazbu při přesouvání záhlaví okna ke spodnímu okraji obrazovky s cílem ukončení běhu programu. Po uzavření okna aplikace můžete vývojové prostředí sady Visual Studio znovu aktivovat stisknutím klávesy Windows a následným klepnutím na ikonu sady Visual Studio.

Svým předcházejícím úkonem jste však pouze vývojovému prostředí předali pokyn k ukončení běhu aplikace. To znamená, že aplikace Základy hudby může ještě chvíli běžet

na pozadí, a to po dobu, než sada Visual Studio tento požadavek zpracuje. (Může se například stát, že po nějakou dobu budete mít v záhlaví okna sady Visual Studio zobrazený text „Základy hudby (Spuštěno)“, což znamená, že aplikace ve vývojovém prostředí stále ještě běží.) Chcete-li okamžitě ukončit jakoukoliv aplikaci, kterou jste předtím spustili v IDE sady Visual Studio, klepněte na tlačítko Ukončit ladění (Stop Debugging), nacházející se na panelu nástrojů vpravo od tlačítka pro spouštění aplikací.

Po skončení běhu programu si pravděpodobně všimnete i několika změn, k nimž v IDE dojde. Je například možné, že se vám ve spodní části IDE zobrazí okno Výstup (Output), obsahující informace o tom, jak se jednotlivá sestavení celé aplikace podařilo zkompileovat a následně i spustit. Toto je chování, které mnozí z vás po kompilaci programu a jeho následném spuštění očekávají. V okně Výstup (Output) najdete poměrně podrobný přehled událostí, k nimž během kompilace došlo. Sami pak uvidíte, že proces kompilace má několik fází a v jeho průběhu dochází k načtení řady souborů a zdrojů, jimž se říká *knihovny*. Takto podrobný záznam celého procesu je užitečný zejména tehdy, je-li kompilace programu z důvodu nějaké neodhalené chyby neúspěšná.

4. Po kontrole obsahu okna Výstup (Output) můžete toto okno uzavřít klepnutím na tlačítko Zavřít (Close).

V několika následujících kapitolách se oknem Výstup (Output) příliš zabývat nebudeme. Měli byste si však zapamatovat, že pokud při psaní svých vlastních programů narazíte na nějaké nečekané chyby, může vám tento nástroj značně pomoci. Ve většině případů však toto okno pouze uzavřete, abyste si uvolnili více prostoru pro práci se samotným kódem.

Práce s oknem Vlastnosti (Properties)

Podobně jako předcházející verze i IDE sady Visual Studio 2013 obsahuje okno Vlastnosti (Properties), které vám umožňuje změny charakteristik neboli *nastavení vlastností* jednoho či více prvků uživatelského rozhraní, tvořeného nějakou stránkou. Přitom obecně lze říci, že nastavením vlastnosti se míní jistá kvalita jednoho z objektů, tvořícího součást vašeho programu. Touto kvalitou může například být umístění objektu na obrazovce, jeho velikost, text, který má být v objektu zobrazen, apod. Objekt textového bloku, v němž je zobrazen dotaz na rockový nástroj, můžete například upravit tak, že pomocí nastavení vlastností změníte písmo anebo jeho velikost.

Okno Vlastnosti (Properties) vždy obsahuje seznam vlastností pouze toho objektu, který je aktuálně vybrán v okně Návrhář (Designer). Tak například je-li v tomto okně vybrán objekt tlačítka, pak v okně Vlastnosti (Properties) uvidíte vlastnosti pouze tohoto tlačítka. Přitom první vlastnost, se kterou se v okně Vlastnosti (Properties) setkáte, je obvykle vlastnost *Name*. Tu využijete především tehdy, budete-li chtít upravovat vlastnosti používaných objektů pomocí kódu v jazyce Visual Basic. (Standardně platí, že všechny nově vytvářené objekty XAML nemají žádný název.) Ačkoliv každý objekt na stránce umožňuje nastavení mnoha vlastností, sada Visual Studio přiřazuje většině z nich nějaké výchozí hodnoty. K rychlému vyhledání těch vlastností, které chcete nastavit, můžete využít možnost jejich uspořádání podle různých kritérií. K tomuto účelu slouží rozbalovací seznam Arrange By, nacházející se v horní části okna Vlastnosti (Properties).

Nastavení vlastnosti můžete změnit přímo v okně Vlastnosti (Properties) v době, kdy pracujete na nějaké stránce; téhož výsledku však můžete docílit i editací XAML kódu dané stránky anebo můžete do souboru kódu na pozadí přidat nějaký kód v jazyce Visual Basic, který systému Windows říká, že v době běhu programu má změnit nastavení jedné či více vlastností.

Jak se dozvíte později, pomocí tlačítka Event handlers (toto tlačítko má ikonu blesku a nachází se vedle názvu) můžete změnit i kód zpracovatelů událostí objektů, obsažených na stránce. Přitom zpracovateli událostí se míní procedury ve Visual Basicu, spouštěné ve chvíli, kdy uživatel interaguje s objekty na stránce tím, že na ně například klepne, přesune je či s nimi provede nějakou jinou akci.

V následujícím cvičení si zopakujete základní postupy při nastavování vlastností. Cílem cvičení je změnit text zobrazovaný v objektu tlačítka, a dále změnit váhu a styl písma použitého v prvním textovém objektu. (Pokud si tyto postupy nepotřebujete zopakovat, můžete následující cvičení vynechat a přejít rovnou ke studiu sekce „Uspořádání nástrojů pro programování“.)

Změny vlastností

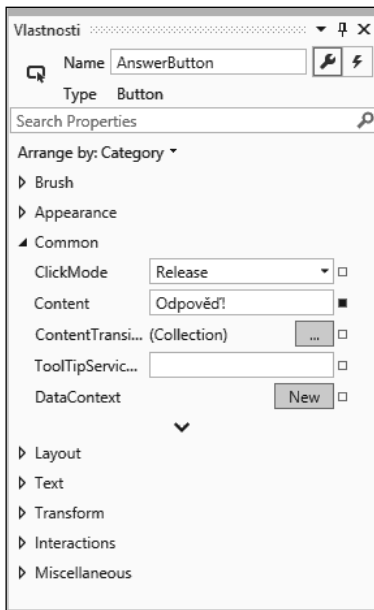
1. Na stránce, kterou máte aktuálně zobrazenou v okně Návrhář (Designer), klepněte na tlačítko Odpověď.

Chcete-li pracovat s nějakým objektem stránky, musíte jej nejprve vybrat. Jakmile jej vyberete, v okně Vlastnosti (Properties) se zobrazí všechny jeho vlastnosti.

2. Není-li okno Vlastnosti (Properties) zobrazené, stisknutím klávesy F4 si jej otevřete.

Okno Vlastnosti (Properties) může, ale nemusí být v sadě Visual Studio zobrazené ihned po jejím spuštění. Toto chování závisí na konfiguraci sady Visual Studio a na vašem používání tohoto okna. Pokud je toto okno zobrazeno, pak obvykle na pravé straně vývojového prostředí, hned pod oknem Průzkumník řešení (Solution Explorer).

Uvidíte okno podobající se následující ukázce:



V okně Vlastnosti (Properties) uvidíte přehled všech vlastností objektu vybraného tlačítka, které můžete nastavovat. Při vytváření ukázkového programu jsem toto tlačítko nazval AnswerButton. Z výše uvedeného obrázku je patrné, že vlastnosti jsou seskupeny do vnořených skupin (kategorií), přičemž ve výchozím nastavení jsou zobrazeny právě tyto kategorie. Pokud některou z kategorií rozbalíte, uvidíte přehled všech vlastností patřících

do dané kategorie. Přitom platí, že názvy vlastností jsou obvykle uvedeny vlevo, zatímco jejich hodnoty vpravo. Nastavení některých vlastností, mezi něž patří například všechny vlastnosti kategorie *Brush*, lze aktualizovat výběrem barvy z barevné palety, což znamená, že existuje více způsobů nastavování vlastností – ne vždy je tedy nutné nastavovat vlastnosti zadáváním nějakých hodnot z klávesnice.

3. V kategorii *Common* (což je kategorie obsahující typické vlastnosti objektu tlačítka) si všimněte, že vlastnosti *Content* je přiřazena hodnota *Odpověď!*

Toto je tedy text, který se nyní zobrazí na jediném tlačítku hlavní stránky aplikace. Pomocí okna *Vlastnosti* (*Properties*) můžete tento text upravit dle svých potřeb. Zkuste si nyní odstranit vykřičník a vyzkoušet si tak postup změny vlastnosti.

4. Levým tlačítkem myši klepněte do textového pole *Content*, a to až za řetězec „Odpověď!“ Vymažte vykřičník (!) a poté stiskněte *Enter*.

Hodnota vlastnosti *Content* se změní na *Odpověď* celkem na třech místech: v okně *Vlastnosti* (*Properties*), na samotné stránce v okně *Návrhář* (*Designer*) a přímo v *XAML* kódu, zobrazeném v editoru kódu.



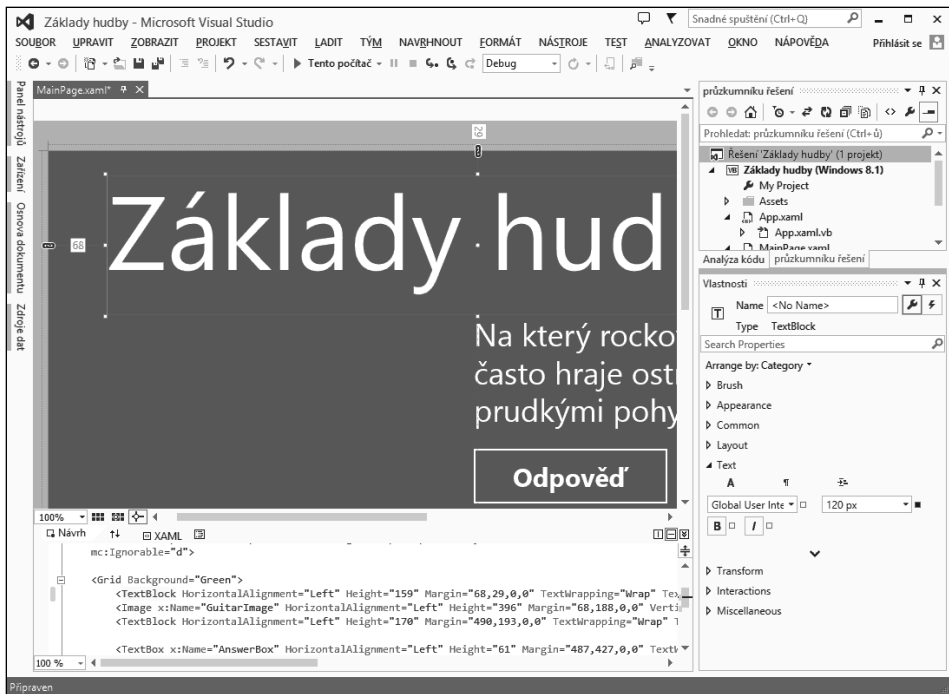
Tip: Změněnou hodnotu vlastnosti nemusíte potvrzovat stisknutím klávesy *Enter*: stačí, když jednoduše klepnete na jakékoliv jiné místo nacházející se v okně *Vlastnosti* (*Properties*). (Například tedy stačí, když klepnete do jiného textového pole.) V tomto případě si však musíte dát pozor na to, abyste omylem – při náhodném klepnutí „někam do okna“ – nezměnili nějakou jinou vlastnost.

Nyní si zkusíme změnit styl písma použitého v tom objektu textového pole, který tvoří titulek stránky. V tuto chvíli tento objekt obsahuje text *Základy hudby*. Naším cílem bude odstranění tučného písma a kurzivy.

5. V okně *Návrháře* (*Designer*) klepněte na ten objekt textového bloku, který obsahuje text *Základy hudby*. Obecně platí, že textový blok představuje ideální prostředek pro zobrazování nějakých popisných textů na stránkách vašich aplikací.
6. V okně *Vlastnosti* (*Properties*) klepněte na kategorii *Text* (nikoliv tedy na vlastnost *Text*, nacházející se v aktuálně zobrazené kategorii *Common*).
7. Klepnutím na tlačítko *Bold* (tlačítko s tučným písmenem *B*) zrušte použití tučného písma.
8. Klepnutím na tlačítko *Italic* (tlačítko s písmenem *I*, napsaným kurzivou) zrušte použití kurzivy.

Sada *Visual Studio* zaktualizuje zobrazení stránky v souladu s vámi provedenými změnami. Vaše obrazovka by nyní měla vypadat tak jako na obrázku na protější straně.

Právě jste zaktualizovali tři vlastnosti. Během celého procesu jste si možná všimli toho, že postup změn je velice podobný postupu používanému i v předcházejících verzích sady *Visual Studio*. V tomto případě však měníte *XAML* vlastnosti, související s aplikací určenou pro *Windows Store*. Tento základní postup byste si měli dobře zapamatovat – během čtení zbývajících částí této knihy budete mít ještě mnoho příležitostí ke změnám písma, jejich barev a stylů.



Objekty a vlastnosti: terminologický přehled

V této části si zopakujeme několik základních poznatků, které byste měli mít na mysli vždy, když budete v sadě Visual Studio pracovat s objekty a vlastnostmi. Zprv si zapamatujte, že každý prvek uživatelského rozhraní programu (včetně samotné stránky) obsahuje sadu nastavitelných vlastností. Přitom jednotlivé vlastnosti můžete nastavovat buď v době návrhu pomocí okna Vlastnosti (Properties), anebo úpravou vlastností v XAML kódu té stránky, která obsahuje příslušný objekt uživatelského rozhraní.

Hodnoty vlastností však mohou být měněny (či načítány) i v kódu jazyka Visual Basic, což znamená, že určité prvky programu je možné měnit i v době běhu programu. (Například ty prvky uživatelského rozhraní, které slouží pro vstup dat, velice často využívají vlastnosti k předávání zadaných dat do programu.) Vlastnosti a jejich nastavení si snáze představíte, pokud se na ně podíváte z pohledu nějakého předmětu, který denně používáte. Zkuste se zamyslet nad následující analogií s jízdním kolem, kterou jsem po řadu let používal při objasňování terminologie týkající se objektů a vlastností.

Jízdní kolo je objektem, který můžete použít k jízdě z jednoho místa do druhého. Jelikož jízdní kolo je fyzickým objektem, má několik základních a nezměnitelných charakteristik. Toto kolo má například nějakou značku, barvu, má převody, brzdy a kola a je postaveno určitým způsobem (může se jednat o silniční kolo, horské kolo či dvojkelo). V terminologii jazyka Visual Basic tyto charakteristiky jsou *vlastnostmi* objektu jízdního kola.

Přitom většina vlastností tohoto jízdního kola byla definována již v době výroby kola. Existují však některé vlastnosti jízdního kola, mezi něž patří například pláště, rychlost a volitelné doplňky typu světel či zrcátek, které se mění i v době používání kola. Jízdní kolo může mít dokonce nehmotné (tj. neviditelné) vlastnosti, k nimž se řadí datum výroby, současný vlastník,

hodnota či stav pronájmu. A aby to nebylo tak jednoduché, firma či obchod mohou vlastnit pouze jedno kolo anebo – což je podstatně pravděpodobnější – celou flotilu jízdních kol, majících rozdílné vlastnosti. Během práce ve Visual Basicu budete nastavovat vlastnosti řady různých objektů. Obecně lze říci, že práce s vlastnostmi je jedním ze základních úkonů nejen práce v sadě Visual Studio 2013, ale i v celém objektově orientovaném programování.

Uspořádání nástrojů pro programování

Aby vám Visual Studio umožnilo úplné řízení tvarů a velikostí jednotlivých prvků IDE, nabízí vám nástroje či funkce pro přesouvání a změny velikostí prakticky všech prvků uživatelského rozhraní. A ty, které nejsou právě potřebné, lze i automaticky skrývat. Práce s těmito funkcemi a nástroji patří k základním dovednostem, které budete v této knize znovu a znovu využívat. Chcete-li si zopakovat, jak se s těmito nástroji pracuje, přečtěte si následující části a proveďte všechna cvičení. Pokud si ale myslíte, že tyto postupy znáte již dostatečně dobře, následující text vynechejte a přejděte rovnou ke studiu části „Základní konfigurace IDE pro cvičení uvedená v této knize“.

Přesun a ukotvení nástrojů

Chcete-li v sadě Visual Studio přesunout jedno z oken nástrojů, pak jednoduše klepněte na jeho záhlaví a poté okno přesuňte na nové umístění. Pokud okno umístíte někam doprostřed celého vývojového prostředí a necháte ho tak být, okno se stane *plovoucím*, tj. nebude připojeno k žádnému dalšímu oknu vývojového prostředí sady Visual Studio. Přesunete-li ale okno poblíž hrany nějakého jiného okna, dojde k automatickému připojení neboli *ukotvení* přesouvaného okna k tomuto druhému oknu.

Okna, která lze ukotvit, jsou poměrně výhodnou funkcí, neboť zůstávají vždy viditelná. (Jinými slovy řečeno, nemůže se stát, že by dané okno bylo překryto jinými okny sady Visual Studio.) Potřebujete-li si zobrazit více informací z nějakého ukotveného okna, pak jednoduše uchopte jednu z hran tohoto okna a zvětšete si jej.

Chcete-li některé z oken úplně zavřít, klepněte na tlačítko Zavřít (Close), nacházející se v pravém horním rohu okna. Takto zavřené okno si vždy můžete otevřít znovu, a to výběrem příslušné volby z nabídky Zobrazit (View).

Automatické skrývání

Jestliže vám ale nevyhovuje ani úplné zavření okna, ani jeho automatické ukotvení, a potřebovali byste nějakou možnost nacházející se „někde mezi těmito dvěma“, pak zkuste *automatické skrývání* oken nástrojů na boční, spodní či horní straně okna vývojového prostředí sady Visual Studio. Pro zapnutí funkce automatického skrývání slouží malé tlačítko Automaticky skrýt (Auto Hide), nacházející se v pravém horním rohu okna hned vedle tlačítka Zavřít (Close). Ikona tohoto tlačítka připomíná malý připínáček. Touto akcí zrušíte ukotvenou pozici okna a zajistíte umístění titulku okna na straně vývojového prostředí formou neobtěžující záložky karty. Pokud zapnete funkci automatického skrývání nějakého okna, možná si všimnete toho, že okno zůstane otevřené po tu dobu, po kterou se nad jeho plochou bude nacházet ukazatel myši. Jakmile však klepnete do jiné části vývojového prostředí (či přesunete ukazatel myši), okno se skryje.

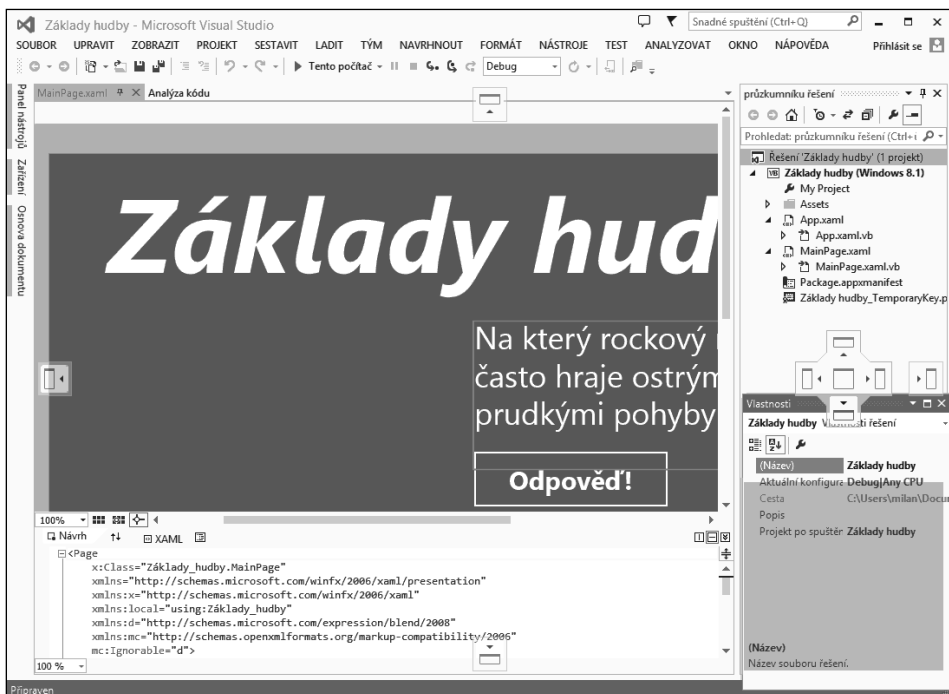
Chcete-li znovu zobrazit okno, které se mezitím automaticky skrylo, pak klepněte na záložku jeho karty, nacházející se na straně vývojového prostředí. (Okno, u něhož je zapnutá funkce automatického skrývání, poznáte poměrně snadno podle toho, že připínáček v ikoně přísluš-

ného tlačítka směřuje do strany.) Opakovaným klepáním na kartu okna na straně IDE můžete dané okno nástrojů používat v režimu, kterému říkám *režim schovávané*: tj. prvním klepnutím na kartu okna si okno otevřete, zkontrolujete si informace, které potřebujete, a druhým klepnutím na kartu okna totéž okno zase skryjete. Pokud se pak rozhodnete, že dané okno potřebujete mít zobrazené pořád, stačí vypnout režim automatického skrývání, a to opětovným klepnutím na tlačítko Automaticky skrýt (Auto Hide). Po tomto úkonu bude přepínač v ikoně tohoto tlačítka směřovat zase směrem dolů a okno bude viditelné.

Dokumenty s kartami, ruční ukotvení a vodička pro ukotvení

Další užitečnou funkcí sady Visual Studio je možnost ukotvení oken editoru kódu (Code Editor) či okna Návrhář (Designer) formou dokumentů s kartami. Přitom dokumentem s kartou se míní okno, které se z větší části skrývá za ostatními okny a zůstává z něj viditelná pouze záložka karty. Toto je výchozí způsob, jímž jsou okna s dokumenty zobrazována.

Kterýkoliv z nástrojů pro programování, jako například okno Vlastnosti (Properties), můžete však ukotvit i ručně, a to tam, kde se vám to líbí. Stačí při přesouvání okna využít vodičky pro ukotvení, zobrazovaných v podobě malých čtverečků na obvodu IDE. S ručním ukotvením oken nástrojů vám bude pomáhat i jakási růžice s vodičky, která se zobrazí ve středu IDE a bude vám naznačovat, kde vlastně bude okno nástroje zobrazeno.



Vodičky pro ukotvení jsou měnící se ikony, zobrazované na povrchu IDE ve chvíli, kdy přesouváte nějaké okno či nástroj ze stávající ukotvené pozice na pozici novou. Jelikož vodička pro ukotvení souvisí se stínovanými pravouhlymi oblastmi IDE, máte tak k dispozici náhled svých změn ještě před tím, než změny skutečně provedete. Změny v uspořádání oken se neprojeví do té doby, dokud opět neuvolníte levé tlačítko myši.

Je možné, že pro úplné zvládnutí technik pro ukotvení oken a pro jejich automatické skrývání budete potřebovat nějaký čas. Proto by bylo dobré, abyste si tyto postupy nejprve vyzkoušeli.

Skrývání oken nástrojů

Chcete-li automaticky skrýt některé z oken nástrojů, klepněte na tlačítko Automaticky skrýt (Auto Hide), jehož ikona připomíná malý přepínač. Toto tlačítko se nachází v pravém horním rohu okna, hned vedle tlačítka Zavřít (Close). Jakmile pak přesunete ukazatel myši z tohoto okna pryč, okno se skryje na stranu IDE a zůstane z něj viditelná pouze záložka. Pokud potřebujete takové okno zobrazit v jeho ukotvené pozici, pouze klepněte na jeho záložku. Kromě zmíněného tlačítka můžete též použít příkaz Automaticky skrýt (Auto Hide), nacházející se v nabídce Okno (Window). Případně můžete také klepnout pravým tlačítkem na záhlaví příslušného okna a z rozbalené místní nabídky vybrat Automaticky skrýt (Auto Hide). Při používání funkce Automaticky skrýt (Auto Hide) postupujte podle následujícího návodu.

Použití funkce Automaticky skrýt (Auto Hide)

1. V záhlaví okna Vlastnosti (Properties) vyhledejte tlačítko s ikonou přepínačku, tj. tlačítko funkce Automaticky skrýt (Auto Hide).

Jak vidíte, přepínač v tuto chvíli směřuje dolů, tj. naznačuje „připnutou“ pozici okna Vlastnosti (Properties). Současně to znamená, že automatické skrývání tohoto okna je vypnuté.

2. V záhlaví okna Vlastnosti (Properties) klepněte na tlačítko Automaticky skrýt (Auto Hide). Okno Vlastnosti (Properties) se odsune z obrazovky a bude nahrazeno malou kartou či záložkou s názvem Vlastnosti (Properties).



Poznámka: Základní výhodou povolení funkce automatického skrývání je to, že tento proces vede k uvolnění další pracovní plochy v sadě Visual Studio. Přitom skrytá okna jsou i nadále velmi rychle přístupná.

3. Klepněte na tuto záložku. Okno Vlastnosti (Properties) by se mělo velice rychle zobrazit.
4. Klepněte kamkoliv do vývojového prostředí, mimo okno Vlastnosti (Properties). Toto okno by se mělo ihned skrýt.
5. Nakonec si znovu zobrazte okno Vlastnosti (Properties) a poté v záhlaví okna klepněte na tlačítko Automaticky skrýt (Auto Hide). Okno Vlastnosti (Properties) by se mělo vrátit do své původní ukotvené pozice, díky čemuž s ním budete moci nadále pracovat, aniž byste se museli obávat jeho dalšího skrývání.

Věnujte ještě nějaký další čas zkoušení přesouvání, změn velikostí, ukotvení a automatického skrývání oken nástrojů sady Visual Studio. Zkuste si tak připravit svoji vlastní verzi ideálního pracovního prostředí. Přitom mějte na paměti, že během čtení dalších kapitol této knihy si zřejmě budete chtít tato nastavení občas upravit a přizpůsobit si tak svoji pracovní plochu novým nástrojům, s nimiž budete pracovat.

Základní konfigurace IDE pro cvičení uvedená v této knize

Podobně jako okna s nástroji a další nastavení vývojového prostředí i kompilátor a osobní nastavení sady Visual Studio 2013 jsou do značné míry přizpůsobitelné. Proto je nezbytné,

abychom si v tuto chvíli některá z těchto nastavení prošli a zajistili jejich nastavení takovým způsobem, který zajistí kompatibilitu vaší sady Visual Studio 2013 s následujícími cvičeními. V této části se také dozvíte, jak se sada Visual Studio nastavuje obecně, takže během získávání dalších programátorských zkušeností si budete schopni sadu Visual Studio upravovat tak, aby přesně odpovídala vašim potřebám.

Pokud jste sadu Visual Studio na svém počítači právě nainstalovali, jste připraveni začít pracovat na jednotlivých cvičeních. Avšak pokud je vaše instalace sady Visual Studio 2013 již poněkud starší anebo je váš počítač využíván i jinými programátory, kteří mohli změnit některá z výchozích nastavení (například na nějakém počítači v učebně), pak proveďte všechny níže popsané kroky a ujistěte se tak, že vaše nastavení týkající se projektů, řešení a kompilátoru odpovídají těm, s nimiž se v této knize pracuje.

Nastavení kompilátoru a projektů

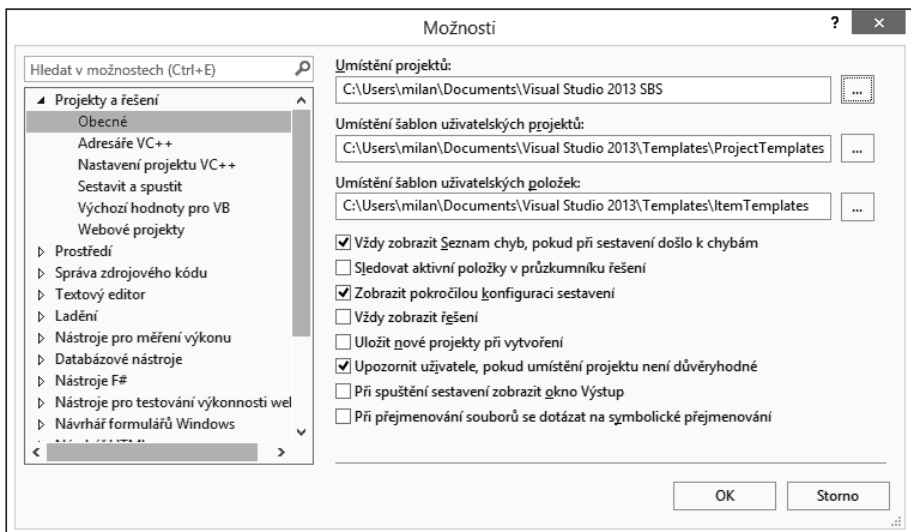
1. Klepnutím na volbu Možnosti (Options) v nabídce Nástroje (Tools) si zobrazíte dialog Možnosti (Options).

Tento dialog představuje jakousi „vstupní bránu“ k mnoha nastavením sady Visual Studio. Aby vám Visual Studio usnadnilo hledání těch nastavení, která chcete změnit, jsou nastavení uspořádána do kategorií.

2. Rozbalte kategorii Projekty a řešení (Projects And Solutions) a poté v ní klepněte na položku Obecné (General).

Tato skupina zaškrťovacích políček a dalších prvků umožňuje úpravu všech obecných nastavení sady Visual Studio, týkajících se projektů a řešení.

3. Má-li nastavení vaší sady Visual Studio 2013 odpovídat těm, s nimiž se pracuje v této knize, prohlédněte si následující obrázek a veškerá nastavení upravte podle něj:



Především vám doporučuji, abyste zrušili zaškrtnutí voleb (pokud je tedy máte zaškrtnuté) Vždy zobrazit řešení (Always show solution) a Uložit nové projekty při vytvoření (Save new projects when created). Je-li totiž zaškrtnutá první ze zmíněných voleb, budou se vám v IDE zobrazovat dodatečné příkazy týkající se řešení, které však nejsou nutné pro řešení obsahující pouze jeden projekt (což je typický případ většiny příkladů v této knize).

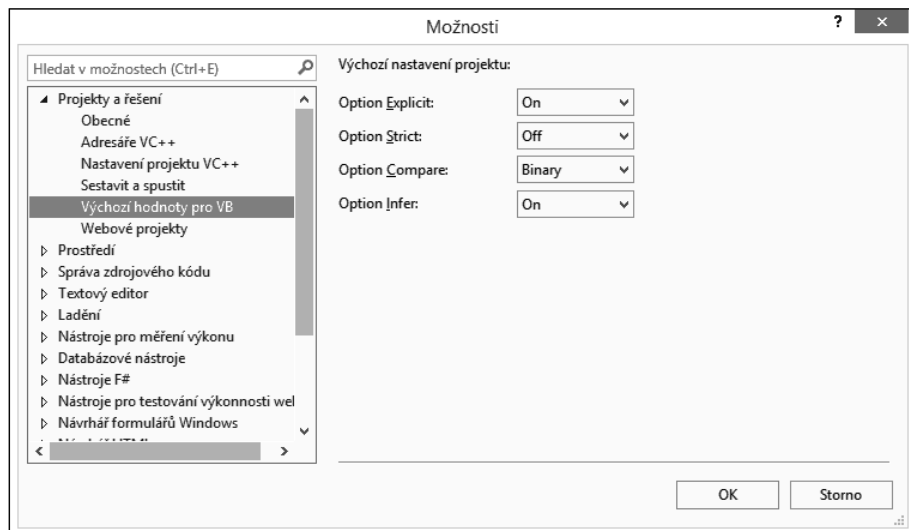
Zrušíte-li zaškrtnutí druhé z uvedených voleb, donutíte sadu Visual Studio k tomu, aby odložilo uložení projektu až do chvíle, dokud vy neklepnete na volbu Uložit vše (Save All) nabídky Soubor (File) a nezadáte cestu pro uložení souboru. Tato možnost odloženého uložení vám umožňuje vytvoření testovacího programu, jeho kompilaci, ladění, a dokonce i spuštění bez toho, že byste museli projekt nejprve uložit na disk – což je užitečná možnost zejména ve chvíli, kdy si potřebujete vytvořit jenom krátký testovací program, který možná po otestování budete chtít úplně zahodit, a ne uložit. (Analogickou situaci při práci s textovým editorem představuje případ, kdy si v editoru Word vytvoříte nový dokument, zadáte adresu pro poštovní štítek, vytisknete štítek a posléze ukončíte Word bez uložení tohoto dokumentu.) Zrušíte-li tedy tuto volbu, budete vyzýváni k uložení cvičení z této knihy až po jejich vytvoření. Samozřejmě se ale můžete rozhodnout i pro opačný přístup, tj. zaškrtnout si volbu Uložit nové projekty při vytvoření (Save new projects when created) a uložit každý z projektů ihned během jeho vytváření.

Možná jste si také stihli všimnout, že v horním políčku je uvedená cesta k souborům s příklady z této knihy (Visual_Basic_KzK), čímž vám naznačují výchozí umístění pro tyto příklady. Většina projektů, které vytvoříte, bude uložena právě v této složce a bude mít v názvu prefix „Muj“ či „Moje“, čímž je odlišíte od dokončených projektů, které jsem pro vás připravil a které si můžete prozkoumat. (Nezapomeňte cestu v tomto poli upravit tak, aby odpovídala umístění souborů s příklady z této knihy na vašem počítači.)

Po úpravě těchto nastavení můžeme přejít ke kontrole nastavení kompilátoru jazyka Visual Basic.

4. Klepněte na položku Výchozí hodnoty pro VB (VB Defaults), nacházející se v rozbalené sekci Projekty a řešení (Projects And Solutions).

Visual Studio vám nabídne celkem čtyři nastavení kompilátoru: Option Explicit, Option Strict, Option Compare a Option Infer.



Ačkoliv podrobný popis uvedených voleb přesahuje rámec této kapitoly, měli byste se ujistit, že volba Option Explicit je nastavena na On a volba Option Strict je nastavena na Off – což jsou ostatně výchozí nastavení pro programování v jazyce Visual Basic v sadě Visual Studio. Přiřazení hodnoty On volbě Option Explicit znamená, že ve svých programech

budete muset všechny proměnné nejprve deklarovat a teprve pak je budete moci použít – to je velice dobrá programátorská praxe, kterou vám mohu jen doporučit. A přiřazení hodnoty Off volbě Option Strict znamená, že za určitých podmínek budete moci vzájemně kombinovat proměnné a objekty různých typů, aniž by kompilátor hlásil nějakou chybu. (Tak například číslo může být přiřazeno objektu textového pole bez jakéhokoliv chybového hlášení.) Ačkoliv toto je programátorská praxe, která není zcela ideální a je vhodné se nad ní zamyslet, příslušné nastavení je pro určité typy ukázkových programů poměrně užitečné.

Volba Option Compare definuje metodu porovnávání, která bude použita v případě porovnávání a třídění různých textových řetězců. Více informací o porovnávání řetězců a třídění textu naleznete v kapitole 14 „Používání polí, kolekcí a generických typů pro práci s daty“ a následně i kapitole 15 „Inovativní práce s daty, založená na jazyce LINQ“.

Volba Option Infer byla novinkou jazyka Visual Basic 2008. Přiřadíte-li volbě Option Strict hodnotu Off a volbě Option Infer hodnotu On, budete moci deklarovat proměnné, aniž byste museli explicitně uvádět jejich datové typy. Případně to můžeme říci také tak, že použijete-li takovou deklaraci, pak kompilátor jazyka Visual Basic odvodí datový typ (či provede nejlepší možný odhad datového typu) na základě prvního přiřazení, které s danou proměnnou provedete. Více informací o této funkci najdete v kapitole 11 „Datové typy, operátory a práce s řetězci“.

Obecně lze říci, že volbu Option Infer byste měli mít nastavenou na Off, abyste se tak vyhnuli poměrně nepředvídatelným výsledkům, souvisejícím s použitím proměnných ve vašem programu. Při vytváření většiny ukázkových příkladů jsem tuto volbu měl nastavenou na Off.

5. Prohlédněte si další volby, které naleznete v dialogu Možnosti (Options) a které souvisí s prostředím pro programování v sadě Visual Studio. Poté klepnutím na tlačítko OK uzavřete dialog Možnosti (Options).

Nyní můžete ukončit práci v sadě Visual Studio a začít programovat.

Ukončení práce v sadě Visual Studio

Jste-li rozhodnutí, že v daném dni „už toho bylo dost“, a chcete ukončit práci v sadě Visual Studio, pak uložte všechny otevřené projekty a zavřete celé vývojové prostředí.

Ukončení práce v sadě Visual Studio

1. Na panelu nástrojů Standardní (Standard) klepněte na tlačítko Uložit vše (Save All) a uložte tak všechny změny, které jste ve svém programu provedli.
Ve svém projektu jste přece něco změnili, a proto byste si tyto změny měli uložit.
2. V nabídce Soubor (File) klepněte na volbu Konec (Exit).
Okno sady Visual Studio se uzavře a její běh skončí. V tuto chvíli jste připraveni začít vytvářet nějaký program od začátku, což je tématem kapitoly 3.

Shrnutí

V této kapitole jste se seznámili se sadou Visual Studio 2013 a s integrovaným vývojovým prostředím, které budete používat k otevírání a spuštění programů napsaných v jazyce Visual Basic. Aplikace pro různé platformy Windows můžete vytvářet otevíráním nových či stávajících projektů v sadě Visual Studio a následným přidáváním dalších funkcí pomocí různých nástrojů pro programování. V této kapitole jsme vám ukázali, jak si můžete zobrazit uživatelské rozhraní aplikace navržené pro Windows Store, jak si můžete prohlédnout XAML objekty na její stránce a jak můžete měnit nastavení jednotlivých vlastností.

Při procházení integrovaným vývojovým prostředím sady Visual Studio jste si zopakovali postup otevření a spuštění aplikace, kontroly kódu v jazyce XAML v okně editoru kódu (Code Editor) a manipulace s okny nástrojů v IDE. Kromě toho jste se dozvěděli, jak si pomocí příkazu Možnosti (Options), který je součástí nabídky Nástroje (Tools), můžete upravit nastavení celé sady Visual Studio.

IDE sady Visual Studio je poměrně rušným místem, a proto jste se v této kapitole seznámili i s několika technikami, které vám usnadní úspěšné dokončení všech následujících cvičení. Je však nutné zdůraznit, že toto vše jsou pouhé základy: chcete-li IDE sady Visual Studio zvládnout na skutečně mistrovské úrovni, měli byste jej ještě chvíli studovat.

V následující kapitole si vytvoříte svoji první aplikaci pro Windows Store od samého začátku, přičemž touto aplikací bude automatová hra hledání šťastného čísla.

Vytvoření první aplikace pro Windows Store

Po dokončení studia této kapitoly budete schopni

- Navrhnout uživatelské rozhraní aplikace určené pro Windows Store
- Používat ovládací prvky pro XAML, které jsou součástí panelu nástrojů
- Pracovat s náhodnými čísly, digitálními fotografiemi a zvukovými efekty
- Napsat v jazyce Visual Basic kód zpracovatele událostí
- Vytvořit spouštěcí obrazovku aplikace určené pro Windows Store
- Uložit, otestovat a sestavit aplikaci určenou pro Windows Store

Jak jste se dozvěděli v kapitole 2 „Integrované vývojové prostředí Visual Studio“, IDE sady Microsoft Visual Studio 2013 je připraveno pomoci vám s vytvářením aplikací v jazyce Visual Basic. V této kapitole se do tohoto tématu plně ponoříme a zkusíme si ve Visual Basicu vytvořit program určený pro Windows Store. A protože cvičení, které zde najdete, je úplné, znamená to, že tato kapitola popisuje všechny základní kroky, které budete muset provést vždy, když budete chtít ve vývojovém prostředí sady Visual Studio 2013 vytvořit aplikaci pro Windows Store. V dalších kapitolách se budeme spíše věnovat odlišnostem jednotlivých typů aplikací, které lze v sadě Visual Studio vytvářet: jak již víte, patří k nim aplikace pro Windows Store, aplikace pro Windows desktop, konzolové aplikace, webové aplikace a aplikace pro Windows Phone. Jakmile se naučíte základní dovednosti programování v jazyce Visual Basic, zjistíte, že všechny zmíněné typy aplikací mají mnoho společného.

V této kapitole si ukážeme, jak se vytváří automatová hra stylu Las Vegas pro Windows Store. Uživatelské rozhraní programu budete navrhovat pomocí ovládacích prvků pro práci s XAML, které jsou součástí Panelu nástrojů (Toolbox). Nastavení vlastností a velikosti jednotlivých objektů na stránce pak budete měnit pomocí nástrojů, které jsou součástí IDE. Během programování budete pracovat s ovládacími prvky typu *TextBlock* pro zobrazení náhodných čísel, ovládacími prvky typu *ImageControl* ke vkládání digitálních fotografií a ovládacími prvky typu *MediaElement* k přehrávání zvukových efektů v případě, že uživatel vytočí číslo 7. Při vytváření základní funkcionality této aplikace určené pro Windows Store napíšete v jazyce Visual Basic kód potřebného zpracovatele událostí. Dále navrhnete i spouštěcí obrazovku aplikace, aplikaci poté uložíte a otestujete ji v IDE. A v samotném závěru kapitoly aplikaci sestavíte, tj. vytvoříte spustitelný soubor, který bude možné spustit z obrazovky Start systému Windows.

Šťastná sedma: aplikace napsaná ve Visual Basicu pro Windows Store

Aplikace pro Windows Store, kterou budete ve Visual Basicu vytvářet, se jmenuje Šťastná sedma. Jedná se vlastně o hru simulující automatovou hru hledání šťastného čísla. Šťastná sedma má jednoduché uživatelské rozhraní a pomocí sady Visual Studio 2013 ji lze vytvořit a zkompileovat během několika málo minut. Níže vidíte ukázkou programu po jeho úplném dokončení:



Programování krok za krokem

Uživatelské rozhraní hry Šťastná sedma obsahuje jedno tlačítko, tři objekty textových bloků pro zobrazování šťastných čísel, digitální fotografii, lákající hráče ukázkou výhry, a textový blok obsahující název hry, tj. text „Šťastná sedma“. Všechny tyto prvky jsem připravil tak, že jsem na hlavní stránce hry Šťastná sedma vytvořil celkem 5 viditelných objektů, u nichž jsem posléze změnil některé vlastnosti. Kromě toho jsem na stránku přidal i ovládací prvek typu *MediaElement*, který v době běhu není viditelný a zajišťuje přehrání speciálního zvukového efektu pouze tehdy, pokud uživatel hru vyhraje.

Po návrhu základního uživatelského rozhraní jsem k tlačítku Vytoč přidal kód zajišťující zpracování události klepnutí na tlačítko myši ze strany uživatele a následné zobrazení náhodných čísel na stránce. Nakonec jsem vytvořil spouštěcí obrazovku aplikace a celou aplikaci jsem pomocí nástrojů, které jsou součástí IDE sady Visual Studio, připravil k distribuci.

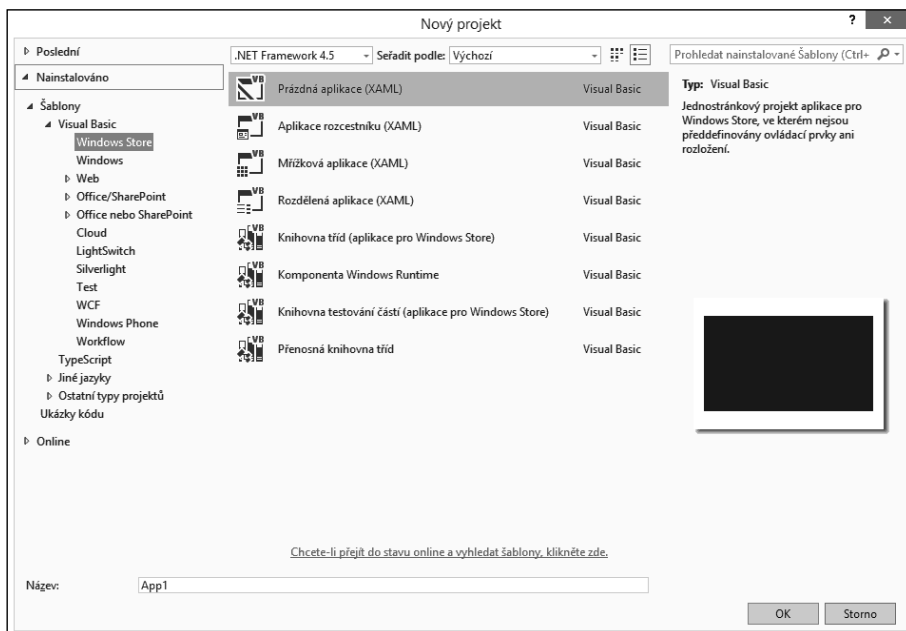
Při vytváření hry Šťastná sedma provedete pět základních programátorských kroků, shodných pro naprostou většinu aplikací, které budete v sadě Visual Studio připravovat. Nejprve pomocí ovládacích prvků, které jsou součástí Panelu nástrojů (Toolbox), vytvoříte uživatelské rozhraní. Následně upravíte nastavení důležitých vlastností. Poté přejdete k psaní kódu ve Visual Basicu. Dalším krokem bude příprava spouštěcí obrazovky aplikace a ostatních potřebných prvků. A nakonec celou aplikaci otestujete a sestavíte spustitelný soubor.

Návrh uživatelského rozhraní

V tomto cvičení zahájíte vývoj hry Šťastná sedma tím, že nejprve vytvoříte nový projekt a následně pomocí ovládacích prvků pro práci s XAML sestavíte uživatelské rozhraní.

Vytvoření nového projektu

1. Spustíte Visual Studio 2013.
2. Z nabídky Soubor (File) sady Visual Studio zvolíte Nový projekt (New Project). Spustí se dialog Nový projekt (New Project), jehož ukázkou vidíte níže:



Dialogové okno Nový projekt (New Project) vám nabízí přístup ke hlavním typům šablon obsažených v sadě Visual Studio a určených pro vytváření jednotlivých typů aplikací v této sadě. Na levé straně tohoto okna vidíte seznam dostupných hlavních typů šablon. Jelikož při poslední práci s tímto dialogem jsem v něm zvolil Visual Basic, jsou nyní automaticky zobrazeny šablony právě pro tento jazyk. Pokud si však rozbalíte sekci Jiné jazyky (Other languages), uvidíte mnoho šablon a dalších prostředků i pro jiné jazyky, mezi něž patří například Visual C#, Visual C++ či Java Script.

3. V podsekcí Windows Store sekce Visual Basic klepněte na Prázdná aplikace (XAML) (Blank App /XAML/). Je však možné, že tato volba bude již přednastavena.

Použijete-li šablonu Prázdná aplikace (Blank App), Visual Studio vytvoří projekt základní aplikace pro Windows Store, obsahující výchozí dlaždice, spouštěcí obrazovku, manifest a spouštěcí kód. Součástí této aplikace však nebudou žádné předdefinované ovládací prvky či rozvržení. Všimněte si také, že vám tento dialog nabízí možnost vytvoření dalších typů aplikací (k nimž se dostaneme později), mezi něž patří aplikace pro Windows desktop, webové aplikace a aplikace pro Windows Phone.

4. Do textového pole Název (Name) zadejte **Moje šťastná sedma**.

Visual Studio přiřadí vašemu projektu název *Moje šťastná sedma*. (Složku pro uložení projektu budete zadávat později.)



Důležité: Doporučuji vám, abyste při práci na tomto projektu použili prefix „Moje“. Zabráníte tak případnému nechtěnému zaměňování své aplikace s tou, kterou jsem pro vás připravil. Jak však sami uvidíte, tento prefix nebudu používat ani v instrukcích, ani v ukázkových projektech a ani jej neuvídíte v ukázkách obrazovek v knize – tento prefix nechávám pro vaše použití.

Obsahuje-li dialog *Nový projekt (New Project)* textová pole *Umístění (Location)* a *Název řešení (Solution Name)*, musíte nyní zadat název složky a řešení svého nového projektu. V části „Základní konfigurace IDE pro cvičení uvedená v této knize“ kapitoly 2 se dozvíte, jak si máte upravit nastavení vývojového prostředí, pokud se tato textová pole objeví. Jak jsem vám řekl již v kapitole 2, o zadání umístění vás budu žádat vždy při prvním uložení projektu – což je krok, který se obvykle nachází poblíž konce každého cvičení.

5. Klepnutím na tlačítko OK vytvoříte v sadě Visual Studio nový projekt.

Visual Studio připraví IDE pro nový projekt a zobrazí kód jazyka Visual Basic, související s šablonou prázdné aplikace. Vaše obrazovka by nyní měla vypadat zhruba takto:

```

App.xaml.vb
(Obecný)
[Deklarace]
'Sablona "Blank Application" je popsána na adrese http://go.microsoft.com/fwlink/?LinkId=234227

''' <summary>
''' Poskytuje chování specifické pro aplikaci, doplňující výchozí třídu Application.
''' </summary>
NotInheritable Class App
    Inherits Application

    ''' <summary>
    ''' Vytvořeno při normálním spuštění aplikace koncovým uživatelem. Ostatní vstupní body
    ''' budou použity při spuštění aplikace pro otevření určitého souboru, ke zobrazení
    ''' výsledků hledání a tak dále.
    ''' </summary>
    ''' <param name="e">Podrobnosti o požadavku spuštění a procesu.</param>
    Protected Overrides Sub OnLaunched(e As Windows.ApplicationModel.Activation.LaunchActivatedEventArgs)

    #IF DEBUG Then
        Zobrazovat během ladění informace o profilování a grafiky.
        If System.Diagnostics.Debugger.IsAttached Then
            ' Zobrazit počítadla aktuální frekvence snímků
            Me.DebugSettings.EnableFrameRateCounter = True
        End If
    #End If

    Dim rootFrame As Frame = TryCast(window.Current.Content, Frame)

    ' Neopakovat inicializaci aplikace, pokud má objekt Window již obsah,
    ' pouze zkontrolovat, zda je okno aktivní

    If rootFrame Is Nothing Then
        ' Vytvořit objekt Frame, který bude fungovat jako kontext navigace, a spustit procházení první strdy
        rootFrame = New Frame()
        ' Nastavte výchozí jazyk.
        rootFrame.Language = Windows.Globalization.ApplicationLanguages.Languages(0)

        AddHandler rootFrame.NavigationFailed, AddressOf OnNavigationFailed

        If e.PreviousExecutionState = ApplicationExecutionState.Terminated Then
            ' TODO: Načíst stav z dříve pozastavené aplikace
        End If
    
```

To, co zde vidíte, je standardní spouštěcí kód aplikace určené pro Windows Store a vytvořené v sadě Visual Studio 2013. Tento kód je uložen v souboru *App.xaml.vb*, tvořícím součást celého projektu. Ačkoliv každý projekt obsahuje soubor *App.xaml*, vaše práce v této kapitole začne úpravami uživatelského rozhraní aplikace, uloženého v souboru *MainPage.xaml*.



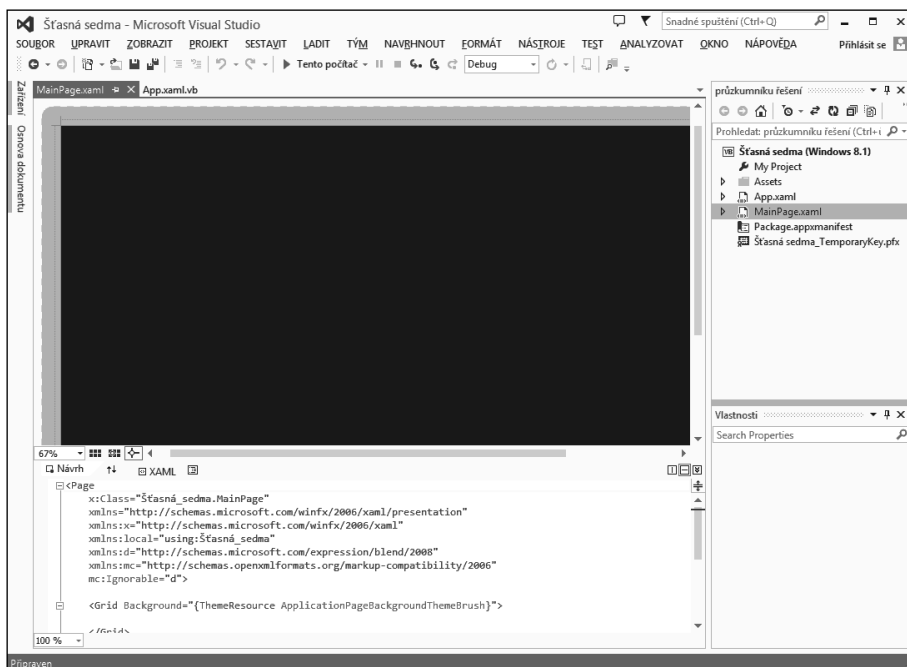
Poznámka: Sekce začínající `#If DEBUG Then` a nacházející se téměř uprostřed této obrazovky zajišťuje zobrazování informací pro ladění na obrazovce ve chvíli, kdy je aplikace pro Windows Store spuštěna v režimu ladění. To znamená, že tato sekce je navržena pro účely testování. Tento kód začal být do aplikací přidáván až v konečné verzi sady Visual Studio 2013. Jeho hlavním smyslem je zobrazování informací o tom, kolik času zabírají různé úlohy, prováděné při běhu aplikace navržené pro Windows Store. Kromě jiného k těmto informacím patří snímková frekvence vlákna uživatelského rozhraní a čas (v milisekundách) potřebný pro načtení uživatelského rozhraní. Chcete-li potlačit zobrazování informací pro ladění, pak odstraňte veškerý kód nacházející se mezi příkazy `#If DEBUG` a `#End If`. Více informací o významu čítačů ladění, zobrazovaných na horním okraji obrazovky během testování, najdete na adrese <http://msdn.microsoft.com> po zadání hesla `EnableFrameRateCounter`.

Nyní si uživatelské rozhraní zobrazíte v Návrhář (Designer) a pomocí Panelu nástrojů (Toolbox) jej rozšíříte o ovládací prvky.

Navigace v Návrhář (Designer)

1. Není-li zobrazený, otevřete si Průzkumník řešení (Solution Explorer). Poté poklepejte na soubor `Main.xaml`.

Visual Studio otevře soubor `Main.xaml` v okně Návrháře (Designer) a zobrazí horní levý roh hlavní stránky aplikace. Pod touto stránkou uvidíte editor kódu (Code Editor) s několika řádky značkovacího jazyka XAML, souvisejícími se stránkou uživatelského rozhraní, zobrazenou v okně Návrháře (Designer). Jakmile začnete v okně Návrháře (Designer) přidávat na stránku aplikace jednotlivé ovládací prvky, příslušné změny se ihned projeví i v editoru kódu (Code Editor), v němž se zobrazí příkazy jazyka XAML sloužící k vytvoření přidávaných prvků uživatelského rozhraní. Vaše obrazovka by nyní měla vypadat zhruba takto:



Vždy, když v sadě Visual Studio a v jazyce Visual Basic vytvoříte novou aplikaci pro Windows Store, budete pro návrh uživatelského rozhraní používat ovládací prvky Panelu nástrojů (Toolbox) a značkovací jazyk XAML. Tato technika bude novinkou pro ty z vás, kteří byli zvyklí v jazyce Visual Basic vytvářet aplikace pro Windows pomocí technologie, jíž se říká Windows Forms. (Určitě jste při tom používali Panel nástrojů /Toolbox/, nikoliv však XAML.) Použití jazyka XAML však může být známé těm programátorům, kteří při vytváření aplikací pro Windows využívali technologii Windows Presentation Foundation (WPF) anebo vytvářeli aplikace pro Windows Phone.

A nyní se podívejme na to, jak se pracuje v Návrháři (Designer).

2. Klepněte na ukazatel svislého posuvníku Návrháře (Designer) a přesuňte jej dolů.

Posunete-li v okně Návrháře (Designer) posuvník, zobrazíte si tak další části uživatelského rozhraní, na němž právě pracujete.

3. Klepněte na ukazatel vodorovného posuvníku Návrháře (Designer) a přesuňte jej doprava. (Analogicky platí, že pokud přesunete vodorovný posuvník, „odhalí“ se vám dosud skryté části uživatelského rozhraní.)

Poblíž levého spodního rohu okna Návrháře (Designer) byste měli vidět nástroj Zoom, umožňující zvětšování (tuto možnost využijete v případě, že si budete chtít zobrazit více detailů) či naopak zmenšování (tuto možnost využijete v případě, že si budete chtít zobrazit stránku jako celek) měřítko zobrazení stránky. Aktuální hodnota, kterou v tomto nástroji nyní vidíte, je 67%. Jiné měřítko zobrazení si můžete nastavit klepnutím na rozbalovací seznam nástroje Zoom a následným výběrem jedné z nabízených hodnot.

4. Klepnutím si rozbalte seznam Zoom a vyberte si z něj hodnotu Fit All.

V okně Návrháře (Designer) se nyní zobrazí celá stránka aplikace. V závislosti na rozlišení vaší obrazovky a na velikosti těch částí obrazovky, které jste si vyhradili pro ostatní části IDE, uvidíte poněkud menší verzi stránky.



Tip: Je-li vaše myš vybavena kolečkem, můžete měřítko zobrazení poměrně rychle měnit tak, že stisknete klávesu Ctrl a současně budete otáčet kolečkem. Tato funkce funguje vždy, je-li aktivní okno Návrháře (Designer).

Při tvorbě stránky aplikace je nezbytné, abyste byli schopní si rychle zobrazit různé části dané stránky v různých velikostech. Někdy si chcete zobrazit celou stránku, neboť potřebujete posoudit celkové rozmístění jednotlivých ovládacích a dalších prvků, zatímco jindy si chcete zobrazit jen určitý detail stránky. Je tedy na vás, abyste si přizpůsobili okno Návrháře (Designer) tak, abyste v době vývoje byli schopní si zobrazit všechny části uživatelského rozhraní.

Nyní zobrazení v okně Návrháře (Designer) změníme tak, aby odpovídalo měřítku 1:1.

5. Klepnutím si rozbalte seznam Zoom a vyberte si z něj hodnotu 100%.
6. Svislý a vodorovný posuvník okna Návrháře (Designer) přesuňte tak, abyste viděli horní levý roh stránky.

Zobrazení okraje stránky vám pomůže zorientovat se na stránce tak, jak se na ní bude orientovat i uživatel.

A nyní zkusíme na stránku přidat jeden z ovládacích prvků nacházejících se na Panelu nástrojů (Toolbox).

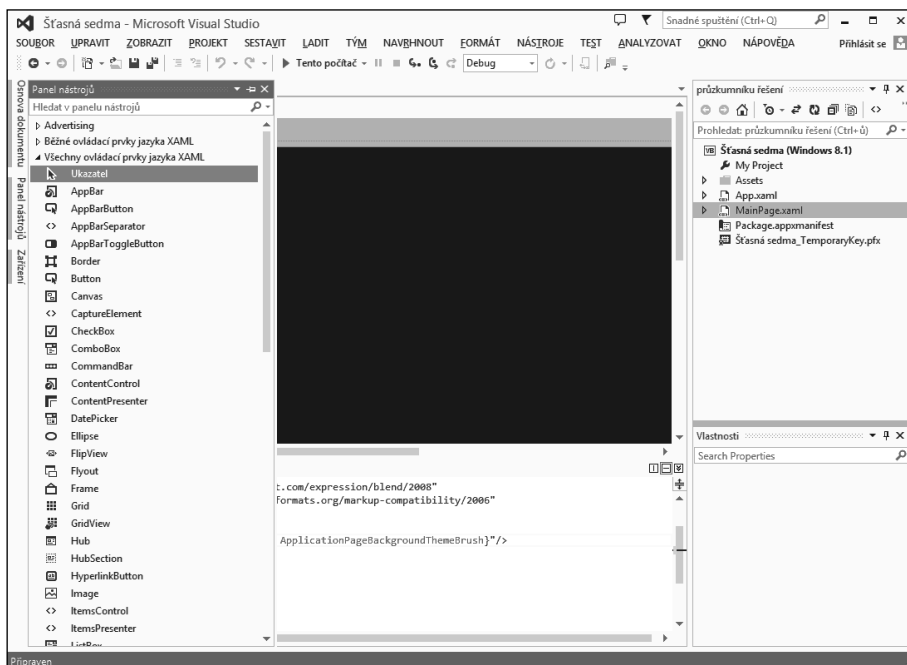
Otevření Panelu nástrojů (Toolbox) a použití ovládacího prvku TextBlock

1. Není-li Panel nástrojů (Toolbox) dosud zobrazený, klepněte na kartu Panel nástrojů (Toolbox), případně z nabídky Zobrazit (View) vyberte Panel nástrojů (Toolbox).

Okno Panelu nástrojů (Toolbox) obsahuje rozsáhlou kolekci ovládacích prvků uživatelského rozhraní, které můžete do svých aplikací přidávat. A jelikož vytváříte aplikaci pro Windows Store určenou pro systém Windows 8.1, jsou v Panelu nástrojů (Toolbox) zobrazeny ovládací prvky toho typu, kterému se říká ovládací prvky XAML – tj. strukturované prvky řídicí vzhled a ovládání aplikace, které lze na stránce úspěšně uspořádat pomocí parseru jazyka XAML, jenž je součástí sady Visual Studio.

Samozřejmě existují i další kolekce ovládacích prvků, zobrazované v Panelu nástrojů (Toolbox) ve chvíli, kdy vytváříte jiné typy aplikací (ovládací prvky Windows Forms, ovládací prvky HTML pro webové aplikace, ovládací prvky pro Windows Phone a další), avšak touto skutečností se nyní nemusíte příliš zabývat – sada Visual Studio totiž automaticky načte do Panelu nástrojů (Toolbox) správnou sadu ovládacích prvků ve chvíli, kdy zahájíte vývoj nového řešení.

Vaše obrazovka by nyní měla vypadat zhruba takto:



Z důvodu snahy o zvýšení přehlednosti byly ovládací prvky v Panelu nástrojů (Toolbox) rozděleny do několika skupin: Advertising, Běžné ovládací prvky jazyka XAML (Common XAML Controls), což jsou ovládací prvky, objevující se v mnoha aplikacích, a Všechny ovládací prvky jazyka XAML (All XAML Controls) což je skupina, obsahující všechny ovládací prvky XAML určené pro Windows Store aplikace a nakonfigurované pro použití v sadě Visual Studio.

Nezapomeňte, že okno Panelu nástrojů (Toolbox) se v IDE sady Visual Studio chová jako každé jiné okno nástrojů. To znamená, že jej zcela v souladu se svými potřebami můžete

přesouvat, přichytávat či měnit jeho velikost. Můžete se také rozhodnout, že po dobu přidávání ovládacích prvků na novou stránku budete chtít mít okno Panelu nástrojů (Toolbox) otevřené, což zajistíte tak, že toto okno přichytíte ke straně IDE. Je však možné, že vám bude více vyhovovat funkce automatického skrývání okna Panelu nástrojů (Toolbox), díky níž se toto okno zavře po výběru a přidání každého ovládacího prvku.

2. V Panelu nástrojů (Toolbox) klepněte na ovládací prvek *TextBlock* a poté přesuňte ukazatel myši do okna Návrhář (Designer).

Ukazatel myši se změní na křížek. Tento křížek byl navržen tak, aby vám usnadnil kreslení pravouhlého tvaru ovládacího prvku *TextBlock* na stránce. Pokud na ovládací prvek *TextBlock* v okně Panelu nástrojů (Toolbox) poklepete, automaticky se bude vytvářet ovládací prvek tohoto typu, mající výchozí velikost.

3. Klepnutím a následným tažením vytvořte na stránce velký pravouhlý objekt textového bloku, vyplňující horní levý roh stránky.

Jakmile uvolníte tlačítko myši, sada Visual Studio vytvoří objekt XAML textového bloku. Obecně platí, že ovládací prvek typu *TextBlock* je navržen pro zobrazování textu na stránce. V našem případě jej využijeme k zadání nějakého vítacího textu pro uživatele vaší aplikace. Text uložený v objektu typu *TextBlock* umístěném na stránce, můžete měnit pomocí vlastnosti *Text*, a to buď v okně Vlastnosti (Properties), nebo v kódu značkovacího jazyka XAML, anebo přímo v kódu programu.

4. V okně Vlastnosti (Properties) si najděte vlastnost *Text* a přiřaďte jí řetězec **Šťastná sedma**. Následně stiskněte klávesu Enter.

Sada Visual Studio zobrazí text „Šťastná sedma“ jak v okně Vlastnosti (Properties), tak i v okně Návrhář (Designer). Nyní zvětšíte velikost použitého písma a nastavíte další formátování tohoto textu.

5. V okně Vlastnosti (Properties) přejděte do kategorie *Text*. Zde klepněte na pole *Font Size*, zadejte hodnotu **98** a stiskněte Enter.

V textovém poli *Font Size* lze pomocí šipek nahoru a dolů vybírat mezi širokou škálou hodnot velikostí písma počínaje 6 a konče 72. V tomto případě však chceme ručně zadat opravdu velké číslo, aby výsledný efekt nápisu byl značný.



Tip: Kterýkoliv objekt stránky můžete kdykoliv vymazat tak, že jej vyberete a následně stisknete klávesu Delete. Poté můžete celý objekt vytvořit znovu. Vyzkoušejte si postup vytváření a mazání objektů a procvičte si tak jednu z částí procesu tvorby uživatelského rozhraní.

6. V okně Vlastnosti (Properties) přejděte do kategorie *Brush*. Poté klepněte na vlastnost *Foreground*, pokud již není vybrána.

Vlastnost *Foreground* řídí barvu textu v objektu typu textový blok.

7. Klepněte na tlačítko Solid Color Brush.

Tlačítko Solid Color Brush je druhou dlaždicí zleva, nacházející se poblíž horního okraje dialogového okna. (Toto tlačítko může být výchozí volbou, a proto již může být vybráno; nic se však nestane, pokud na něj klepnete ještě jednou.)

Je-li vybráno tlačítko Solid Color Brush, zobrazí se vám editor Color Resources.

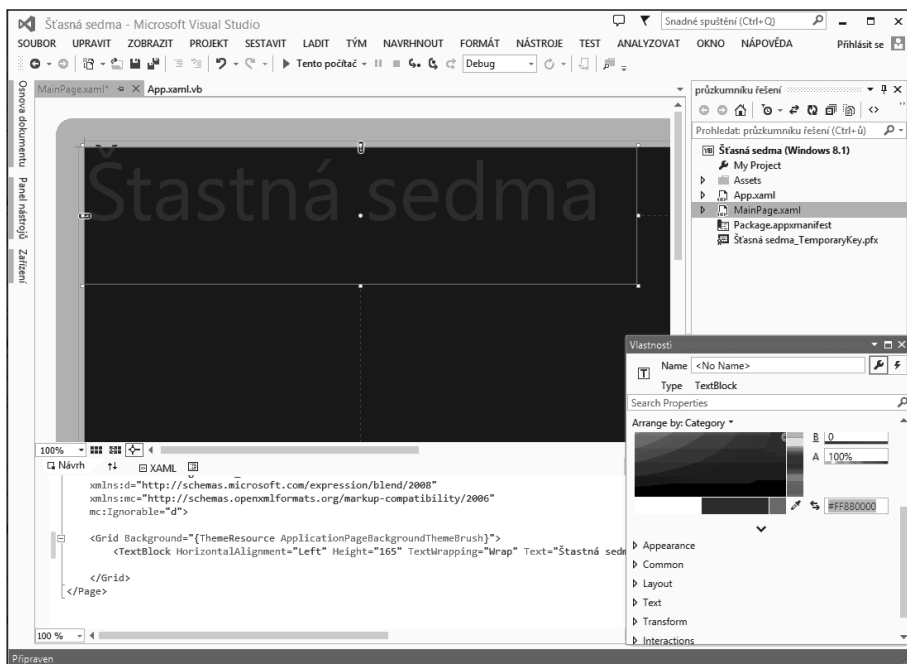
8. Potřebujete-li pro zobrazení obsahu okna Vlastnosti (Properties) více místa, zvětšíte si toto okno anebo si jej nakonfigurujete jako plovoucí okno. Důležité je, abyste viděli celý editor Color Resources.

9. Poblíž spodního okraje editoru vyberte číslo, na jehož začátku je znak mřížky (#).

Toto číslo, tvořené 8 číslicemi, je známé jako hexadecimální hodnota barvy – tj. číslo vyjádřené pomocí šestnáctkové číselné soustavy a určující barvu pomocí hodnot RGBA. Při zadávání nové barvy textu můžete určit jednotlivé hodnoty červené, zelené a modré (R, G a B) anebo můžete použít jeden ze standardizovaných názvů, jako například Red, DarkRed, White, Black, Purple, Lime či Aquamarine.

10. Do pole zadejte text **DarkRed** a stiskněte Enter.

Všimněte si, že jakmile v editoru Color Resources stisknete Enter, sada Visual Studio převede text „DarkRed“ na hodnotu #FF8B0000. To je ostatně zřetelné i z následující obrazovky:



11. Pokud jste si nějak upravili velikost okna Vlastnosti (Properties), vraťte jej nyní do původní ukotvené polohy.

Nyní na stránku přidáte další tři ovládací prvky typu *TextBlock*, které budou umístěné pod hlavičkou Šťastná sedma a budou sloužit k zobrazování náhodných čísel, která hra vybere. Kdykoliv uživatel klepne na tlačítko Vytoč, v těchto textových blocích se objeví tři nová čísla. Bude-li jedno z těchto čísel rovno 7, uživatel vyhrává a hra přehraje zvuk.

Přidání textových bloků pro náhodná čísla

1. V okně Panelu nástrojů (Toolbox) poklepejte na ovládací prvek *TextBlock*.

Sada Visual Studio vytvoří na stránce objekt textového bloku. Výsledný objekt bude sice poměrně malý, ale vy můžete jeho velikost změnit.

2. V okně Vlastnosti (Properties) přejděte do kategorie *Text*, klepněte na pole *Font Size*, zadejte 72 a nakonec stiskněte Enter.

Sada Visual Studio zvětší objekt textového bloku tak, aby do něj bylo možné zapisovat text pomocí písma majícího velikost 72.

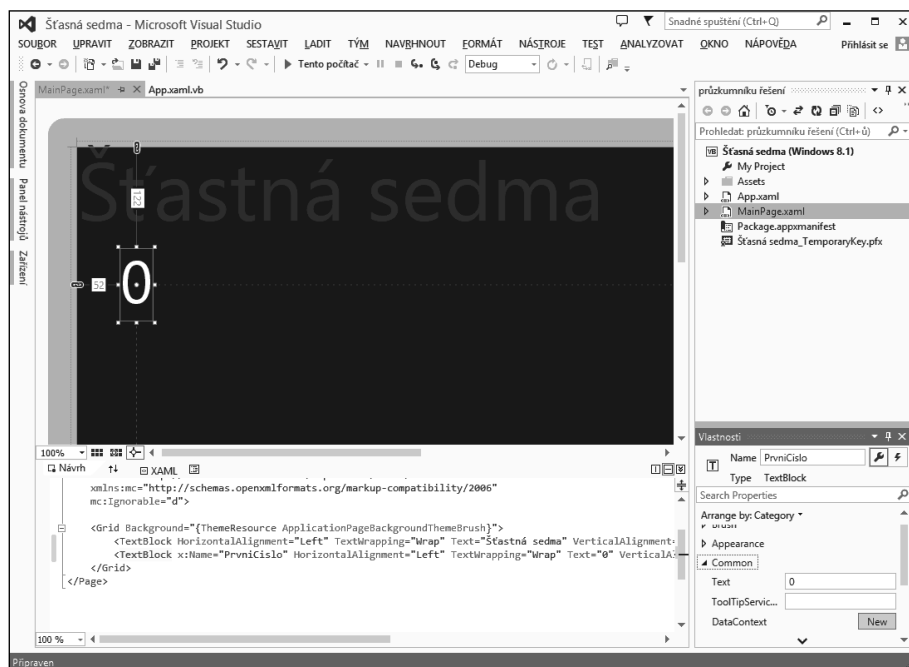
3. V okně Vlastnosti (Properties) rozbalte kategorii *Common* a klepněte na pole *Text*. Do tohoto pole zadejte hodnotu **0** a stiskněte Enter.

0 bude počáteční hodnotou prvního šťastného čísla našeho programu.

4. Přešuněte se do horní části okna Vlastnosti (Properties) a do pole *Name* zadejte název objektu textového bloku **PrvniCislo**.

Není nutné, aby všechny objekty uživatelského rozhraní vaší aplikace měly nějaký název, nicméně je důležité pojmenovat ty objekty, na které se budete odkazovat v kódu své aplikace. Jelikož hodnotu tohoto prvního šťastného čísla chcete řídit pomocí zpracovatele událostí, napsaného ve Visual Basicu, přiřadte tomuto objektu název **PrvniCislo**.

5. Objekt textového bloku *PrvniCislo* přešuněte pod písmeno „t“ v nadpise „Šťastná sedma“. V tuto chvíli by se vaše obrazovka měla podobat následující ukázce:



6. V Panelu nástrojů (Toolbox) znovu poklepejte na ovládací prvek *TextBlock* a vytvořte tak další objekt textového bloku.

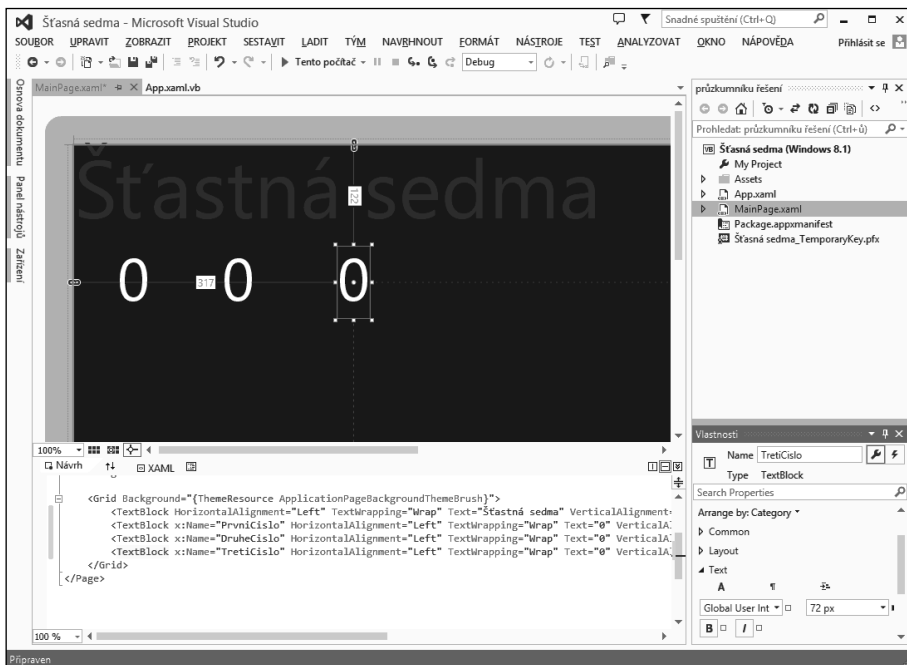
Tento objekt bude sloužit k zobrazení druhého šťastného čísla na stránce.

7. Pomocí okna Vlastnosti (Properties) přiřadte vlastnosti *Name* hodnotu **DruheCislo**, vlastnosti *Font Size* hodnotu **72** a do pole *Text* zadejte výchozí hodnotu textového bloku, tj. **0**.
8. Nový objekt *DruheCislo* přešuněte napravo od objektu *PrvniCislo*, přímo pod písmeno „t“ v nadpise „Šťastná sedma“.

Nyní vytvoříme třetí – a tedy poslední – objekt textového bloku pro zobrazování šťastných čísel na stránce.

9. V Panelu nástrojů (Toolbox) opět poklepejte na ovládací prvek *TextBlock* a vytvořte tak poslední objekt textového bloku.
10. Pomocí okna Vlastnosti (Properties) přiřaďte vlastnosti *Name* hodnotu **TretiCislo**, vlastnosti *Font Size* hodnotu **72** a do pole *Text* zadejte výchozí hodnotu textového bloku, tj. **0**.
11. Nový objekt *TretiCislo* přesuňte napravo od objektu *DruheCislo*, přímo pod mezeru v nadpise „Šťastná sedma“.

Po dokončení všech těchto kroků by stránka vaší aplikace se všemi čtyřmi objekty typu textový blok měla zhruba odpovídat následující ukázce. (Máte-li pocit, že objekty na vaší stránce nejsou rozmístěny zcela správně, můžete je samozřejmě přesunout.)



A nyní na stránku přidáme ovládací prvek typu tlačítko.

Přidání ovládacího prvku typu tlačítko

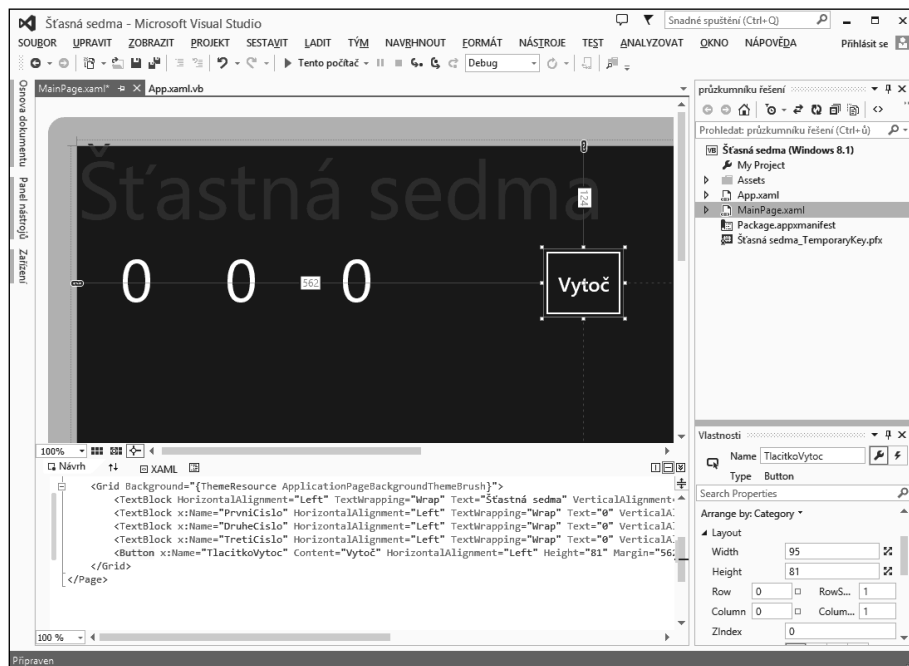
1. V Panelu nástrojů (Toolbox) klepněte na ovládací prvek *Button* a poté přesuňte ukazatel myši nad stránku vaší aplikace.
2. Klepněte myší do místa, kde chcete mít levý horní roh tlačítka. Tlačítko držte stisknuté a ukazatel táhnete směrem dolů a doprava. Tímto způsobem nakreslete celé tlačítko.
3. V okně Vlastnosti (Properties) přejděte do kategorie *Common* a vlastnosti *Content* přiřaďte hodnotu **Vytoč**. Poté stiskněte Enter.

Všimněte si, že obsah objektu typu tlačítko je definován vlastností *Content*, a nikoliv vlastností *Text*, jako tomu bylo u objektů typu textového bloku. To proto, že tlačítka mohou obsahovat i grafická a jiná data.

4. V horní části okna Vlastnosti (Properties) vyhledejte vlastnost *Name* a přiřaďte jí hodnotu **TlacitkoVytoč**. Poté stiskněte Enter.
5. V okně Vlastnosti (Properties) přejděte do kategorie *Text* a do pole *Font Size* zadejte **24**.

- Velikost objektu *TlačítkoVytoc* upravte tak, aby jeho výška činila 81 pixelů a šířka 95 pixelů. K tomuto účelu můžete využít vlastnosti *Height* a *Width*, nacházející se v kategorii *Layout* okna Vlastností (Properties).
- Objekt tlačítka přesuňte napravo od třetího šťastného čísla. Při přesouvání tohoto objektu se vám znovu zobrazí čáry pro automatické přichycení. Objekt přesouvejte tak dlouho, dokud se jeho horní okraj nepřichytí k čáře označující horní okraj objektů textových bloků, sloužících pro zobrazení tří šťastných čísel. Současně tlačítko umístěte pod poslední písmeno nadpisu „Šťastná sedma“.

Vaše obrazovka by nyní měla vypadat asi takto:



V dalších krocích přidáme na stránku obrázek, graficky ilustrující tu tučnou sumu, kterou vyhrájete, když vytočíte číslo 7 a trefíte tak jackpot. Pro zobrazování rastrových obrázků, ikon, digitálních fotografií a dalších typů grafických dat je určen ovládací prvek typu *Image*. Lze říci, že se jedná o jednu ze základních funkcí většiny aplikací určených pro Windows Store. A pravděpodobně nejčastěji se tento ovládací prvek používá pro zobrazování souborů ve formátu PNG či JPEG.

Vložení obrázku

- V Panelu nástrojů (Toolbox) klepněte na ovládací prvek typu *Image*.
- Pomocí ukazatele pro kreslení, který se zobrazí po přechodu na stránku, vytvořte velký obdélník, nacházející se pod objekty pro zobrazování šťastných čísel a pod tlačítkem *TlačítkoVytoc*.
- V případě potřeby si pomocí nástroje Zoom upravte zobrazení v okně Návrhář (Designer) tak, aby se vám zobrazila větší část stránky. Vhodné by například mohlo být nastavení na 50%.

Bylo by dobré, kdyby objekt obrázku pokrýval většinu zbývajících plochy stránky, nacházející se pod šťastnými čísly a pod tlačítkem *Tlačítko Vytoc*. Někdy je vhodné si pomocí nástroje Zoom zmenšit velikost stránky v okně Návrháře (Designer) a usnadnit si tak provádění podobných operací.

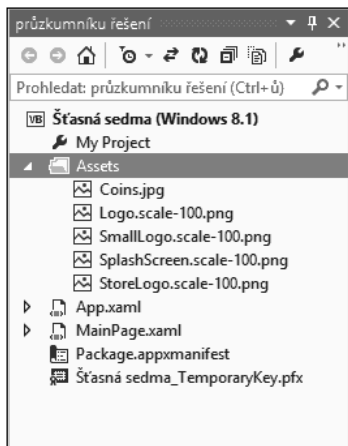
Nyní do celého projektu přidáme vhodnou fotografii, k čemuž využijeme Průzkumník řešení (Solution Explorer) a složku Assets, což je speciální kontejner pro ukládání zdrojových souborů, využívaných ve vašem projektu.

4. Není-li okno Průzkumníka řešení (Solution Explorer) dosud otevřené, přejděte do nabídky Zobrazit (View) a z ní vyberte Průzkumník řešení (Solution Explorer).

Jak jsme si již řekli, Průzkumník řešení (Solution Explorer) umožňuje přístup ke většině souborů tvořících daný projekt. A hned na začátku seznamu v tomto okně uvidíte složku Assets, což je kontejner pro logo projektu, jeho spouštěcí obrazovku a další soubory. V následujícím kroku přidáte do složky Assets digitální fotografii, čímž si ji zpřístupníte ve vytvářeném programu.

5. Pravým tlačítkem klepněte na složku Assets v Průzkumníku řešení (Solution Explorer) a zobrazte si tak nabídku několika užitečných příkazů sady Visual Studio.
6. Z této nabídky vyberte Přidat (Add) a z rozbalené podnabídky pak zvolte Existující položka (Existing Item).
7. V dialogu Přidat existující položku (Add Existing Item) přejděte do složky Dokumenty\Visual_Basic_KzK\Kapitola 03 a klepněte na soubor Coins.jpg. To je obrázek ve formátu JPEG, obsahující fotografii nejrůznějších mincí z celého světa – což je vhodná vizuální reprezentace výher, čekajících na výherce v aplikaci Šťastná sedma.
8. Klepnutím na tlačítko Přidat (Add) fotografii přidejte do složky Assets vytvářeného projektu.

Visual Studio zkopíruje vybraný soubor do této složky. Z následujícího obrázku je zřejmé, že vybraný obrázek se také ihned zpřístupní v Průzkumníku řešení (Solution Explorer), a to ve složce Assets:



Jakmile soubor přidáte do složky Assets, stane se součástí projektu, na němž právě pracujete. Díky tomu se pak na tento soubor můžete odkazovat v okně Vlastnosti (Properties). Daleko důležitější je však to, že se daný soubor stává součástí projektu i v okamžiku jeho kompilace a přípravy na distribuci – není tedy nutné pamatovat si, kde byl daný soubor

původně uložen na pevném disku, neboť jeho kopie bude automaticky distribuována spolu s projektem.

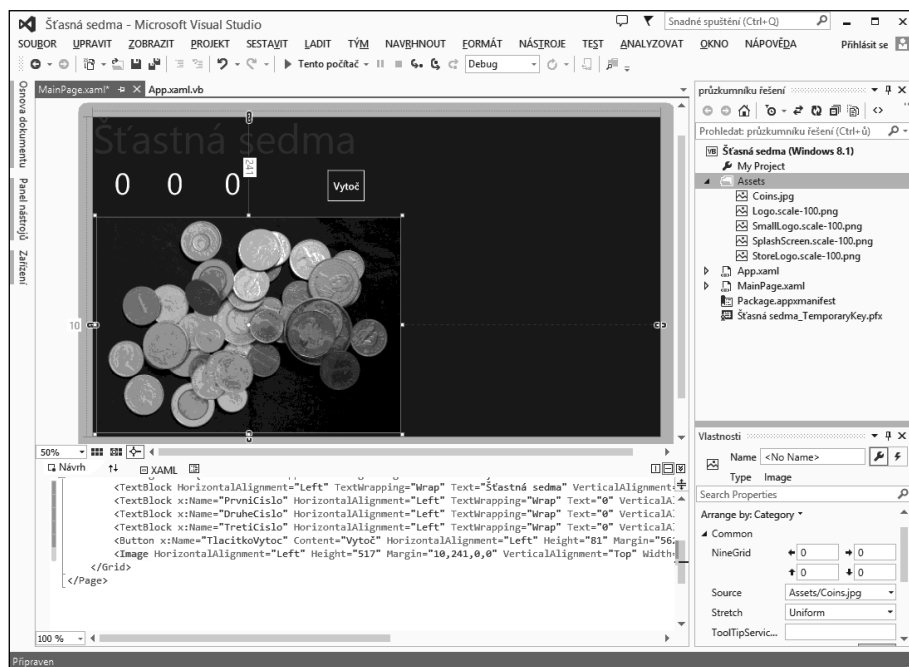
9. Vraťte se do okna Návrhář (Designer) a vyberte v něm objekt obrázku. V okně Vlastnosti (Properties) se vám musí zobrazit vlastnosti tohoto objektu.
10. V okně Vlastnosti (Properties) přejděte do kategorie *Common* a klepnutím si rozbalte seznam *Source*. Následně z tohoto seznamu vyberte soubor *Coins.jpg*.

Je možné, že si budete muset okno Vlastnosti (Properties) poněkud rozšířit, abyste si všimli malé šipky říkající, že pole *Source* je vlastně rozbalovacím seznamem. A jak sami uvidíte, v tomto seznamu se vám budou nabízet pouze soubory ze složky *Assets*.

Po výběru souboru fotografie nejrůznějších mincí z celého světa vyplní objekt obrázku v okně Návrhář (Designer).

11. Upravte velikost objektu obrázku a jeho odstupy od ostatních objektů tak, aby tento objekt vyplňoval většinu levé části stránky, zobrazené v okně Návrhář (Designer).

Po dokončení těchto úkonů by stránka vaší aplikace měla vypadat zhruba takto:



12. V horní části okna Vlastnosti (Properties) vyhledejte vlastnost *Name* a přiřaďte jí hodnotu **ObrazekMinci**.

Přiřazení názvu tomuto objektu je dosti důležité, neboť na tento objekt se budete odkazovat v kódu jazyka Visual Basic. V následujících cvičeních si možná všimnete toho, že poměrně často vytvářím název objektu tak, že spojím název typu objektu (v tomto případě tedy *Obrazek*) s nějakým popisným názvem (v tomto případě tedy *Minci*). Díky tomu je typ objektu zřejmý ze samotného názvu.

Nyní do programu přidáme zvukový efekt, který bude přehráván ve chvíli, kdy uživatel vytočí číslo 7. K přidání zmíněného zvukového efektu využijete ovládací prvek typu *MediaElement*, umožňující přehrávání audio a video souborů v aplikacích určených pro Windows Store. Zvuk, který bude aplikace přehrávat, je uložen v krátkém souboru typu WAV, nazvaném *ArcadeRiff*.

Přehrání zvukového souboru pomocí ovládacího prvku typu *MediaElement*

1. V Panelu nástrojů (Toolbox) rozbalte kategorii Všechny ovládací prvky jazyka XAML (All XAML Controls) a poklepejte na ovládací prvek *MediaElement*.

Visual Studio umístí do levého horního rohu stránky nový objekt typu přehrávač médií. Podobně jako všechny ostatní nové objekty i tento můžete v okně Návrháře (Designer) přesunout na novou pozici a upravit jej pomocí nastavení různých vlastností. Je však nutné zdůraznit, že ovládací prvek typu *MediaElement* je prvkem, který funguje v podstatě „v pozadí stránky“: s výjimkou případů přehrávání videí jej uživatel vůbec nevidí. V tuto chvíli můžete tento nově vytvořený objekt ponechat tam, kde se nachází.

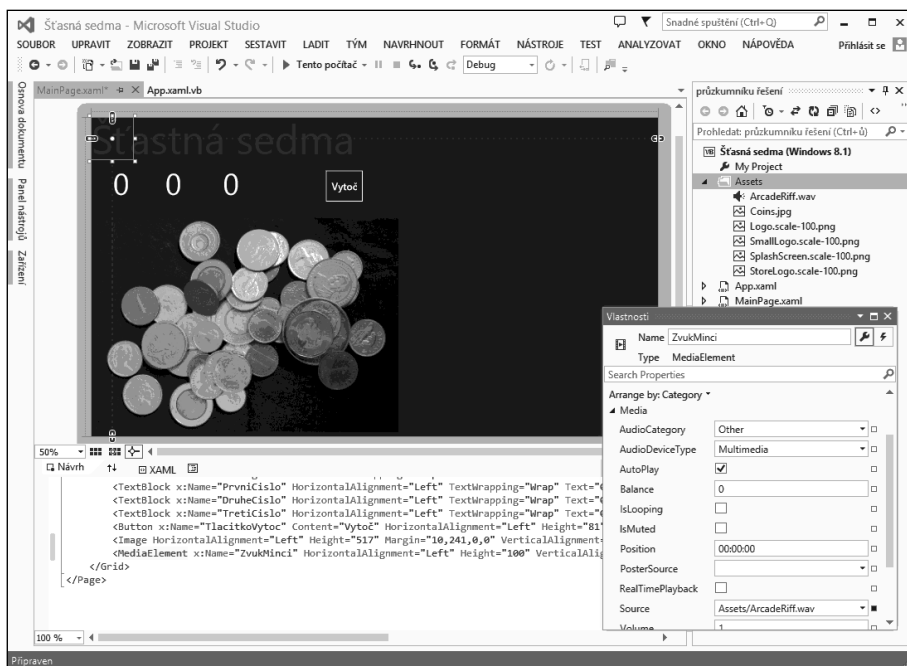
Vlastnost *Source* ovládacího prvku typu *MediaElement* určuje název multimediálního souboru, který bude načten do ovládacího prvku za účelem přehrání. Avšak ještě než můžete upravit nastavení této vlastnosti, musíte nejprve do složky *Assets* přidat platný multimediální soubor. V podstatě tedy musíte provést totéž, co jste provedli i v případě ovládacího prvku typu *Image*.

2. Pravým tlačítkem klepněte na složku *Assets* v Průzkumníku řešení (Solution Explorer) a zobrazte si tak místní nabídku.
3. V této nabídce klepněte na Přidat (Add) a následně vyberte Existující položka (Existing Item).
4. V dialogu Přidat existující položku (Add Existing Item) přejděte do složky *Dokumenty\Visual_Basic_KzK\Kapitola 03* a zde vyberte soubor *ArcadeRiff.wav*.
5. Klepnutím na tlačítko Přidat (Add) přidejte zvukový soubor ke svému projektu.

Visual Studio zkopíruje soubor do složky *Assets* projektu a následně jej zpřístupní v Průzkumníku řešení (Solution Explorer).

Nyní jste připraveni přiřadit vytvořenému objektu typu *MediaElement* nějaký název a pomocí vlastnosti *Source* jej nastavit tak, aby přehrával určený soubor.

6. V okně Návrháře (Designer) klepněte na objekt typu *MediaElement*. (V případě potřeby si stránku v okně Návrháře /Designer/ zvětšete – nezapomeňte, že objekt typu *MediaElement* je neviditelný, nicméně je možné jej vybrat. Tento objekt můžete vždy nalézt tak, že na záložce XAML editoru kódu /Code Editor/ klepnete na záznam *MediaElement*.)
7. V okně Vlastnosti (Properties) změňte vlastnost *Name* tak, aby obsahovala název **ZvukMinci**.
8. V okně Vlastnosti (Properties) rozbalte kategorii *Media* a v ní klepněte na seznam *Source*. V seznamu se objeví nový multimediální soubor (*ArcadeRiff.wav*).
Klepnutím na soubor *ArcadeRiff.wav* přiřadíte tento soubor objektu *ZvukMinci*.
Po všech těchto úpravách by vaše obrazovka měla vypadat zhruba takto (všimněte si záznamů v Průzkumníku řešení /Solution Explorer/ a v okně Vlastnosti /Properties/):



V okně Vlastnosti (Properties) vidíte také několik dalších důležitých vlastností, vztahujících se k objektům typu *MediaElement*, které si můžete podrobně prohlédnout a v případě potřeby i upravit.

Například standardně je zaškrtnuta volba *AutoPlay*, která ovládacímu prvku typu *MediaElement* říká, že ihned po načtení stránky má zadaný multimediální soubor automaticky přehrát. Jelikož naším cílem je přehrání souboru pouze v případě výhry uživatele, je nutné tuto volbu nyní zrušit.

9. Zrušte zaškrtnutí volby *AutoPlay*, nacházející se v kategorii *Media* dialogu Vlastnosti (Properties).

V celém dialogu ještě najdete několik dalších možností, které si můžete nyní prohlédnout (nikoliv však nastavovat). Například vlastnost *Position* říká, kde – ve kterém místě multimediálního souboru – má proces přehrávání začít. Tato vlastnost je užitečná především tehdy, pokud chcete zajistit přehrávání videa či zvuku od určitého místa.

Další zajímavou vlastností je *IsLooping*, což je booleovská hodnota umožňující přehrávání daného souboru neustále dokola. A nakonec se musíme zmínit i o vlastnosti *Volume*, sloužící k nastavení počáteční hlasitosti přehrávání multimediálního souboru. Tuto vlastnost můžete nastavit i pomocí kódu zpracovatele událostí, tj. v době běhu programu.

Konečná nastavení vlastností a další úpravy

Hlavní stránka vaší aplikace Šťastná sedma je téměř dokončena. V tuto chvíli již potřebujete nastavit několik posledních vlastností, napsat veškerý potřebný kód v jazyce Visual Basic a navrhnout spouštěcí obrazovku, která se uživateli zobrazí v době načítání programu.

Avšak ještě než se pustíme do plnění těchto úkolů, měli bychom se poněkud podrobněji zamyslet nad tím, jak vlastně bude program v době svého běhu fungovat. Hra začne ve chvíli, kdy

uživatel spustí program a klepne na tlačítko Vytoč. Jakmile uživatel klepne na toto tlačítko, aplikace musí vygenerovat tři náhodná čísla a zobrazit je v textových blocích, které jsou součástí stránky. Pokud uživatel trefí jackpot (tj. ve chvíli, kdy se alespoň v jednom ze tří textových bloků objeví číslo 7), musí se zobrazit objekt obsahující fotografii mincí, a současně musí být přehrán zvuk přiřazený objektu typu *MediaElement*.

Ačkoliv tok událostí se zdá být poměrně jednoduchý a přímočarý, program musí zůstat plně funkční i po první „výhře“. Což ovšem znamená, že pokud uživatel klepne na tlačítko Vytoč, musí ze stránky „zmizet“ obrázek mincí a musí zůstat skrytý až do chvíle, než uživatel vytočí další 7. Teprve poté se musí obrázek znovu objevit a musí dojít k opětovnému přehrání zvuku.

Chcete-li zajistit požadované chování aplikace, musíte najít mechanismus umožňující zobrazení či skrytí objektu obrázku dle potřeby. Tohoto cíle můžete dosáhnout nastavením vlastnosti *Visibility* objektu obrázku, která může nabývat buď hodnoty *Visible* (tj. obrázek bude zobrazený), anebo hodnoty *Collapsed* (tj. obrázek bude skrytý) dle potřeby. Ve skutečnosti je tomu dokonce tak, že většinu objektů v aplikacích navržených pro Windows Store lze zobrazit či naopak skrýt právě pomocí zmíněné vlastnosti – lze tedy říci, že se jedná o vestavěný nástroj umožňující řízení toho, co bude či nebude na stránce vidět. Zkusme jej nyní použít.

Nastavení vlastnosti *Visibility*

1. Přejděte do okna Návrháře (Designer) a klepněte na objekt obrázku.
2. V okně Vlastnosti (Properties) rozbalte kategorii *Appearance* a poté klepněte na vlastnost *Visibility*.
3. Z rozbaleného rozbalovacího seznamu vyberte hodnotu *Collapsed*.

Objekt obrázku ze stránky zmizí. Nemusíte se však ničeho obávat – toto je přesně stav, kterého jsme chtěli docílit. Objekt obrázku jsme neodstranili, pouze jsme v tuto chvíli zabránili jeho zobrazení. Jeho zviditelnění zajistíte pomocí kódu, který bude součástí zpracovatele událostí.

Nyní je nutné změnit barvu pozadí stránky. Výchozí barvou pozadí všech stránek aplikací navržených pro Windows Store je černá, avšak poněkud barevnější provedení může hru učinit přitažlivější. Barvu pozadí můžete změnit výběrem objektu typu *Grid*, který je standardní součástí stránky, a následnou úpravou hodnot v kategorii *Brush*, kterou najdete v okně Vlastnosti (Properties).

Nastavení barvy pozadí stránky

1. Klepnutím na pozadí stránky (tj. nikoliv na kterýkoliv z těch objektů, které jste před chvílí přidali) v okně Návrháře (Designer) vyberte objekt typu *Grid*.

To, že se vám podařilo vybrat objekt typu *Grid*, poznáte vcelku snadno podle toho, že jeho vlastnosti se zobrazí v okně Vlastnosti (Properties).

Jak se dozvíte v kapitole 7 „Značkovací jazyk XAML krok za krokem“, každý z objektů aplikace navržené pro Windows Store je definován nějakým kódem značkovacího jazyka XAML a daty, která lze zadat či upravit v editoru kódu (Code Editor). Objekt typu *Grid* je základním prvkem rozvržení stránky a všechny prvky stránky jsou vnořeny do tohoto objektu. Kromě toho, že objekt typu *Grid* slouží jako užitečný kontejner dalších objektů, má tento objekt také několik nastavení, která můžete upravovat. A jedním z nich je i barva pozadí, se kterou bude vaše aplikace pracovat. Tuto barvu nyní upravíme.

2. Přejděte do okna Vlastnosti (Properties) a rozbalte kategorii *Brush*. V ní pak klepněte na tlačítko Solid color brush.

3. Poblíž spodního okraje editoru Color resources vyberte číslo, na jehož začátku vidíte znak mřížky (#). Toto číslo nahraďte slovem **Green** a poté stiskněte Enter.

V textovém poli se objeví alfanumerická hodnota zelené barvy (#FF008000) a barva pozadí objektu typu *Grid* se změní na zelenou. Pokud chcete, můžete si vyzkoušet i použití jiných barev.

A to je vše, co se týká návrhu uživatelského rozhraní. Než přejdete k vlastnímu psaní kódu, měli byste svoji práci uložit.

Uložení změn

1. V nabídce Soubor (File) klepněte na příkaz Uložit vše (Save All). Uložíte tak všechny změny, které jste dosud v projektu Šťastná sedma provedli.

Příkaz Uložit vše (Save all) zajistí uložení opravdu všech součástí projektu – souboru projektu, všech stránek, souborů kódu na pozadí, všech zdrojů, manifestu balíčku a všech dalších součástí, souvisejících s vytvářenou aplikací. A jelikož svůj nový projekt ukládáte poprvé, zobrazí se dialog Uložit projekt (Save Project), požadující od vás zadání názvu a umístění projektu. (Máte-li ovšem sadu Visual Studio nastavenou tak, aby od vás vyžadovala umístění projektu v okamžiku jeho vytváření, pak v tuto chvíli již dialog Uložit projekt /Save Project/ nevidíte – Visual Studio pouze uloží všechny vámi provedené změny.)

2. Pomocí tlačítka Procházet (Browse) vyhledejte umístění pro svoje soubory. Doporučuji vám, abyste použili složku Dokumenty\Visual_Basic_KzK\Kapitola 03 (což je umístění ukázkových souborů, doprovázejících tuto knihu), nicméně skutečné umístění záleží jen na vás. A protože jste při vytváření projektu použili prefix „Moje“, vaše verze aplikace nepřepíše ty soubory, které jste si stáhli z webu.
3. Zrušte zaškrtnutí volby Vytvořit adresář pro řešení (Create Directory For Solution).
Je-li tato volba zaškrtnutá, Visual Studio vytvoří druhou složku pro soubory související s vámi vytvářeným programem. V případě řešení obsahujících pouze jediný projekt (což platí pro většinu programů v této knize) toto rozhodně není nutné.
4. Klepnutím na tlačítko Uložit (Save) celý projekt uložte.



Tip: Chcete-li uložit pouze ten prvek projektu, na němž právě pracujete (například hlavní stránku projektu, nějaký modul s kódem či nějakou jinou část), můžete k tomuto účelu použít volbu Uložit (Save) z nabídky Soubor (File). Pokud potřebujete uložit daný prvek pod jiným názvem, pak použijte volbu Uložit jako (Save As), kterou také najdete v nabídce Soubor (File). Obě zmíněné volby jsou v nabídce Soubor (File) vždy doprovázeny názvem dané komponenty.

Psaní kódu

Nyní jste připraveni začít psát kód programu Šťastná sedma. Jelikož většina objektů, které jste dosud vytvořili, již „ví“, jakým způsobem mají v době běhu programu fungovat, jsou tyto objekty připraveny přijímat vstupy od uživatele a zpracovávat je. „Vrozená“ funkcionality objektů je jednou z velkých výhod sady Visual Studio a jazyka Visual Basic – po umístění objektů na stránku a nastavení jejich vlastností je možné tyto objekty použít bez jakéhokoliv dalšího programování.

Nicméně našemu programu, nazvanému Šťastná sedma, stále chybí „základ“ – ten kód, který zajistí samotný výpočet náhodných čísel, jejich zobrazení v textových blocích a případné roz-

poznání výhry. Avšak tato výpočetní logika může být do naší aplikace určené pro Windows Store přidána pouze pomocí programových příkazů – tedy kódu, který jasně říká, co má program v každém okamžiku dělat. A jelikož program je spuštěn stiskem tlačítka Vytoč, propojíte kód hry se zpracovatelem událostí tohoto tlačítka.

V následujících krocích zadáte pomocí editoru kódu (Code Editor) v jazyce Visual Basic kód hry Šťastná sedma.

Použití editoru kódu

1. V okně Návrhář (Designer) sady Visual Studio klepněte na objekt *TlacitkoVytoc*.
2. Přejděte do okna Vlastnosti (Properties), případně si toto okno otevřete, a sbalte všechny kategorie, které se vám zobrazí jako rozbalené.
3. Poblíž horního okraje okna Vlastnosti (Properties), napravo od vlastnosti *Name* a tlačítka *Properties*, klepněte na tlačítko *Event handlers* (malé čtveraté tlačítko, mající ikonu připomínající blesk).

Okno Vlastnosti (Properties) se vyplní kolekcí různých akcí či událostí, na něž je objekt typu tlačítko schopen reagovat. K typickým událostem, které je tlačítko schopné rozpoznat, patří například *Click* (tj. klepnutí na tlačítko), *DragOver* (tj. přetažení nějakého objektu přes tlačítko), *Tapped* (tj. dotknutí se tlačítka prstem) či *Drop* (tj. přetažení nějakého objektu přes tlačítko a uvolnění tohoto objektu nad tlačítkem).

Jelikož Visual Basic je v podstatě programovacím jazykem řízeným událostmi, ve většině případů budete muset jako softwaroví vývojáři vždy vytvořit nějaká uživatelská rozhraní, reagující na různé typy vstupů ze strany uživatelů, a poté budete muset napsat kód různých zpracovatelů událostí, schopných takové vstupy zpracovat. Přitom lze říci, že obvykle budete muset vytvořit kód zpracovatelů událostí pouze pro několik událostí, souvisejících s objekty, které ve svých programech použijete. (Nicméně nabízený seznam událostí je poměrně komplexní, a to proto, abyste při výběru vhodné události měli co nejvíce možností.)

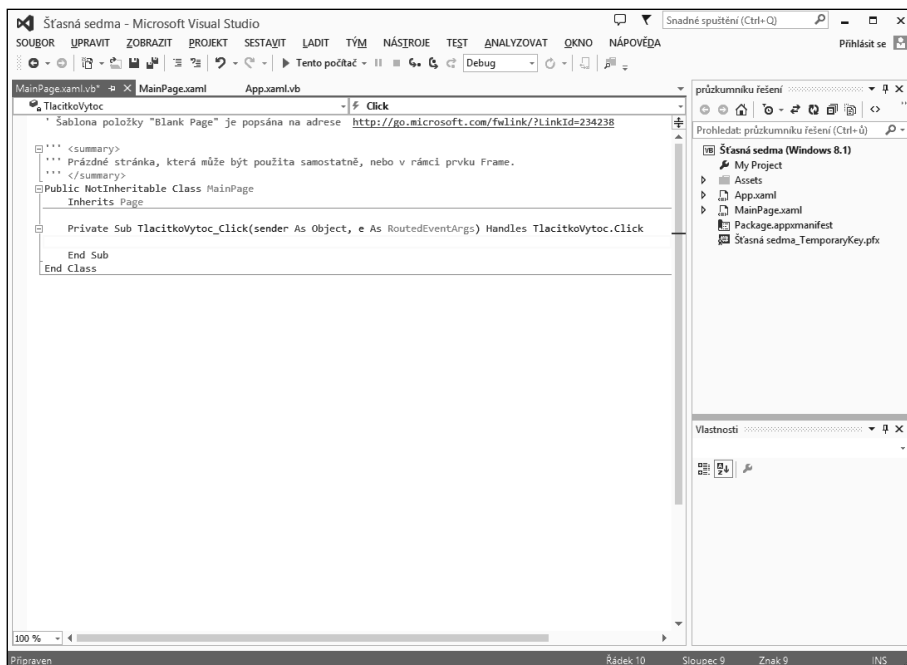
Chcete-li vytvořit zpracovatele události pro danou událost, pak v okně Vlastnosti (Properties) poklepejte na okénko umístěné vedle názvu vybrané události. A jelikož v našem případě chcete zajistit vygenerování tří náhodných čísel vždy, když uživatel klepne na tlačítko Vytoč, budete psát kód zpracovatele události *Click* tohoto tlačítka.

4. V okně Vlastnosti (Properties) poklepejte na textové okénko nacházející se vedle události *Click*.

Sada Visual Studio vloží do tohoto pole zpracovatele události s názvem *TlacitkoVytoc_Click*. Současně v okně editoru kódu (Code Editor) otevře soubor kódu na pozadí, nazvaný *MainPage.xaml.vb*. Po provedení tohoto kroku by vaše obrazovka měla vypadat zhruba tak jako na obrázku na následující straně.

Uvnitř editoru kódu (Code Editor) najdete programové příkazy související se šablonou *Prázdná aplikace (XAML)*, kterou jste si zvolili při vytváření tohoto projektu. Jedná se o programový kód v jazyce Visual Basic a vy si možná ihned všimnete toho, že část kódu je uspořádána do jednotek, jimž se říká *procedury*. A v dolní části souboru se nachází procedura toho zpracovatele událostí, kterého jste právě vytvořili. Jak vidíte, tato procedura je nazvaná *TlacitkoVytoc_Click*.

Celá procedura je vymezena klíčovými slovy *Sub* a *End Sub*, zatímco klíčová slova *Protected* a *Private* říkají, jakým způsobem bude procedura použita. Blíže se s těmito klíčovými slovy seznámíte později.



Níže vidíte ukázkou toho kódu, který sada Visual Basic automaticky vytvořila ve chvíli, kdy jste poklepali na textové pole u vlastnosti *Click*. Jedná se vlastně o první a poslední řádek celé procedury. (Kód, který uvidíte na obrazovce, nebude zalomen tak jako v naší ukázce. Toto zalomení bylo dáno nutností respektování okrajů stránky pro účely tisku.)

```
Private Sub TlacitkoVytoc_Click(sender As Object, e As RoutedEventArgs)
    Handles TlacitkoVytoc.Click
```

```
End Sub
```

Samotný kód procedury pak musíte vložit mezi tyto řádky. Tento kód bude proveden vždy, když uživatel aktivuje prvek či událost rozhraní související s danou procedurou. V našem případě je touto událostí klepnutí na tlačítko Vytoč, ale jak uvidíte v dalších kapitolách této knihy, může se jednat i o zcela jiný druh události. Programátoři nazývají tuto posloupnost kroků „spuštěním události“.



Tip: V okně editoru kódu (Code Editor) si můžete také všimnout několika řádek kódu, v nichž má písmo zelenou barvu. Ve výchozím nastavení sady Visual Studio platí, že zelenou barvou jsou zvýrazňovány *komentáře* neboli vysvětlující poznámky, které do kódu zadává přímo jeho tvůrce proto, aby zajistil vyšší srozumitelnost kódu či vysvětlil ostatním některé jím provedené kroky. Platí, že kompilátor jazyka Visual Basic komentáře neprovádí neboli *nevyhodnocuje*.

5. Mezi uvedené řádky запиšte následující kód. Po zadání posledního řádku stiskněte **Enter**:

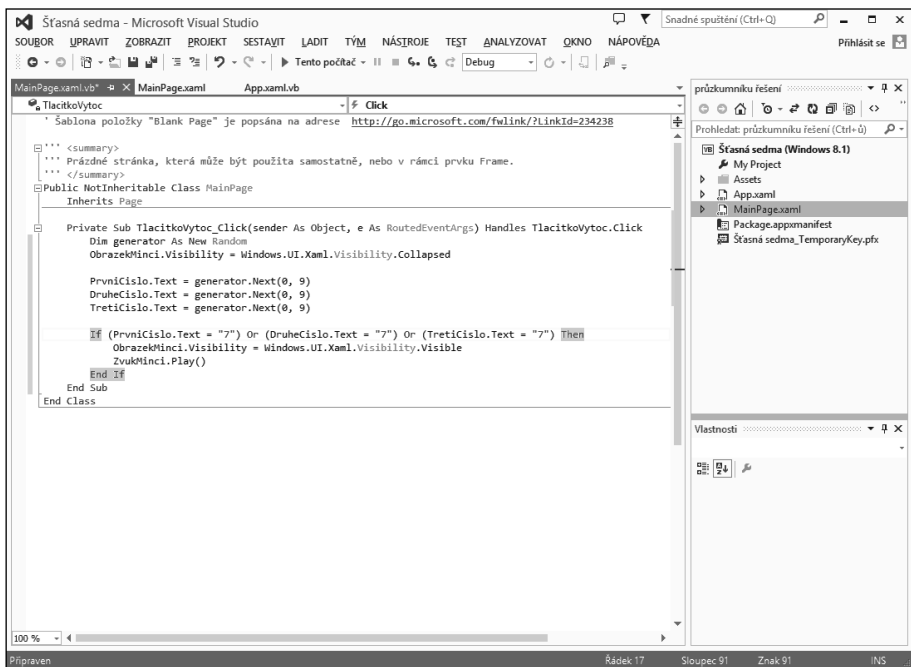
```
Dim generator As New Random
ObrazekMinci.Visibility = Windows.UI.Xaml.Visibility.Collapsed
```

```
PrvniCislo.Text = generator.Next(0, 9)
DruheCislo.Text = generator.Next(0, 9)
TretiCislo.Text = generator.Next(0, 9)
```

```
If (PrvniCislo.Text = "7") Or (DruheCislo.Text = "7") Or
    (TretiCislo.Text = "7") Then
    ObrazekMinci.Visibility = Windows.UI.Xaml.Visibility.Visible
    ZvukMinci.Play()
End If
```

Při psaní kódu programu Visual Studio provádí formátování kódu a zobrazuje různé části kódu různými barvami, aby vám usnadnilo rozlišování jednotlivých prvků. Když začnete psát název vlastnosti nějakého objektu, Visual Studio zobrazí seznam se všemi dostupnými vlastnostmi daného objektu, díky čemuž můžete vlastnost buď pouze vybrat ze seznamu, anebo můžete pokračovat dál v psaní jejího názvu.

V tuto chvíli by vaše obrazovka měla vypadat zhruba takto:



Poznámka: Zobrazí-li vám Visual Studio v průběhu psaní kódu nějaké chybové hlášení, pak jste pravděpodobně špatně zapsali některý z příkazů. V takovém případě pečlivě porovnejte daný řádek kódu s textem v této knize, proveďte nezbytné úpravy a pokračujte v psaní dalších částí kódu. (Samozřejmě můžete daný řádek i vymazat a zapsat jej celý znovu.)

V sadě Visual Studio mohou být příkazy programu tvořeny klíčovými slovy, názvy vlastností, názvy objektů, proměnnými, čísly, speciálními znaky a dalšími hodnotami. Při psaní těchto prvků v okně editoru kódu (Code Editor) bude sada Visual Studio využívat funkci zvanou IntelliSense k tomu, aby vám s psaním kódu pomohla. Díky této funkci je

sada Visual Studio schopna automaticky dokončit mnoho výrazů, jakmile je jednoznačně rozpozná.

6. Klepnutím na tlačítko Uložit vše (Save All) uložte všechny provedené změny.

Rozbor kódu zpracovatele události `TlacitkoVytoc_Click`

Kód zpracovatele události `TlacitkoVytoc_Click` je prováděn vždy, když uživatel klepne na tlačítko Vytoč, které je součástí hlavní stránky aplikace. Lze říci, že v podstatě tento zpracovatel události provádí čtyři základní úlohy:

1. Deklaruje generátor náhodných čísel, nazvaný *generator*.
2. Skrývá fotografii mincí.
3. Generuje tři náhodná čísla a zobrazuje je v objektech typu textový blok.
4. Je-li jedno z vygenerovaných čísel rovno 7, zobrazuje fotografii mincí a přehrává zvuk.

Podívejme se na jednotlivé uvedené kroky poněkud podrobněji.

Generátor náhodných čísel je deklarován tímto řádkem kódu:

```
Dim generator As New Random
```

Je pravděpodobné, že jste již někdy dříve v nějakých programech deklarovali proměnné a následně je i využívali. V tomto případě si však všimněte typu proměnné – generátor je deklarován pomocí typu *Random*, který byl speciálně navržen proto, aby podporoval vytváření tzv. „pseudonáhodných“ čísel – tj. čísel neodpovídajících žádné číselné řadě či žádnému vzoru a spadajících do zadaného rozsahu. Náhodná čísla budete v následujících kapitolách této knihy používat velmi často. Více informací o datových typech a jejich vzájemných převodech se dozvíte v kapitole 11 „Datové typy, operátory a práce s řetězci“.

Skrýtí fotografie mincí je zajištěno následujícím řádkem kódu:

```
ObrazekMinci.Visibility = Windows.UI.Xaml.Visibility.Collapsed
```

Jak jsme si již řekli, vlastnost *Visibility* říká, zda daný objekt bude či nebude na stránce viditelný. Tato konkrétní syntaxe využívá objekty platformy .NET Framework ke skrytí fotografie mincí. (Tento řádek byl do kódu přidán proto, aby zajistil výchozí stav v případě, že v předcházejícím kole hry byla vytočena výhra a fotografie tedy byla zobrazena.)

Následující tři řádky kódu zajišťují výpočty náhodných čísel. Zdá se vám snad toto tvrzení poněkud podivné? Ve skutečnosti můžete Visual Basic „přesvědčit“ k tomu, aby generoval nepředvídatelná čísla v určitém rozsahu – tj. můžete vytvářet náhodná čísla pro loterijní hry, vrhání kostek či jiné podobné hry, založené na statistických vzorech. Metoda *Next* instance *generator* v každém z následujících řádků kódu vytváří náhodné číslo v rozmezí 0 až 9 – což je přesně to, co potřebujete pro tuto hru.

```
PrvniCislo.Text = generator.Next(0, 9)
```

```
DruheCislo.Text = generator.Next(0, 9)
```

```
TretiCislo.Text = generator.Next(0, 9)
```

Poslední skupina příkazů programu pak provádí kontrolu, zda některé z náhodných čísel není rovno 7. Je-li jedno či více náhodných čísel rovno 7, program zobrazí grafické znázornění výhry a přehraje zvuk, jímž oznamuje výhru.

```

If (PrvniCislo.Text = "7") Or (DruheCislo.Text = "7") Or
  (TretiCislo.Text = "7") Then
  ObrazekMinci.Visibility = Windows.UI.Xaml.Visibility.Visible
  ZvukMinci.Play()
End If

```

Kdykoliv uživatel klepne na tlačítko Vytoč, je proveden (či též zavolán) kód zpracovatele události *TlacitkoVytoč_Click*. To znamená, že jsou znovu provedeny všechny příkazy tohoto kódu. Pokud však klepnete na tlačítko Vytoč mnohokrát, a to velmi rychle za sebou, je možné, že neuslyšíte jeden či více zvukových efektů, neboť objekt typu *MediaElement* je schopen přehrávat pouze jeden zvuk najednou.

Spouštění aplikací navržených pro Windows Store

Gratulujeme! V tuto chvíli máte dokončenou svoji první aplikaci pro Windows Store a jste připraveni ji spustit. Chcete-li spustit program napsaný ve Visual Basicu přímo z IDE, pak můžete použít kterýkoliv z následujících postupů:

- V nabídce Ladit (Debug) můžete vybrat Spustit ladění (Start Debugging).
- Na panelu nástrojů Standardní (Standard) můžete klepnout na tlačítko Spustit ladění (Start Debugging). (Vedle tohoto tlačítka obvykle uvidíte nápis „Tento počítač“ / „Local Machine“ /, neboť standardně se ladění spouští právě na lokálním počítači.)
- Anebo můžete stisknout klávesu F5.

Zkuste si nyní spustit program Šťastná sedma. Zobrazí-li Visual Studio nějaké chybové hlášení, pak jste pravděpodobně při psaní zdrojového kódu udělali jednu či dvě chyby. Pokuste se tyto chyby opravit na základě porovnání vámi napsaného kódu a toho kódu, který je otištěný v této knize.



Poznámka: Předpokládám, že jste svůj projekt nazvali Moje šťastná sedma, nicméně pokyny a ukázky obrazovek, uvedené níže, zobrazují projekt Šťastná sedma. Je totiž možné, že namísto psaní vlastního kódu jste se rozhodli spustit ukázkový kód, který jsem pro vás připravil.

Spuštění programu Šťastná sedma

1. Na panelu nástrojů Standardní (Standard) klepněte na tlačítko Spustit ladění (Start Debugging).
Program Šťastná sedma se zkompileje a spustí. Po několika vteřinách se zobrazí uživatelské rozhraní v té podobě, v jaké jste jej navrhli.
2. Klepněte na tlačítko Vytoč.
Program vygeneruje tři náhodná čísla a zobrazí je v textových blocích připravených na stránce. Podaří-li se vám vytočit číslo 7, bude vaše obrazovka vypadat takto:



Vytočení čísla 7 způsobí také přehrání zvukového efektu, trvajícího pouze několik vteřin. Samotný efekt se snaží napodobit výherní automat. Vyhrali jste!

3. Klepněte na tlačítko Vytoč ještě 15x či 16x a sledujte změny čísel v textových blocích.

Sami uvidíte, že v přibližně polovině případů trefíte jackpot – což je docela slušná šance. (Skutečná šance je zhruba 2,8x z 10; zpočátku máte prostě jenom větší štěstí.) Později si můžete zkusit zvýšit obtížnost hry tím, že zobrazíte fotografii a přehrajete zvukový efekt pouze v případě, že se číslo 7 objeví ve 2 či všech 3 textových blocích současně. Můžete se také pokusit o zobrazení průběžného počtu výher.

4. Až dokončíte veškeré experimenty se svojí novou aplikací pro Windows Store, můžete ji zavřít.

Program se zastaví a na obrazovce se znovu objeví IDE. Klepnutím na tlačítko Ukončit ladění (Stop Debugging) na panelu nástrojů ukončete běh programu. Nyní si k programu přidáte spouštěcí obrazovku.

Vytvoření spouštěcí obrazovky aplikace

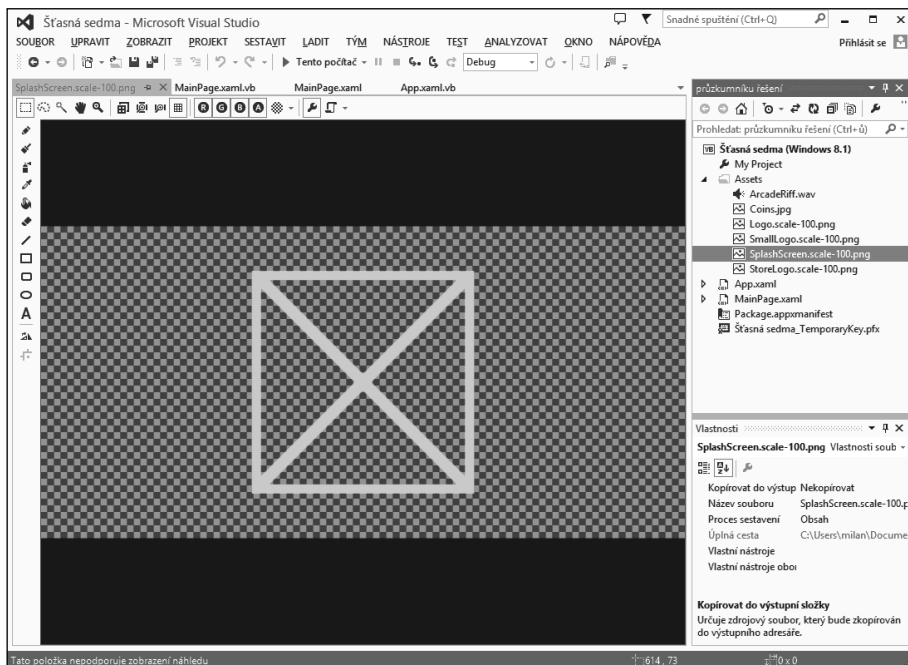
Spouštěcí obrazovkou se míní obrázek, který se na obrazovce zobrazuje pouze po dobu spouštění aplikace. Platí, že každá aplikace navržená pro Windows Store musí mít spouštěcí obrazovku tvořenou nějakým obrázkem (či textem) a barvou pozadí. Spouštěcí obrazovka je v Průzkumníku řešení (Solution Explorer) uložena ve složce Assets. Sami jste si již možná stačili všimnout toho, že jakmile vytvoříte novou aplikaci pro Windows Store, stane se její součástí jednoduchá základní spouštěcí obrazovka, kterou automaticky vytvoří Visual Studio. Ve složce Assets také uvidíte obrázky pro dlaždice, jejichž úpravami se budeme zabývat v kapitole 9 „Seznámení se základními prvky návrhu pro Windows 8.1: řádek nabídek, informační rámeček, dlaždice a dotyk“.

Ačkoliv si spouštěcí obrazovku můžete samozřejmě vytvořit pomocí programu Malování či jakéhokoliv pokročilejšího grafického editoru od nějakého třetího výrobce, jednoduchou spouštěcí obrazovku lze navrhnout i přímo v sadě Visual Studio. Mějte však na paměti, že spouštěcí obrazovka je zobrazena opravdu jen krátce, v době spuštění vašeho programu. Z toho ovšem vyplývá, že spouštěcí obrazovka rozhodně není tím správným místem pro zobrazování nějakých složitých pokynů pro používání programu či informací o ochraně autorských práv. Stejně tak byste se měli vyhnout i zobrazování reklam a informací o verzi na spouštěcí obrazovce.

Namísto toho zkuste využít spouštěcí obrazovku k tomu, abyste uživateli nabídli jakýsi náhled funkcionality vaší aplikace nějakým jedinečným způsobem. Zvažte například takový obrázek či takovou fotografii, kterou bude možné snadno přizpůsobit pro uživatele z jiných zemí a kulturních prostředí (tj. bude možné ji snadno *lokalizovat*) a kterou bude možné efektivně zobrazit na displejích s různým rozlišením. Pro tyto účely se převážně používá formát Portable Network Graphics (soubory s příponou .png), neboť tento formát umožňuje zobrazování průsvitnosti alfa kanálu a obrázků s 24bitovou barevnou hloubkou. Je-li část obrázku označena či naformátována jako průsvitná, pak za obrázkem bude zobrazena barva pozadí. (Tento jev je využíván u většiny spouštěcích obrazovek a dlaždic aplikací navržených pro Windows Store a Windows Phone.)

Vytvoření spouštěcí obrazovky hry Šťastná sedma

1. V okně Průzkumník řešení (Solution Explorer) otevřete složku Assets a poté poklepejte na soubor SplashScreen.scale-100.png.
2. V důsledku této akce se v sadě Visual Studio otevře Editor obrázků (Image Editor), do něhož se načte soubor SplashScreen.scale-100.png. Vaše obrazovka by nyní měla vypadat zhruba takto:



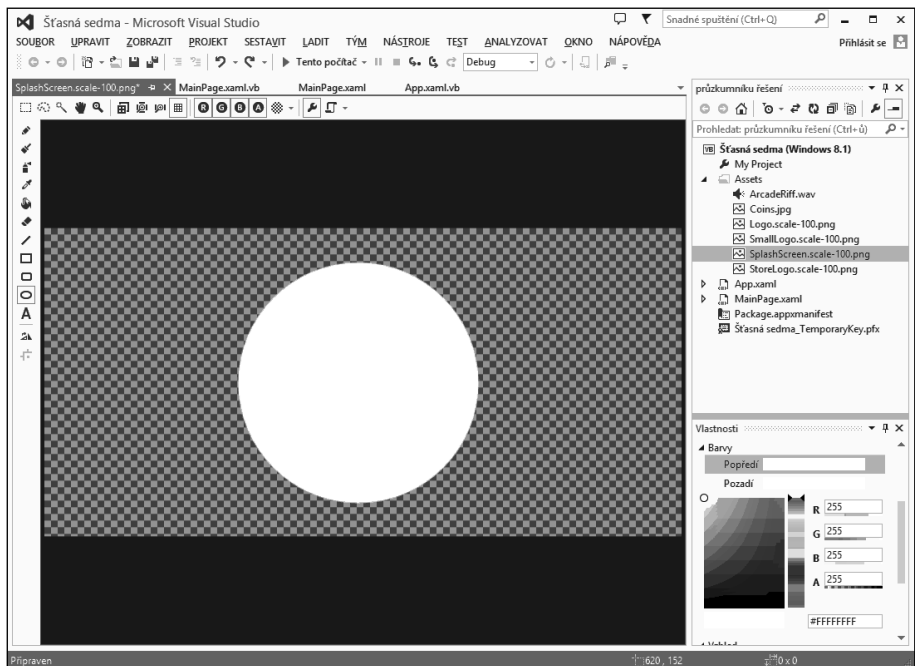
Okna Průzkumník řešení (Solution Explorer) a Vlastnosti (Properties) jsou stále viditelná. Aktivním oknem je však Editor obrázků (Image Editor), přičemž plocha pro návrh je obklopena nástroji pro práci s grafikou. Ten tvar „X“, který vidíte uprostřed, je právě oním zmíněným výchozím obrázkem pro soubor `SplashScreen.scale-100.png`. Jinými slovy řečeno, to je ten obrázek, který nyní chcete nahradit.

3. V levém horním rohu Editoru obrázků (Image Editor) klepněte na nástroj pro výběr. Poté vyberte celý tvar „X“ a stiskněte klávesu Delete.

Nyní máte před sebou prázdné „plátno“, na němž můžete vytvořit vlastní obrázek spouštěcí obrazovky. Ten šachovnicový vzor, který vidíte, je speciální barevné schéma, znázorňující ty části obrázku, které jsou průsvitné – jinými slovy řečeno, to, co je nyní zobrazováno jako zelená šachovnice, bude v okamžiku zobrazení spouštěcí obrazovky po spuštění programu nahrazeno pozadím.

4. Na levé straně Editoru obrázků (Image Editor) klepněte na nástroj Nakreslit elipsu (Draw an ellipse) a poté uprostřed spouštěcí obrazovky nakreslete kružnici.

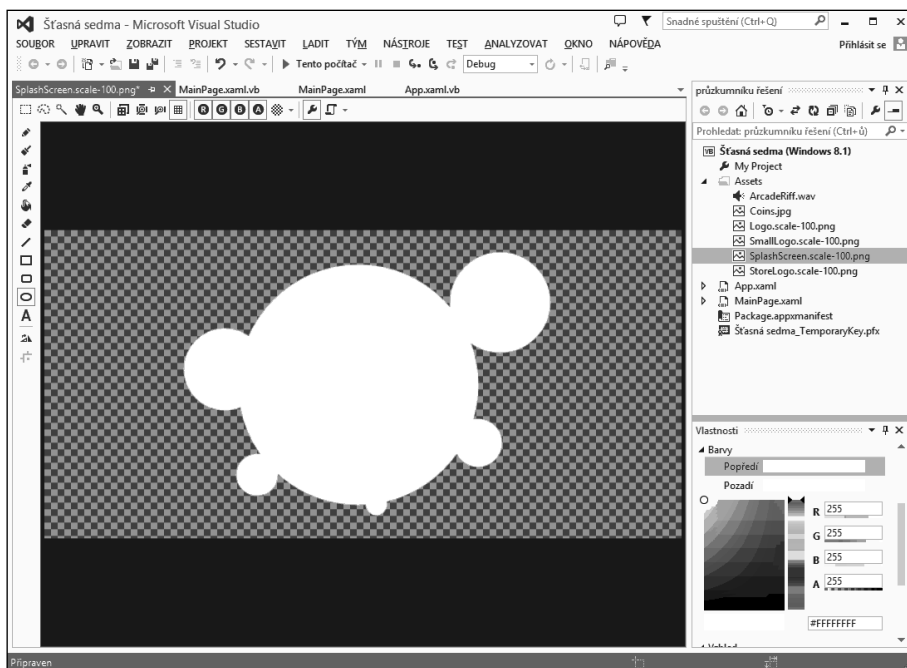
Pokud chcete, můžete při kresbě kružnice využít informace o souřadnici X a souřadnici Y, zobrazené v pravé části stavového řádku sady Visual Studio. Úplně vpravo pak ve stavovém řádku naleznete informace o rozměrech kresleného objektu ve směru osy X a Y. Případně se můžete rozhodnout přesunout kružnici do středu obrazovky pomocí nástroje pro výběr. Vaše spouštěcí obrazovka by v Editoru obrázků (Image Editor) měla nyní vypadat asi takto:



5. Pomocí nástroje Nakreslit elipsu (Draw an ellipse) přidejte do spouštěcí obrazovky další čtyři nebo pět menších kružnic a umístěte je podél obvodu právě vytvořené kružnice.

Typické spouštěcí obrazovky zobrazují jednoduché geometrické tvary, podobající se našemu příkladu. Zvažte například použití zjednodušené verze loga vaší firmy.

Po těchto úpravách by vaše jednoduchá spouštěcí obrazovka měla vypadat takto:



Tuto spouštěcí obrazovku můžete obohatit ještě o další efekty: můžete ji například „vyzdobit“ dalšími barvami, obrázky, textem či animacemi. Avšak pro účely naší první aplikace je tato obrazovka zcela dostačující a v dalších krocích máme s čím pracovat.

6. V nabídce Soubor (File) zvolte Uložit vše (Save All) a uložte tak všechny právě provedené změny.
7. Stisknutím klávesy F5 spustíte celý projekt a prohlédnete si spouštěcí obrazovku. Všimněte si, že spouštěcí obrazovka se zobrazí jenom na chvilku. Stihli jste vůbec postřehnout jednotlivé nakreslené kruhy a černou barvu pozadí?
8. Program uzavřete, ukončete ladění a zavřete okno Editoru obrázků (Image Editor).

Váš projekt je tímto dokončen. Nastal tedy čas jej přidat na obrazovku Start vašeho počítače a vyzkoušet jej tak v „reálném“ provozu. V této chvíli je však nutné zdůraznit, že pokud by se jednalo o komerční aplikaci určenou pro Windows Store a připravovanou pro distribuci ostatním uživatelům právě prostřednictvím Windows Store museli byste nyní ke své aplikaci přidat další položky, popsané v tabulce 1-1. Více informací najdete v kapitole 1 „Možnosti vývoje v jazyce Visual Basic a Windows Store“.

Ukázkové projekty na disku

Pokud jste se rozhodli nevytvářet projekt *Moje šťastná sedma* od počátku (anebo pokud jste projekt vytvářeli a nyní chcete porovnat výsledek svého snažení s tím, co jsem pro vás připravil během psaní této kapitoly), zkuste si otevřít a spustit dokončený projekt *Šťastná sedma*, který najdete na svém pevném disku ve složce `Visual_Basic_KzK\Kapitola 03` (to je výchozí umístění všech ukázkových souborů souvisejících s touto kapitolou). Potřebujete-li si zopakovat postup otevírání projektů, vraťte se ke kapitole 2: ta obsahuje podrobný popis těch kroků, které musíte provést.

Tato kniha je zamýšlena jako učebnice, která vás bude programováním ve Visual Basicu provádět krok za krokem. To ovšem znamená, že její studium bude pro vás největším přínosem právě tehdy, pokud se rozhodnete vytvářet jednotlivé projekty během čtení příslušných cvičení a pokud si vyhradíte dostatek času i pro nějaké další experimentování s vytvářenými projekty. Nicméně po dokončení takových projektů je určitě dobré porovnat svoje výsledky s tím, co jsem pro vás připravil sám. Toto platí zejména v případech, kdy jsou výsledky vašeho snažení poněkud neočekávané. Abych vám toto porovnání usnadnil, ve většině cvičení uvedu název souvisejících hotových projektů na disku.

Po porovnání svého projektu *Moje šťastná sedma* se soubory řešení *Šťastná sedma* na disku si znovu otevřete projekt *Moje šťastná sedma* a připravte se na jeho kompilaci do podoby spustitelného souboru. Pokud jste projekt *Moje šťastná sedma* nevytvářeli, použijte k dokončení cvičení soubory řešení, které jsem pro vás připravil.

Sestavení spustitelného souboru

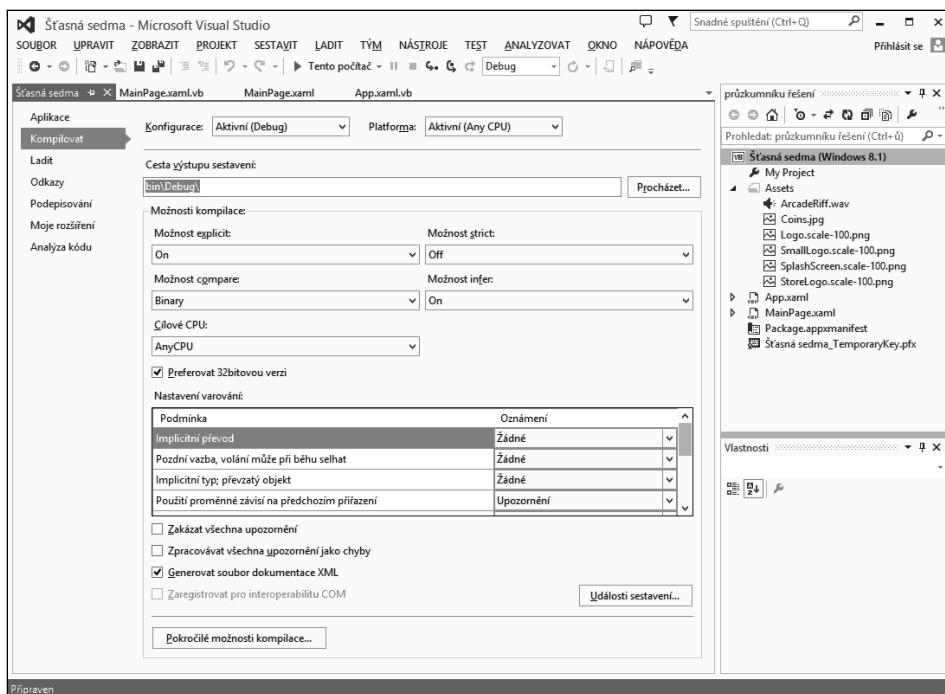
Vášim posledním úkolem v této kapitole je dokončení celého procesu vývoje, tj. vytvoření aplikace pro Windows neboli vytvoření *spustitelného souboru*. Aplikace pro Windows vytvořené v sadě Visual Studio jsou charakteristické příponou *.exe*. Přitom platí, že takové soubory lze spustit na jakémkoliv systému Windows obsahujícím nezbytné podpůrné soubory. Je-li vaším cílem distribuce aplikace prostřednictvím Windows Store, bude celý balíček pro nasazení bezpečně přenesen na Windows Store a bude zde zpřístupněn zákazníkům, kteří si jej budou chtít stáhnout. Vy se však můžete rozhodnout nasadit aplikaci na jednotlivé počítače, a to spuštěním Windows přímo z prostředí sady Visual Studio.

Jelikož jste právě vytvořili aplikaci navrženou pro Windows Store a operační systém Windows 8.1 pak podmínkou spuštění tohoto konkrétního programu je právě operační systém Windows 8.1. Tuto ukázkovou aplikaci nebudeme zatím vystavovat na Windows Store, neboť dosud nebyla registrována a důkladně otestována. Můžete však tuto aplikaci nasadit na svém vlastním počítači, což je proces, který není provázen takovými požadavky na registraci jako nasazení prostřednictvím Windows Store.

Aby vám sada Visual Studio usnadnila proces testování a kompilace, umožňuje vytváření dvou typů spustitelných souborů, souvisejících s vašimi aplikacemi pro Windows: *sestavení pro ladění* (*debug build*) a *sestavení pro uvolnění* (*release build*).

Sestavení pro ladění jsou sadou Visual Studio automaticky vytvářena vždy, když se během příprav své aplikace rozhodnete ji otestovat. Tato sestavení jsou ukládána do složky *bin\Debug*, kterou najdete ve složce příslušného projektu. Spustitelný soubor vytvořený pro účely ladění obsahuje další informace pro ladění, díky nimž program běží o něco pomaleji.

Sestavení pro uvolnění jsou optimalizované spustitelné soubory, ukládané do složky *bin\Release*, kterou najdete ve složce příslušného projektu. Chcete-li upravit nastavení sestavení pro uvolnění, pak v nabídce Projekt (Project) zvolte *Vlastnosti název_projektu* (*Název_projektu Properties*) a poté přejděte na kartu *Kompilovat* (Compile). Zde najdete řadu voleb pro kompilaci, jejichž ukázkou vidíte na následujícím obrázku. Součástí panelu nástrojů Standardní (Standard) sady Visual Studio je i seznam *Konfigurace řešení* (Solution Configurations), z něhož je patrné, zda vytvářený spustitelný soubor je určen pro ladění (volba *Debug*) či pro uvolnění (volba *Release*).



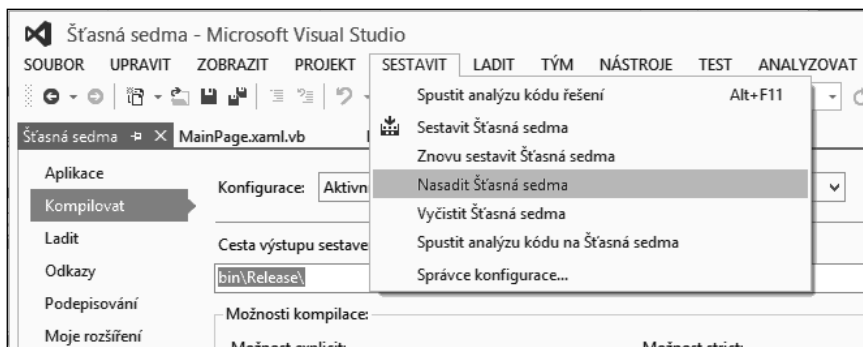
Proces přípravy spustitelného souboru pro určitý počítač je nazýván *nasazením aplikace*. Jak jsme si již řekli, rozhodnete-li se nasadit aplikaci v sadě Visual Studio, vývojové prostředí zajistí celý proces kopírování všech spustitelných a doprovodných souborů, které potřebujete k zaregistrování programu v operačním systému a jeho následnému spuštění. Visual Studio vám umožňuje nasazení aplikace *lokálně* (tj. na tom počítači, na kterém právě pracujete) anebo *vzdáleně* (tj. na počítači připojeném k síti či internetu).

V následujících krocích nasadíte sestavení pro uvolnění aplikace *Moje šťastná sedma* na lokálním počítači a na obrazovce Start systému Windows vytvoříte ikonu této aplikace.

Nasazení sestavení pro uvolnění aplikace Šťastná sedma

1. Na panelu nástrojů Standardní (Standard) rozbalte seznam Konfigurace řešení (Solution Configurations) a vyberte z něj volbu Release. Visual Studio připraví váš projekt tak, aby bylo možné vytvořit sestavení pro uvolnění. To znamená, že výsledný spustitelný soubor nebude obsahovat žádné informace pro ladění. Výstupní cesta, do níž bude sestavení uloženo, se změní na bin\Release.
2. V nabídce Sestavit (Build) klepněte na Nasadit Šťastná sedma (Deploy Šťastná sedma).

Příkaz Nasadit (Deploy) vytvoří složku bin\Release (pokud tato složka zatím neexistuje), určenou pro uložení souborů projektu, a následně zkompiluje zdrojový kód vašeho projektu. V průběhu tohoto procesu se může zobrazit okno Výstup (Output), informující vás o jednotlivých milnících procesu sestavení a nasazení. Výsledkem je spustitelný soubor s názvem Šťastná sedma.exe, který sada Visual Studio zaregistruje v operačním systému vašeho počítače.



Visual Studio nasadí aplikaci lokálně, neboť na panelu nástrojů Standardní (Standard), hned vedle tlačítka pro spuštění, je vybráno Tento počítač (Local Machine). V našem případě se jedná o požadované chování, nicméně vy máte samozřejmě možnost nasadit svoji aplikaci i na vzdálený počítač (tj. počítač připojený k tomu vašemu prostřednictvím počítačové sítě či internetu). K tomuto účelu je určena volba Vzdálený počítač (Remote Machine) výše zmíněného seznamu. Vyberete-li však tuto volbu, sada Visual Studio zobrazí dialog požadující zadání bližších informací týkajících se vzdáleného počítače. Zapamatujte si ale, že nasazení na vzdálený počítač je určeno a užíváno především pro účely testování. Nejlepší způsob nasazení či instalace hotových aplikací prostřednictvím internetu totiž představuje Windows Store.

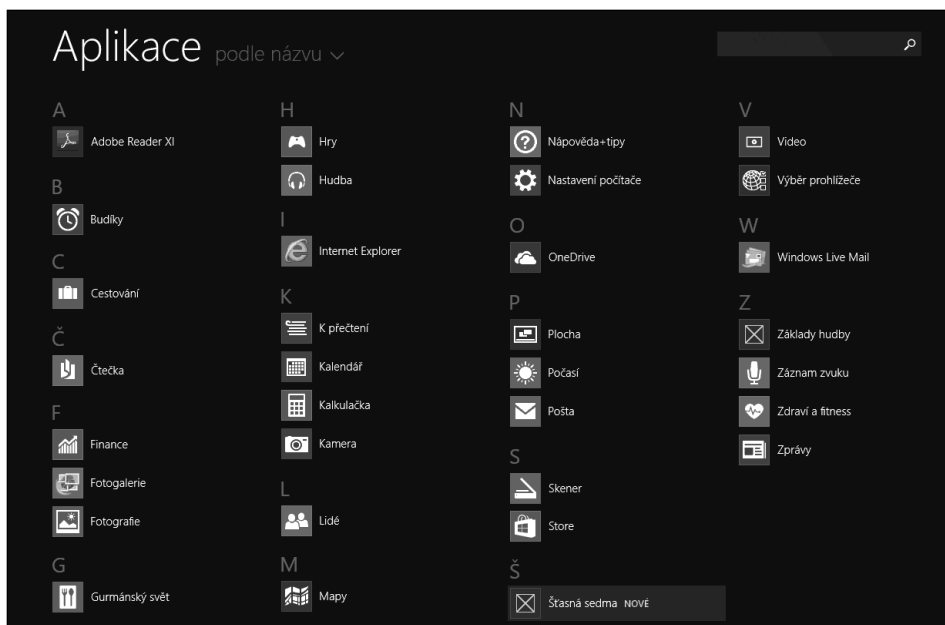
Nasadíte-li nějakou aplikaci určenou pro uživatelské rozhraní systému Windows 8.1 systém Windows automaticky vytvoří na obrazovce Start ikonu tohoto nového programu. Tuto ikonu pak můžete použít ke spuštění programu. Zkuste si nyní spustit program Moje šťastná sedma pomocí ikony na obrazovce Start vašeho počítače.

3. Otevřete si obrazovku Start a přejděte na seznam nainstalovaných aplikací.
4. Vaše aplikace se může nacházet na dvou místech: buď přímo na hlavní obrazovce Start, anebo na sekundární obrazovce Start, zvané Aplikace a obsahující dlouhý seznam nainstalovaných aplikací. (Operační systém Windows 8.1 mého počítače umístil ikonu programu Šťastná sedma právě sem.)

Jelikož jsme nevytvořili žádnou barevnou dlaždici naší nové aplikace pro obrazovku Start, najdete na této obrazovce výchozí (tj. šedou) dlaždici. Vaše obrazovka se zřejmě bude podobat ukázce na následující straně (všimněte si aplikace Šťastná sedma ve třetím sloupci ikon):

5. Klepněte na ikonu aplikace Šťastná sedma. Program se spustí.
6. Znovu si aplikaci vyzkoušejte. Opakovaně klepejte na tlačítko Vytoč a pokuste se několikrát trefit jackpot. Po skončení testování aplikaci uzavřete.
7. Vraťte se do sady Visual Studio. Zavřete okno Výstup (Output) a dialog s vlastnostmi projektu Šťastná sedma. Všimněte si, že volby pro kompilaci si můžete prohlížet a měnit, kdykoliv chcete – stránka s těmito vlastnostmi je dostupná vždy.
8. V nabídce Soubor (File) klepněte na Konec (Exit) a ukončete tak práci v sadě Visual Studio. Tím je samozřejmě ukončena i veškerá práce na projektu Šťastná sedma.
9. Budete-li k tomu vyzváni, klepněte na volbu Uložit (Save). Vývojové prostředí sady Visual Studio se uzavře.

Gratuluji vám k dokončení vaší první aplikace pro Windows Store!



Shrnutí

V této kapitole najdete podrobný popis vytvoření aplikace Šťastná sedma v sadě Visual Studio 2013 a jazyce Visual Basic. Přitom tato aplikace je určena pro Windows Store. Jak vidíte, proces vývoje se značně podobá procesu vývoje v předcházejících verzích sady Visual Studio a jazyka Visual Basic. Na stránku aplikace přidáte ovládací prvky pomocí Panelu nástrojů (Toolbox) a nastavíte jejich vlastnosti. V dalším kroku přejdete k psaní kódu programu. Po jeho dokončení se začnete věnovat testování aplikace a přípravám pro její nasazení. Je však nutné zdůraznit, že Panel nástrojů (Toolbox) obsahující ovládací prvky jazyka XAML se značně liší od Panelu nástrojů (Toolbox) používaného při vytváření aplikací využívajících Windows Forms a určených pro Windows desktop. V této kapitole jsme si krok za krokem ukázali způsob použití ovládacích prvků jazyka XAML. V kapitole následující se naopak budeme věnovat Panelu nástrojů (Toolbox) Windows Forms, přičemž naším cílem bude vytvoření desktopové aplikace pro Windows 8.1, Windows 8 či Windows 7.

Při práci na automatové hře Šťastná sedma jste si vyzkoušeli použití ovládacího prvku typu *TextBlock*, ovládacího prvku typu *Button*, ovládacího prvku typu *Image* a ovládacího prvku typu *MediaElement*. Kromě toho jste si i zkusili nastavit barvu pozadí ovládacího prvku typu *Grid*. Dále jste se dozvěděli, jak pomocí Editoru obrázků (Image Editor) sady Visual Studio můžete vytvořit spouštěcí obrazovku aplikace. A nakonec jste svoji aplikaci otestovali a nasadili ji na obrazovku Start svého počítače. S vynaložením nějaké další práce byste byli schopni aplikace podobající se aplikaci Šťastná sedma i šířit prostřednictvím Windows Store.

Aplikace pro Windows desktop: stručný úvod do používání Windows Forms

Po dokončení studia této kapitoly budete schopni

- Vytvořit aplikaci pro Windows desktop s využitím Windows Forms
- Navrhnout uživatelské rozhraní pomocí Návrháře Windows Forms
- Používat ovládací prvky, které jsou součástí Panelu nástrojů Windows Forms
- Napsat v jazyce Visual Basic kód zpracovatele událostí
- Spustit, otestovat a nasadit aplikaci, určenou pro Windows desktop

Součástí kapitoly 3 „Vytvoření první aplikace pro Windows Store“ byla cvičení, která vás krok za krokem provázela procesem tvorby aplikace pro Windows Store, navrhované v IDE sady Visual Studio 2013 a v jazyce Visual Basic. Aplikace pro Windows Store představují pro vývojáře softwaru novou a zásadní obchodní příležitost. A jak tato platforma poroste, budete se dostávat do výhodné pozice pro to, abyste začali pocítovat přínosy nejen zlepšené funkcionality systému Windows, ale i obrovského dosahu globálního distribučního systému Windows Store.

Nicméně při výběru vývojové platformy pro aplikace, které budete vytvářet v jazyce Visual Basic, byste měli zvážit ještě další faktory. Například můžete pracovat ve společnosti, která bude i nadále podporovat starší verze systémů Windows, byť bude současně procházet procesem přechodu směrem k vývoji aplikací pro Windows Store. Anebo můžete pracovat ve firmě, která již dříve investovala značné prostředky do vývoje zdrojového kódu v některé z předcházejících verzí Visual Basicu. Může se například jednat o uživatelská rozhraní navržená s využitím technologií Windows Forms či Windows Presentation Foundation (WPF). Což ovšem znamená, že i když Windows Store bezesporu představuje budoucnost v oblasti vývoje aplikací, současné softwarové platformy a technologie jsou přinejmenším stejně tak důležité. Proto se v této kapitole seznámíte s podporou, jakou sada Visual Studio 2013 nabízí pro starší verze systémů Windows a jazyka Visual Basic.

Nejprve se dozvíte, jak se pomocí poměrně efektivní technologie sady Visual Studio, známé pod označením *Windows Forms*, vytvářejí aplikace pro Windows desktop, podporující operační systémy Windows 8.1, Windows 8 a Windows 7. Přitom pojem „aplikace pro Windows desktop“ představuje pouze nové označení toho, co programátoři dlouhou dobu popisovali jako „aplikace založené na využití oken“ – tj. plně funkční programy běžící v rámci (okně) na pracovní ploše systému Windows a obsahující záhlaví okna, nabídky, dialogy, tlačítka a další

ovládací prvky. V této kapitole se vrátíme k automatové hře Šťastná sedma, kterou jste vyvíjeli v kapitole 3, avšak tentokrát zkusíme celý projekt v sadě Visual Studio 2013 vytvořit jako aplikaci pro Windows desktop.

Celé uživatelské rozhraní této automatové hry sestavíte z ovládacích prvků, které najdete v Panelu nástrojů (Toolbox) Windows Forms. K vytvoření okna aplikace neboli *formuláře* (anglicky *form*) využijete Návrhář (Designer) Windows Forms, který je součástí IDE sady Visual Studio. Dále budete pokračovat úpravami formuláře a jeho ovládacích prvků pomocí nastavení vlastností a ke generování náhodných čísel a vyvolávání speciálních efektů využijete kód dvou zpracovatelů událostí, napsaný v jazyce Visual Basic. Pro přehrání hudby vám poslouží objekt platformy .NET Framework, který se jmenuje *My.Computer.Audio*. Při vytváření této aplikace se vám budu snažit ukázat a zdůraznit podobnosti a rozdíly mezi aplikacemi pro Windows desktop a aplikacemi pro Windows Store. Kromě toho vám také ukážu, jakým způsobem se dá přecházet z jedné platformy do druhé a zase zpět. Ve druhé části této knihy, nazvané „Návrh uživatelského rozhraní“ najdete spoustu ukázek kódu napsaných pro obě platformy.

Tato kapitola bude přínosná především pro ty z vás, kteří máte programátorské zkušenosti s některou z předcházejících verzí jazyka Visual Basic. Sami uvidíte, že svoje stávající vývojové dovednosti můžete poměrně rychle využít i v IDE sady Visual Studio 2013. A pokud jste používali Windows Forms, pak zjistíte, že svoje stávající programy můžete poměrně snadno udržovat a dále rozvíjet i v sadě Visual Studio 2013, přičemž ale budete mít k dispozici veškerá vylepšení této nové verze vývojového prostředí. V průběhu čtení dalších kapitol této knihy se pak budete postupně dozvídat, co je v jazyce Visual Basic nového a vylepšeného.

Uvnitř aplikací pro Windows desktop

A co vlastně *představuje* aplikace pro Windows desktop? Jak jsme si řekli již v předcházející části, aplikace pro Windows desktop je v podstatě aplikace pro systém Windows navržená tak, aby ji bylo možné provozovat v prostředí operačních systémů Windows 8.1, Windows 8 či Windows 7. Tyto „tradiční“ aplikace prezentují svoje funkce pomocí viditelných a trvalých prvků uživatelského rozhraní, jimiž jsou například záhlaví okna, panely nástrojů, příkazy nabídek, tlačítka, dialogová okna, posuvníky, stavové řádky a další. Ačkoliv směrnice pro vývoj aplikací pro Windows Store požadují, aby vývojáři minimalizovali, či dokonce zcela vynechali využití mnohých těchto prvků tradičního uživatelského rozhraní (v případě mobilních zařízení a tabletů je totiž volný prostor obrazovky velmi vzácný), společnost Microsoft uznává, že tisíce oblíbených aplikací pro systémy Windows tyto prvky využívají a že zákazníci chtějí s těmito aplikacemi pracovat nejen nyní, ale i v budoucnu.

Aplikace pro Windows desktop mají také jinou strukturu než aplikace pro Windows Store: používají jiné ovládací prvky a komponenty, v běhovém prostředí Windows využívají jiné funkce a mají jiné požadavky na instalaci a zabezpečení. Z důvodu integrace aplikací pro Windows desktop do operačního systému Windows 8.1 se součástí tohoto systému stalo samostatné provozní prostředí pro spouštění těchto aplikací. Toto prostředí je reprezentováno dlaždicí Plocha (Desktop) na obrazovce Start, sloužící k jeho otevření a umožňující snadnou interakci mezi systémem Windows 8.1 a aplikacemi pro Windows desktop.

Následující obrázek ukazuje obrazovku Start systému Windows 8.1 s dlaždicí Plocha, viditelnou v horním levém rohu obrazovky. Aplikace pro Windows desktop běží v tomto prostředí v systému Windows 8.1, Windows 8 či i na tradiční ploše systému Windows 7.



Visual Basic a aplikace pro Windows desktop

V rámci sad Visual Studio 2013 Professional, Premium a Ultimate typ projektu Windows Forms představuje vyzkoušený model, na jehož základě je možné vytvářet aplikace pro Windows desktop. Jelikož tento typ projektu v podstatě je původním modelem uživatelského rozhraní pro aplikace napsané v jazyce Visual Basic, integrované vývojové prostředí sady Visual Studio 2013 je velmi dobře připraveno pro vytváření aplikací s využitím technologie Windows Forms. Součástí IDE je Návrhář (Designer) Windows Forms a Panel nástrojů (Toolbox) Windows Forms, přičemž obě tyto komponenty IDE se snadno používají, mají velké množství funkcí a jsou vyzrálé.

Z čistě technického hlediska lze Windows Forms považovat za komponentu chytrého klienta platformy .NET Framework, představující sadu spravovaných knihoven umožňujících a usnadňujících provádění některých základních úkonů v aplikacích, jako například čtení ze souborového systému či zapisování do souborového systému. V sadě Visual Studio se aplikace využívající Windows Forms vytvářejí na základě tříd, které jsou součástí jmenného prostoru *System.Windows.Forms*.

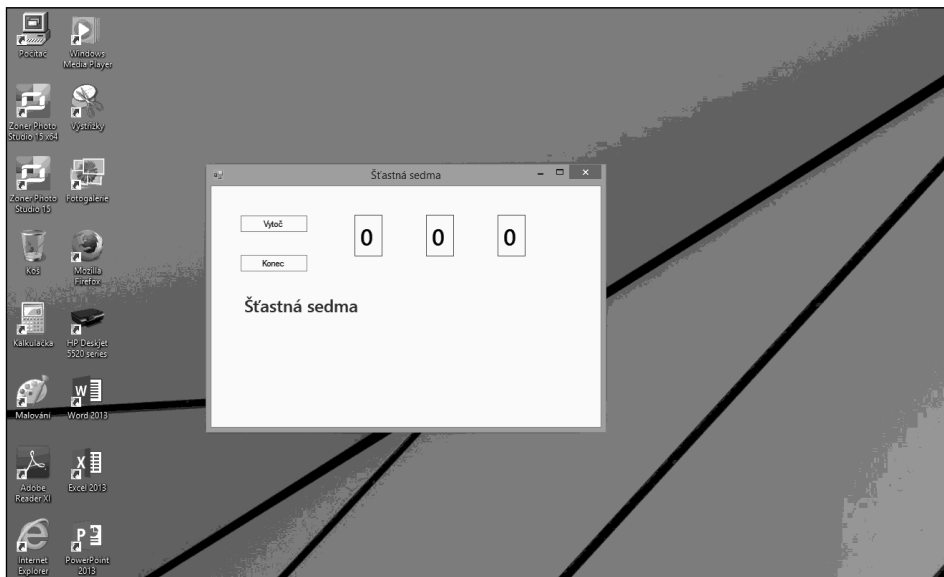
Vytváříte-li aplikaci pro Windows desktop využívající technologii Windows Forms, pak pracujete s ovládacími prvky, objekty, vlastnostmi a událostmi, které jsou dostupné přinejmenším od verze Visual Studio 2005. Avšak protože vy pracujete s nejnovější verzí sady Visual Studio, máte možnost využívat ta vylepšení a ty novinky, které souvisejí s operačním systémem Windows 8.1, platformou .NET Framework 4.5.1 a rozšířeními programovacího jazyka Visual Basic, dále se vyvíjejícího s každou novou verzí sady Visual Studio. V důsledku toho můžete využít svoje stávající programátorské dovednosti, získané při práci s technologií Windows Forms, pro vytváření nových či údržbu stávajících aplikací pro Windows desktop v sadě Visual

Studio 2013. A během rozšiřování stávajícího kódu o nové funkce budete získávat další zkušenosti a dovednosti, vyplývající z používání novější technologie.

Kromě toho můžete pro tvorbu uživatelského rozhraní a funkcí své aplikace pro Windows desktop v sadě Visual Studio 2013 využít i technologii Windows Presentation Foundation (WPF). Tato vyzrálá technologie (poprvé představená spolu se sadou Visual Studio 2008) vám umožňuje navrhování uživatelských rozhraní s využitím značkovacího jazyka XAML a efektivních ovládacích prvků WPF. Avšak protože použití sady Visual Studio a technologie WPF se do značné míry podobá vytváření aplikací pro Windows Store v sadě Visual Studio 2013 (zejména tedy s ohledem na využití jazyka XAML a ovládacích prvků stylu WPF), soustředím se v této knize na problematiku vývoje aplikací pro Windows Store a – alternativně – problematiku vývoje aplikací pro Windows desktop s využitím technologie Windows Forms.



Poznámka: Chcete-li si v rámci studia této kapitoly vytvořit vlastní aplikaci pro Windows desktop, musíte mít k dispozici sadu Visual Studio 2013 ve verzi Professional, Premium či Ultimate. Pouze tyto verze totiž obsahují šablony pro vývoj aplikací pro Windows desktop. Sada Microsoft Visual Studio Express 2013 for Windows podporu takových aplikací nenabízí. Nemáte-li ale k dispozici Visual Studio 2013 Professional či Ultimate, můžete si z webu sady Microsoft Visual Studio (<http://www.visualstudio.com/en-us/products/visual-studio-express-vs.aspx>) zdarma stáhnout sadu Microsoft Visual Studio Express 2013 for Windows Desktop. Toto je tedy alternativní řešení, které můžete získat zdarma a které vám umožní dokončení veškerých cvičení nejen v této kapitole, ale i ve všech dalších částech této knihy, v nichž se bude hovořit o programování s využitím technologie Windows Forms.



V následujících částech této kapitoly budete vytvářet automatovou hru Šťastná sedma ve verzi pro Windows desktop. Výsledek vašeho snažení by se měl podobat tomu, co vidíte na předchozím obrázku. Ačkoliv uživatelské rozhraní této verze se na první pohled poněkud liší od uživatelského rozhraní verze pro Windows 8.1, kód v jazyce Visual Basic, který je základem obou verzí, je velice podobný. A právě tato podobnost je jedním z důležitých závěrů této

kapitoly – ačkoliv technologie uživatelského rozhraní se mezi jednotlivými platformami liší, základní logika aplikace je často velice podobná, neboť vychází ze stejných jazykových struktur a slovníku jazyka Visual Basic.

Vytvoření aplikace pro Windows desktop

V tomto cvičení zahájíte vývoj celé aplikace. Šťastná sedma tím, že nejprve vytvoříte nový projekt aplikace pro Windows desktop a následně v něm pomocí Panelu nástrojů (Toolbox) Windows Forms připravíte uživatelské rozhraní.

Vytvoření nového projektu

1. Spustíte sadu Visual Studio 2013.
2. Z nabídky Soubor (File) zvolte Nový projekt (New Project).



Tip: Nový projekt můžete vytvořit také tak, že na Úvodní stránce (Start page) sady Visual Studio klepnete na modře zvýrazněný odkaz Nový projekt (New Project), nacházející se v levé části této stránky.

Otevře se dialog Nový projekt (New Project), na jehož levé straně budou vyjmenovány dostupné šablony projektů, uspořádané do kategorií. Jak jsme si řekli již v kapitole 3, šablony jazyka Visual Basic jsou uvedeny podle typu, přičemž pod nimi pak následují šablony platné pro jiné jazyky, jako například Visual C#, Visual C++ či JavaScript.

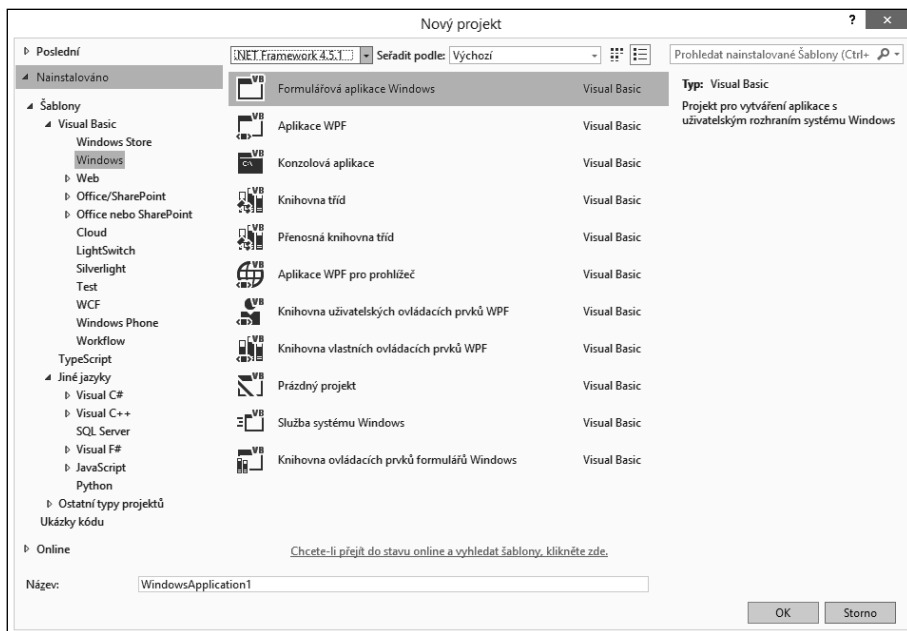
V horní části tohoto dialogu si také všimněte důležitého rozbalovacího seznamu. Díky němu totiž můžete říci, pro jakou verzi platformy .NET Framework bude vaše aplikace určena. Tato funkce se také někdy nazývá *vícenásobným cílením* (*multitargeting*), což znamená, že díky ní můžete vybrat cílové prostředí, pro něž bude váš program určen. Například pokud v tomto seznamu ponecháte výchozí volbu .NET Framework 4.5.1, pak na kterémkoliv počítači, jehož uživatel si bude chtít vaši aplikaci spustit, bude muset být nainstalovaná platforma .NET Framework 4.5.1. (Obvykle se tedy bude jednat o počítač s operačním systémem Windows 8.1.) Nemáte-li tedy nějaké specifické požadavky týkající se prostředí, v němž má být vaše aplikace provozována, můžete toto nastavení ponechat beze změny. Potřebujete-li však zajistit podporu starších verzí operačních systémů Windows (což také znamená starší verzi platformy .NET Framework), můžete zde vybrat potřebnou verzi.



Poznámka: Výše popsaný seznam není součástí verze Visual Studio Express 2013 for Windows Desktop. Kromě toho tato verze nabízí i menší počet šablon aplikací. Nicméně pro dokončení všech cvičení v této kapitole je zcela postačující.

3. V levé části dialogu Nový projekt (New Project) rozbalte kategorii Visual Basic (není-li již rozbalená) a poté přejděte do kategorie Windows.

Dialog Nový projekt (New Project) by měl nyní vypadat takto:



4. **Není-li dosud vybrána, pak ve střední části dialogu Nový projekt (New Project) klepněte na volbu Formulářová aplikace Windows (Windows Forms Application).**

Visual Studio připraví vývojové prostředí pro vývoj aplikace pro Windows desktop v jazyce Visual Basic.

5. **Ve spodní části dialogu Nový projekt (New Project) se nachází pole Název (Name), do něhož zadejte **MojeŠťastnáSedmaWF**.**

Sada Visual Studio přiřadí vašemu projektu název **MojeŠťastnáSedmaWF**. (Složku, do níž budou ukládány soubory projektu, zadáte později.) Jak jsem zmínil již v kapitole 3, doporučuji vám, abyste použili prefix **Moje** a zabránili tak případné neúmyslné záměně s projektem **ŠťastnáSedmaWF**, který jsem pro vás připravil a který najdete na disku.

K názvu projektu jsem přidal i doplněk „WF“, abych tak jednoznačně naznačil, že se jedná o aplikaci využívající technologii Windows Forms. (Platí však, že tyto aplikace nevyžadují používání žádných specifických názvů.)

6. **Klepnutím na tlačítko OK vytvoříte svůj nový projekt.**

Sada Visual Studio uzavře dialog **Nový projekt (New Project)** a namísto něj zobrazí okno **Návrhář (Designer)**, obsahující prázdný formulář Windows. To znamená, že můžete začít s vývojem uživatelského rozhraní.

Nyní formulář zvětšíme a přidáme do něj dvě tlačítka.

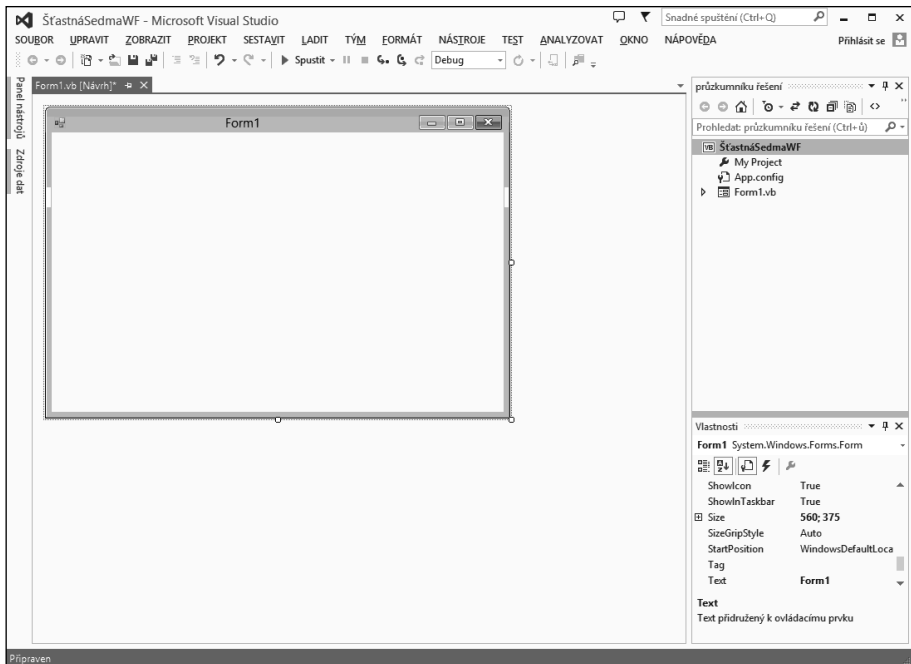
Vytvoření uživatelského rozhraní

1. **Ukazatel myši přesuňte nad spodní pravý roh formuláře v okně Návrhář (Designer). Ukazatel by se měl změnit ze standardního ukazatele na ukazatel pro změnu velikosti (oboustranná šipka). V tuto chvíli stiskněte tlačítko myši a tažením zvětšíte velikost formuláře tak, aby se do něj vešly všechny nezbytné objekty.**

Při změně velikosti formuláře se může stát, že se po stranách okna Návrháře (Designer) zobrazí posuvníky, umožňující vám přístup k těm částem formuláře, které v danou chvíli nejsou zobrazené. V závislosti na rozlišení vaší obrazovky a na počtu nástrojů, které máte v sadě Visual Studio otevřené, totiž může dojít k tomu, že nebude možné zobrazit celý formulář najednou.

Velikost svého formuláře upravte tak, aby zhruba odpovídala velikosti formuláře znázorněného na následujícím obrázku. Chcete-li, aby váš projekt přesně odpovídal mémuto vzoru, pak můžete použít šířku a výšku formuláře (560 pixelů x 375 pixelů) ukázanou ve spodním pravém rohu obrazovky.

Chcete-li si zobrazit celý formulář bez jakýchkoliv překážek, můžete si změnit velikost dalších nástrojů pro programování, případně můžete okna těchto nástrojů zcela uzavřít. Příslušné postupy jsme si ukázali v kapitole 2 „Integrované vývojové prostředí Visual Studio“ (Máte-li tedy jakékoliv potíže při provádění zmíněných úkonů, vraťte se ke kapitole 2.)



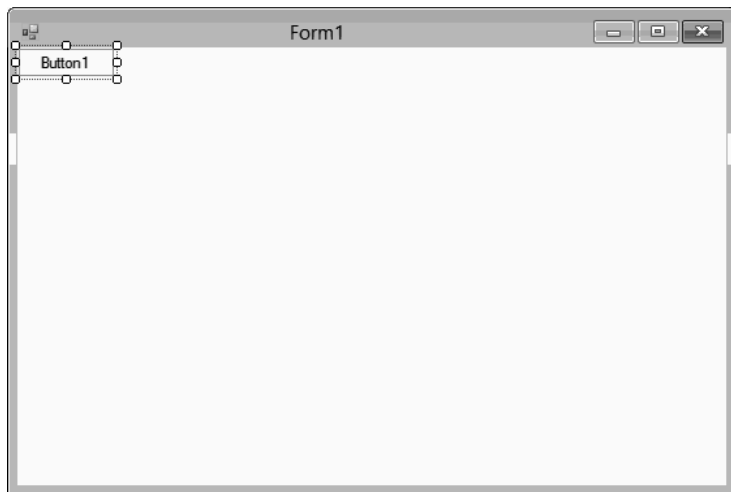
Nyní do formuláře přidáte tlačítko.

2. Na levé straně IDE klepněte na záložku Panel nástrojů (Toolbox) a zobrazte si tak panel s nástroji pro vývoj formulářových aplikací Windows.

Jak vidíte, tento Panel nástrojů (Toolbox) obsahuje jiné ovládací prvky než ten, se kterým jsme pracovali v kapitole 3. Namísto ovládacích prvků aplikací pro Windows Store vám tento Panel nástrojů (Toolbox) nabízí ovládací prvky aplikací využívajících technologii Windows Forms. A odlišné jsou nejen názvy samotných ovládacích prvků: odlišné jsou totiž i vlastnosti a události těchto prvků. Ovládací prvky technologie Windows Forms jsou uspořádány do kategorií, přičemž pokud si rozbalíte kategorii Všechny Windows formuláře (All Windows Forms), uvidíte, kolik takových ovládacích prvků máte k dispozici. To je nepřímý důkaz toho, jak dlouho je již tato technologie při programování využívána.

3. V kategorii Všechny Windows formuláře (All Windows Forms) poklepejte na ovládací prvek *Button* a poté přesuňte ukazatel myši mimo okno Panelu nástrojů (Toolbox).

Visual Studio vytvoří ve formuláři objekt tlačítka, majícího nějakou standardní velikost. Současně dojde ke skrytí okna Panelu nástrojů (Toolbox). Ukázkou formuláře s tlačítkem vidíte níže:



Tlačítku je automaticky přiřazen název *Button1*, neboť se jedná o první tlačítko, které jste ve svém novém projektu vytvořili. Všimněte si, že nové tlačítko je automaticky vybráno a jsou zvýrazněny úchopy pro změnu jeho velikosti. Pracuje-li totiž sada Visual Studio v režimu návrhu (tj. v takovém režimu, během něž program vyvíjíte, avšak nespouštíte), můžete jednotlivé objekty na formuláři přesouvat tažením pomocí myši. V tomto režimu můžete jejich velikost měnit pomocí zobrazených úchopů. Avšak v době běhu programu uživatel jednotlivé prvky uživatelského rozhraní přesouvat nemůže s výjimkou případů, kdy nastavením určité vlastnosti programu takové úkony umožníte.

4. Přejděte do okna Vlastnosti (Properties) a vlastnost *Name* upravte tak, aby obsahovala text **TlačítkoVytoc**.

Vlastnost *Name* je zobrazena v horní části okna Vlastnosti (Properties). Přiřazením jedinečného názvu každému objektu uživatelského rozhraní své aplikace si usnadníte následnou práci s těmito objekty v kódu programu. A jak jsme si řekli již v kapitole 3, máte-li k tomu dostatek prostoru, doporučuji vám, abyste do názvu objektu přidali i název typu ovládacího prvku.

Nyní do formuláře přidáme druhé tlačítko, které umístíme pod první.

Přidání druhého tlačítka

1. Na levé straně IDE klepněte na záložku Panel nástrojů (Toolbox) a zobrazte si tak panel s nástroji pro vývoj formulářových aplikací Windows.
2. V kategorii Všechny Windows formuláře (All Windows Forms) klepněte (ano, tentokrát pouze klepněte) na ovládací prvek *Button* a poté přesuňte ukazatel myši nad formulář. Ukazatel myši se změní na křížek a ikonu tlačítka. Křížek je použit proto, aby vám usnadnil kreslení potřebného pravouhlého tvaru tlačítka na formuláři. Tuto metodu můžete použít

jako alternativu k poklepání na ovládací prvek *Button*, vytvářejícímu vždy ovládací prvek o výchozí velikosti.

3. Klepněte do formuláře a označte tak polohu levého horního rohu tlačítka. Následným tažením směrem doprava a dolů naznačte velikost tlačítka. Uvolněním tlačítka myši pak objekt dokončete.
4. Velikost nového tlačítka upravte tak, aby odpovídala velikosti prvního přidaného tlačítka.
5. Obě tlačítka přesuňte poněkud doprava dolů, aby se nedotýkala okraje formuláře. (Při přesunu využijte automaticky zobrazované přichytávací čáry /snapline/.)
6. Přejděte do okna Vlastnosti (Properties) a vlastnost *Name* druhého tlačítka upravte tak, aby obsahovala text **TlačítkoKonec**.

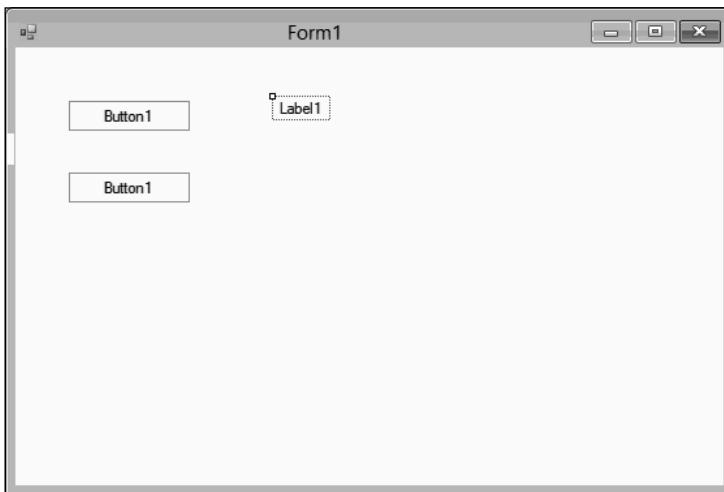


Tip: Pokud potřebujete, kterýkoliv objekt formuláře můžete vymazat a následně začít znovu tak, že objekt ve formuláři vyberete a poté stisknete klávesu Delete. Procvičte si proces tvorby objektů uživatelského rozhraní tím, že několik dalších objektů vytvoříte a zase smažete.

Nyní do formuláře přidáme popisky sloužící k zobrazení náhodných čísel, vypočítaných programem. Přitom pojmem *popisek (label)* se míní speciální prvek uživatelského rozhraní aplikací využívajících technologii Windows Forms, určený k zobrazování textů, čísel či symbolů v době běhu programu. Naše hra by měla fungovat tak, že pokud uživatel klepne na tlačítko *Vytoč*, ve třech popiscích se objeví vypočítaná náhodná čísla. Bude-li jedno z nich rovno 7, uživatel vyhrává.

Přidání popisků pro náhodná čísla

1. Otevřete si okno Panelu nástrojů (Toolbox) a poklepejte na ovládací prvek *Label*. Sada Visual Studio vytvoří ve formuláři objekt popisku. Jak vidíte, velikost objektu prakticky přesně odpovídá velikosti zobrazeného textu, což znamená, že v tuto chvíli je objekt spíše malý. Velikost objektu však lze změnit.
2. Přesuňte objekt *Label1* doprava od objektů obou tlačítek. Po provedení těchto úprav by váš formulář měl vypadat zhruba takto (všimněte si také pozice obou tlačítek):



3. Znovu si otevřete okno Panelu nástrojů (Toolbox) a poklepejte na ovládací prvek *Label*. Vytvoříte tak druhý objekt popisku.

Tento objekt bude v programu nazván *Label2*.

4. V okně Panelu nástrojů (Toolbox) poklepejte na ovládací prvek *Label* a vytvoříte třetí objekt popisku.

5. Druhý a třetí objekt popisku přesuňte na formuláři doprava od prvního objektu popisku.

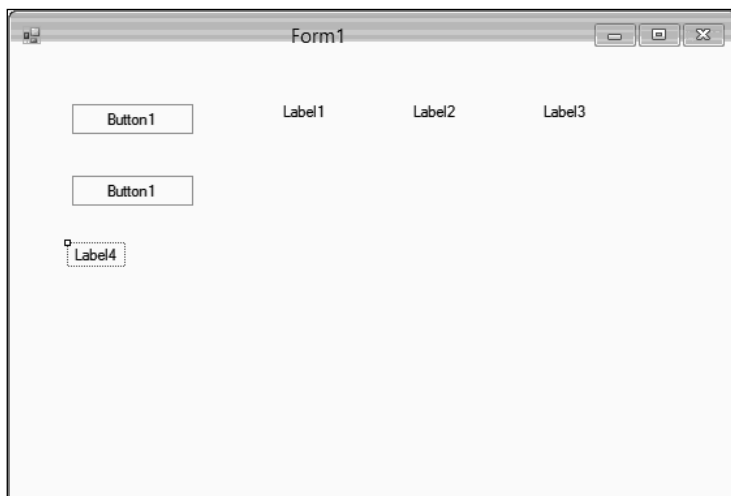
Mezi jednotlivými objekty popisků ponechte dostatek místa, neboť v době běhu programu budete v těchto objektech zobrazovat velká čísla.

Nyní využijeme ovládací prvek typu *Label* k přidání popisku formuláře. Toto tedy bude čtvrtý a také poslední objekt tohoto typu, který v programu použijeme.

6. V okně Panelu nástrojů (Toolbox) poklepejte na ovládací prvek *Label*.

7. Objekt *Label4* přesuňte pod obě tlačítka.

Po dokončení všech těchto kroků by váš formulář měl vypadat asi tak, jak vidíte na následující ukázce. (Nebudete-li po tomto porovnání spokojeni s polohou objektů jednotlivých popisků, můžete je přesunout.)



Nyní do formuláře přidáme okénko pro zobrazení obrázku, graficky ilustrujícího výhru, kterou vyhrajete, pokud při hře vytočíte číslo 7 a trefíte tak jackpot. Přitom použijeme objekt typu *PictureBox*, což je objekt, který byl navržen pro zobrazování rastrových obrázků, ikon, digitálních fotografií a dalších grafických dat v programech založených na technologii Windows Forms. Jedno z nejlepších využití tohoto objektu představuje zobrazení obrázku ve formátu JPEG. V našem programu využijeme stejný obrázek, jaký jsme použili i v kapitole 3 – tj. fotografii mincí z celého světa, mající tmavé pozadí.

Vložení obrázku

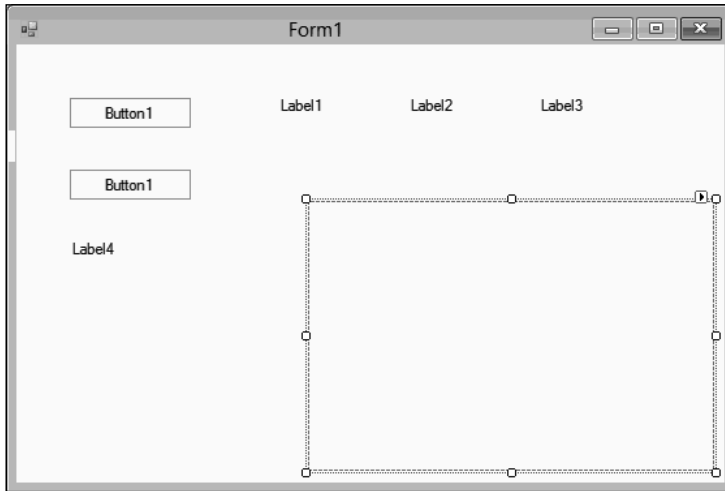
1. V okně Panelu nástrojů (Toolbox) klepněte na ovládací prvek *PictureBox*.

Ovládací prvek *PictureBox* je součástí Panelu nástrojů (Toolbox) pro Windows Forms již od verze 1 jazyka Visual Basic. Připomeňme si ale, že když jste v kapitole 3 vytvářeli aplikaci pro Windows Store, tak jste pro zobrazení fotografie na stránce použili ovládací prvek typu *Image*. Při přecházení mezi těmito dvěma variantami Panelu nástrojů (Toolbox) proto

vždy mějte na paměti, že odlišné typy projektů znamenají používání rozdílných ovládacích prvků, majících jiné vlastnosti a reagujících na jiné události.

2. Přesuňte se do okna Návrhář (Designer), a dokud máte vybraný ovládací prvek typu *PictureBox*, nakreslete ve formuláři velké pravoúhlé okno, nacházející se pod druhým a třetím popiskem.

Pod popisky ponechte dostatek místa, abyste tak umožnili jejich zvětšování. Po dokončení by váš formulář s objektem typu *PictureBox* měl vypadat asi takto:



V tuto chvíli má tento objekt ve vašem programu název *PictureBox1*. (Později tento název změníme na *ObrazekMinci*.)

Nyní vytvoříme složku Resources a do celého projektu pak přidáme fotografii Coins.jpg.

Přidání fotografie do složky Resources

1. Nemáte-li v tuto chvíli otevřené okno Průzkumník řešení (Solution Explorer), otevřete si jej výběrem volby Průzkumník řešení (Solution Explorer) z nabídky Zobrazit (View).

Jak již víte, Průzkumník řešení (Solution Explorer) vám umožňuje přístup k většině souborů tvořících váš projekt. V kapitole 3 jste si vyzkoušeli použití složky Assets v projektu aplikace pro Windows Store k úpravě spouštěcí obrazovky aplikace a dalších souborů. Složka Assets je však jedinečná pro aplikace navržené pro Windows Store; není tedy standardní součástí aplikací pro Windows desktop. Namísto ní můžete v okně Průzkumník řešení (Solution Explorer) vytvořit složku Resources, která vám umožní přístup k důležitým souborům a jejich uložení do projektu.

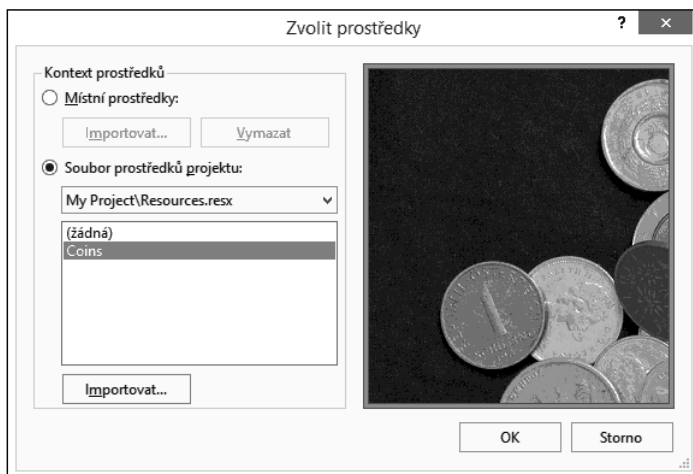
Přítom nejjednodušším způsobem vytvoření složky Resources je v tuto chvíli využití inteligentní značky, která je součástí objektu typu *PictureBox*.

2. Vraťte se zpět do vytvářeného formuláře a vyberte v něm objekt typu *PictureBox* (pokud ještě není vybrán).

Podíváte-li se nyní pozorně na hranice tohoto objektu, pak poblíž jeho pravého horního rohu objevíte malou šipku směřující doprava. Té se říká *inteligentní značka* (*smart tag*). Přítom inteligentní značkou se míní kontextově citlivé tlačítko, které můžete využít k rychlým změnám často používaných nastavení.

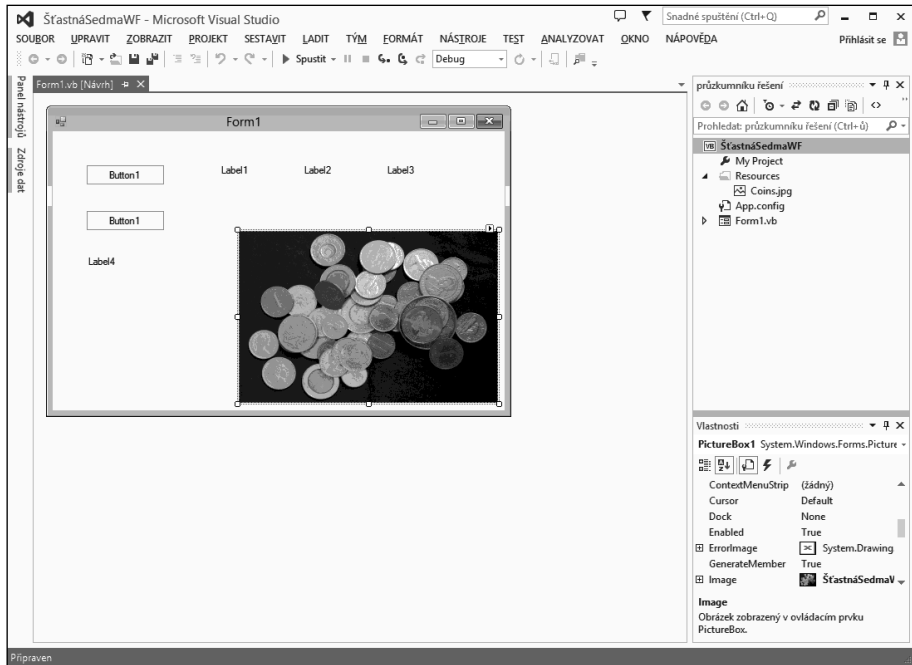
3. Klepněte na inteligentní značku, zobrazenou u objektu typu *PictureBox*, a otevřete si tak místní nabídku příkazů.
4. Klepnutím na odkaz Vybrat obrázek (Choose Image) si zobrazte dialog Zvolit prostředky (Select Resource).
5. V tomto dialogu vyberte volbu Soubor prostředků projektu (Project Resource File). Poté klepněte na tlačítko Importovat (Import). Vytvoříte tak složku Resources a budete do ní moci uložit soubor.
6. Poklepejte na soubor Coins.jpg, nacházející se ve složce Dokumenty\Visual_Basic_KzK\Kapitola 04.

Sada Visual Studio načte tento soubor, který se vzápětí zobrazí v dialogu Zvolit prostředky (Select Resource), jehož ukázkou vidíte níže:



Po přidání obrázku či souboru do projektu se tento obrázek či soubor stane součástí té aplikace, na níž právě pracujete, díky čemuž se na něj můžete začít odkazovat pomocí dialogu Průzkumník řešení (Solution Explorer) či okna Vlastnosti (Properties). A navíc bude tento obrázek či soubor automaticky zahrnut do aplikace i v době její kompilace, která je přípravou na distribuci – to ovšem znamená, že se vůbec nemusíte starat o to, kde ten obrázek či soubor byl původně uložen.

7. Klepněte na tlačítko OK.
V okně Průzkumník řešení (Solution Explorer) se nyní zobrazí složka Resources, obsahující soubor Coins.jpg.
8. V dialogu PictureBox – úlohy (PictureBox Tasks) přejděte do seznamu Režim velikosti (Size Mode) a vyberte v něm volbu StretchImage.
Jak vidíte, fotografie mincí z celého světa nyní vyplní celý objekt typu *PictureBox*, který je součástí vámi navrhovaného formuláře.
9. Upravte mezery mezi obrázkem a dalšími prvky uživatelského rozhraní tak, aby obrázek zabíral větší část plochy pravé spodní části formuláře.
Vaše obrazovka by nyní měla vypadat asi takto (všimněte si zejména složky Resources v Průzkumníku řešení /Solution Explorer/):



Nyní do složky Resources přidáme soubor se zvukem, který bude přehrán ve chvíli, kdy uživatel vyhraje hru. Tentokrát ale soubor přidáme pomocí dialogu Vlastnosti (Properties) našeho projektu.

Přidání souboru typu .wav do složky Resources

1. V nabídce Projekt (Project) zvolte Vlastnosti MojeŠťastnáSedmaWF (MojeŠťastnáSedmaWF Properties).
V IDE sady Visual Studio se otevře dialog Návrhář vlastností projektu (Project Properties Designer), v němž jsou nastavení projektu rozdělena do 12 kategorií.
2. Klepněte na kategorii Prostředky (Resources).
V okně Návrháře (Designer) uvidíte soubor Coins.jpg. V horním levém rohu okna Návrháře (Designer) se nachází rozbalovací seznam, nabízející 7 kategorií prostředků, přičemž v tuto chvíli je automaticky vybrána kategorie Obrázky (Images). Namísto zobrazování dlouhého seznamu všech prostředků najednou vám tento seznam dává možnost přehledného uspořádání jednotlivých prostředků do různých kategorií.
3. V rozbalovacím seznamu vyberte kategorii Zvuk (Audio).
V tuto chvíli je seznam prostředků tohoto typu prázdný. Nyní do projektu přidáme soubor ArcadeRiff.wav. Jedná se o ten soubor elektronické hudby, který jsme použili i v kapitole 3 k přehrání zvuku výherního automatu.
4. Klepněte na šipku zobrazenou vedle příkazu Přidat prostředek (Add Resource) a z rozbalené nabídky vyberte Přidat existující soubor (Add Existing File).
5. Přejděte do složky Dokumenty\Visual_Basic_KzK\Kapitola 04 a vyberte v ní soubor ArcadeRiff.wav. Poté klepněte na tlačítko Otevřít (Open).

Soubor *ArcadeRiff.wav* se zobrazí v okně *Resources*, čímž bude připravený pro použití v programu. Na tento prostředek se budete odkazovat až později, ve chvíli, kdy budete psát kód zpracovatele událostí.

6. Na panelu nástrojů *Standardní (Standard)* klepněte na tlačítko *Uložit vše (Save All)*.
7. Do pole *Umístění (Location)* dialogu *Uložit projekt (Save Project)* zadejte cestu *Dokumenty\Visual_Basic_KzK\Kapitola 04* a poté klepněte na tlačítko *Uložit (Save)*.
8. Jste-li vyzváni uložit či znovu načít projekt po vložení nových prostředků, odpovězte *Ano všem (Yes To All)*.
9. Uzavřete okno *Návrhář vlastností projektu (Project Properties Designer)*.

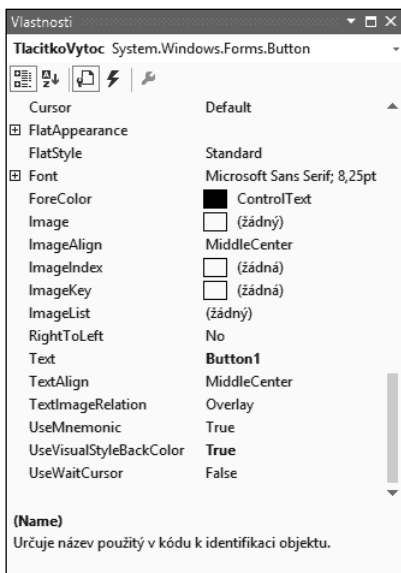
V tuto chvíli jste připraveni zahájit další úpravy uživatelského rozhraní programu, a to nastavením několika vlastností.

Nastavování vlastností

V kapitolách 2 a 3 jsme si řekli, že vlastnosti nějakého objektu můžete změnit tak, že příslušný objekt na stránce (či formuláři) nejprve vyberete a poté v okně *Vlastnosti (Properties)* upravíte jeho vlastnosti. Nyní nastal čas provést podobné změny i v naší aplikaci pro Windows desktop, a to konkrétně u obou tlačítek formuláře.

Nastavení vlastností tlačítka

1. Klepněte na první tlačítko (*Tlačítko Vytoč*) formuláře.
Tímto způsobem tlačítko vyberete, což poznáte podle toho, že bude zvýrazněno tečkovanou čarou a budou zobrazené úchopy pro změnu velikosti.
2. V horním okraji okna *Vlastnosti (Properties)* klepněte na tlačítko *Zařazené do kategorií (Categorized)*. (Jedná se o první tlačítko zleva.)
3. V případě potřeby si upravte velikost okna *Vlastnosti (Properties)* tak, aby nabízelo dostatek prostoru jak pro zobrazení názvů vlastností, tak i jejich nastavení.
K těmto vlastnostem patří například nastavení barvy pozadí, barvy textu, výšky písma či šířky tlačítka. Okno *Vlastnosti (Properties)* vašeho IDE by nyní mělo vypadat asi tak jak na obrázku na vpravo.
4. Není-li vlastnost *Text* dosud viditelná, pomocí svislého posuvníku se v dialogu *Vlastnosti (Properties)* přesuňte směrem dolů, do kategorie *Vzhled*. V ní pak vyhledejte vlastnost *Text*.
5. V prvním sloupci okna *Vlastnosti (Properties)* poklepejte na vlastnost *Text*.
V okně *Vlastnosti (Properties)* se zvýrazní aktuální nastavení této vlastnosti, tj. text „*Button1*“.
6. Zadejte **Vytoč** a poté stiskněte klávesu *Enter*.
Hodnota vlastnosti *Text* se změní na „*Vytoč*“. Tato změna se ihned projeví jak v okně *Vlastnosti (Properties)*, tak i na tlačítku ve formuláři. Nyní změníme vlastnost *Text* druhého tlačítka, a to tak, že do ní zadáme řetězec „*Konec*“.
7. V prvním řádku okna *Vlastnosti (Properties)* rozbalte seznam *Objekt (Object)*.
Jak vidíte, tento rozbalovací seznam obsahuje všechny objekty uživatelského rozhraní vašeho programu.
8. V tomto seznamu klepněte na *TlačítkoKonec* (což je druhé tlačítko programu).



V okně Vlastnosti (Properties) se zobrazí vlastnosti tohoto druhého tlačítka. Sada Visual Studio současně vybere toto tlačítko i ve formuláři.

9. Do vlastnosti *Text* zadejte **Konec** a stiskněte Enter.

Nápis na druhém tlačítku se změní na „Konec“.



Tip: Používání seznamu Objekt (Object) představuje poměrně jednoduchý a rychlý způsob přecházení mezi jednotlivými objekty programu. Mezi jednotlivými objekty programu však můžete přecházet také tak, že vždy klepnete na ten objekt, na který budete chtít právě přejít.

V následujících krocích se budeme věnovat úpravám vlastností popisků programu. První tři popisky jsou určeny pro zobrazování náhodných čísel, generovaných programem. To znamená, že všechny tyto popisky budou mít shodné vlastnosti, a proto většinu z nich nastavíme ve skupině. Poslední popisek, určený pro zobrazení popisného textu, bude mít poněkud odlišná nastavení.

Nastavení vlastností popisků pro zobrazení náhodných čísel

1. Přejděte do formuláře a klepněte v něm na první popisek pro zobrazení náhodného čísla (*Label1*). Poté stiskněte klávesu Shift a klepněte na druhý a třetí popisek. Následně uvolněte klávesu Shift. (Pokud vám ve výběru překáží zvětšené okno Vlastnosti /Properties/, přesuňte si jej někam jinam.)

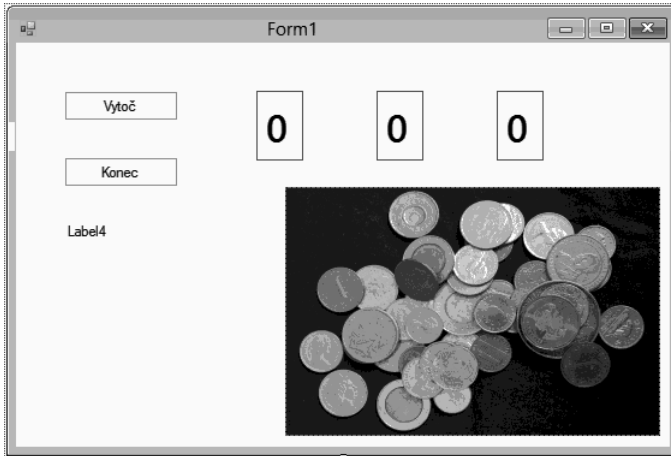
Okolo každého vámi vybraného popisku se zobrazí tečkovaný pravoúhelník s úchopem pro změnu polohy. Naším cílem je změnit vlastnosti *TextAlign*, *BorderStyle* a *Font*, a to tak, aby jednotlivá čísla byla zobrazována ve středu popisku, byla ohraničená a pro jejich zobrazení bylo použito stejné písmo o shodné velikosti. (Všechny tyto vlastnosti jsou v okně Vlastnosti /Properties/ součástí kategorie Vzhled /Appearance/.) Kromě toho vlastnosti *AutoSize* přiřadíte hodnotu *False*, a to proto, abyste velikost popisků mohli nastavovat přesně v souladu se svými požadavky. (Vlastnost *AutoSize* se nachází v kategorii Rozložení /Layout/.)



Poznámka: Je-li vybráno více objektů, jsou v okně Vlastnosti (Properties) zobrazeny pouze ty vlastnosti, které mohou být změněny pro všechny vybrané objekty, tj. pro celou skupinu.

2. V okně Vlastnosti (Properties) klepněte na vlastnost *AutoSize* a poté klepněte na šipku, která se zobrazí ve druhém sloupci.
3. Z rozbaleného seznamu vyberte *False*. Díky tomu budete moci velikost popisků zadávat ručně.
Všimněte si také, že ve formuláři se u jednotlivých popisků najednou zobrazí úchopy pro změnu velikosti.
4. Přejděte na vlastnost *TextAlign* a klepněte na ni. Poté opět klepněte na šipku, která se zobrazí ve druhém sloupci.
V seznamu se objeví grafické znázornění jednotlivých možností zarovnání textu. Tyto možnosti můžete využít k zarovnání textu do kterékoliv části objektu popisku.
5. Klepněte na prostřední možnost (*MiddleCenter*).
Hodnota vlastnosti *TextAlign* každého z vybraných popisků se změní na *MiddleCenter*.
6. Přejděte na vlastnost *BorderStyle* a klepněte na ni. Poté opět klepněte na šipku, která se zobrazí ve druhém sloupci.
V seznamu se zobrazí možná nastavení této vlastnosti (*None*, *FixedSingle* a *Fixed3D*).
7. V seznamu vyberte *FixedSingle*. Tímto krokem ke každému popisku přidáte tenké ohraničení.
8. Přejděte na vlastnost *Font* a klepněte na ni. Poté ve druhém sloupci klepněte na tlačítko se symbolem výpustky (tři tečky), nacházející se vpravo od názvu aktuálního použitého písma.
Otevře se dialog *Písmo (Font)*.
9. Písmo změňte na *Segoe UI*, řez písma změňte na *Polotučné (SemiBold)* a velikost písma upravte na 24. Poté klepněte na tlačítko *OK*.
Pokud měly popisky příliš malou výšku na zobrazení písma, jejich texty najednou zmizí. V takovém případě si upravte výšku popisků tak, abyste ve formuláři viděli v každém popisku alespoň písmeno *L*. Zjistíte, že toto písmeno bude zobrazeno již pomocí vybraného písma, řezu a velikosti.
Nyní vlastnosti *Text* všech tří vybraných popisků přiřadíme hodnotu 0 – což je docela dobrý „zástupný znak“ pro čísla, která budou později – během hry – v popiscích zobrazována. (Jelikož zobrazovaná čísla jsou generována programem, mohli byste také hodnotu vlastnosti *Text* vymazat, nicméně použití nějakého zástupného znaku vám dává reálnou představu o tom, jak velký by každý popisek měl být.)
10. Klepněte někde do prázdné části formuláře. Zrušíte tak skupinový výběr všech tří popisků. Poté klepněte na první popisek.
11. V okně Vlastnosti (Properties) poklepejte na vlastnost *Text* (ta se opět nachází v kategorii *Vzhled /Appearance/*), zadejte **0** a stiskněte klávesu *Enter*.
Text objektu *Label1* se změní na 0. V dalších částech této kapitoly budete tuto vlastnost pomocí kódu programu měnit tak, aby odpovídala aktuálně vygenerovanému náhodnému číslu.
12. Stejným způsobem změňte vlastnost *Text* i u druhého a třetího popisku.

13. Proveďte konečné úpravy mezer mezi jednotlivými popisky tak, aby mezery byly pokud možno stejné. Případně můžete ještě upravit i polohu jednotlivých popisků.
Po všech těchto úpravách by váš formulář měl zhruba odpovídat následující ukázce:



Nyní upravíme vlastnosti *Text*, *Font* a *ForeColor* čtvrtého popisku.

Nastavení vlastností popisku určeného pro zobrazení popisu formuláře

1. Ve formuláři klepněte na čtvrtý popisek (*Label4*).
2. Přejděte do okna Vlastnosti (Properties) a vlastnosti *Text* přiřadte hodnotu **Šťastná sedma**.
3. Klepněte na vlastnost *Font* a poté ve druhém sloupci klepněte na tlačítko se symbolem výpustky (tři tečky), nacházející se vpravo od názvu aktuálního použitého písma.
4. Pomocí dialogu Písmo (Font) změňte písmo na Segoe UI, řez písma změňte na Polotučné (SemiBold) a velikost písma upravte na 18. Poté klepněte na tlačítko OK.

Písmo objektu *Label4* se zaktualizuje. Současně se velikost tohoto objektu změní tak, aby bylo možné ve formuláři zobrazit celou hodnotu vlastnosti *Text* pomocí zvoleného písma. Toto vše proběhne proto, že vlastnost *AutoSize* má hodnotu *True*.

5. V okně Vlastnosti (Properties) klepněte na vlastnost *ForeColor*.
6. Zadejte **DarkRed** a poté stiskněte Enter.

Sada Visual Studio změní barvu popisku Šťastná sedma (objekt *Label4*) na tmavě červenou (139, 0, 0), což je stejná barva, kterou jste použili i při vývoji hry Šťastná sedma pro Windows Store v kapitole 3.

Nyní upravíme vlastnost *Text* samotného formuláře, a to tak, aby se přímo v záhlaví formuláře zobrazoval text „Šťastná sedma“.

Nastavení textu záhlaví formuláře

1. V okně Návrhář (Designer) klepněte na formulář, avšak nikoliv na kterýkoliv z jeho objektů.
2. V okně Vlastnosti (Properties) změňte vlastnost *Text* na **Šťastná sedma** a stiskněte Enter.
Váš formulář by nyní měl vypadat asi takto:



V další části můžeme přejít k úpravám objektu *PictureBox1*, což je poslední dosud neupravený objekt formuláře.

Vlastnosti okénka pro zobrazování obrázku

Pokud osoba hrající vaši hru trefí jackpot (tj. přinejmenším v jednom z popisků formuláře se objeví číslo 7), musí se v okénku pro zobrazování obrázku zobrazit fotografie (ve formátu .jpg) mincí z celého světa. Tato fotografie je součástí ukázkových souborů, které jsem pro vás připravil, nicméně vy ji můžete nahradit jakoukoliv jinou fotografií.

V jedné z předcházejících částí této kapitoly jste upravili vlastnost *SizeMode* objektu *PictureBox1* tak, aby bylo zajištěno zobrazení celé fotografie v tomto objektu. Dále jste vlastnosti *Image* přiřadili název toho souboru ve formátu JPEG, který chcete během hry zobrazit. Nyní tedy musíte upravit vlastnost *Visible*, určující stav obrázku v okamžiku spuštění aplikace. (Ne vždy je nutné či žádoucí, aby všechny objekty byly viditelné ihned po spuštění aplikace; v době běhu této hry budete v závislosti na zobrazených číslech obrázků skrývat nebo zobrazovat.)

Nastavení vlastnosti *Visible* u objektu typu *PictureBox*

1. Ve formuláři klepněte na objekt typu *PictureBox*.
2. V okně Vlastnosti (Properties) přejděte do kategorie Chování (Behavior) a klepněte na vlastnost *Visible*. Poté ve druhém sloupci klepněte na šipku, která se zobrazí vedle aktuálního nastavení této vlastnosti.

Rozbalí se seznam obsahující všechna platná nastavení vlastnosti *Visible*.

3. Vyberte hodnotu *False*. Díky tomuto nastavení nebude obrázek po spuštění programu zobrazený.

V této souvislosti je nutné zdůraznit, že přiřazení hodnoty *False* vlastnosti *Visible* ovlivňuje chování objektu typu *PictureBox* v době běhu programu, nikoliv však nyní, tj. v době návrhu.



Tip: U tzv. booleovských vlastností (tj. vlastností, u nichž lze nastavit pouze hodnotu *True* nebo *False*) je možné přecházet z jedné hodnoty na druhou také poklepáním na název příslušné vlastnosti. Přitom platí, že výchozí nastavení booleovských vlastností lze poznat podle normálního písma, zatímco změněné hodnoty jsou vždy zobrazeny tučně.

Zvýšení srozumitelnosti díky pojmenování objektů

V jedné z předcházejících částí této kapitoly jste objektům tlačítek přiřadili nějaké názvy, což jsem vám doporučil jako jeden z osvědčených programátorských postupů. Osobně vám doporučuji, abyste vždy pojmenovali všechny objekty, s nimiž hodláte v kódu programu pracovat jakýmkoliv způsobem, a zabránili tak vzniku případných chyb. Pro pojmenování objektů můžete využít vlastnost *Name*, kterou najdete v okně Vlastnosti (Properties). V následujícím cvičení přiřadíte názvy všem popiskům, v nichž budou v době běhu programu zobrazována vygenerovaná náhodná čísla. Dále přiřadíte název i objektu typu *PictureBox*, zobrazujícímu fotografii *Coins.jpg*.

Nastavení vlastnosti *Name*

1. Vraťte se zpět do formuláře a klepněte na objekt *Label1* (tj. popisek pro zobrazení prvního šťastného čísla). V okně Vlastnosti (Properties) pak vyhledejte vlastnost *Name* a zadejte do ní **PrvniCislo**.

Pokud chcete, aby vlastnost *Name* byla v okně Vlastnosti (Properties) uvedena poblíž jeho horního okraje, pak si vlastnosti zobrazte v abecedním pořadí.

2. Ve formuláři klepněte na objekt *Label2* (tj. popisek pro zobrazení druhého šťastného čísla). V okně Vlastnosti (Properties) pak vyhledejte vlastnost *Name* a zadejte do ní **DruheCislo**.
3. Ve formuláři klepněte na objekt *Label3* (tj. popisek pro zobrazení třetího šťastného čísla). V okně Vlastnosti (Properties) pak vyhledejte vlastnost *Name* a zadejte do ní **TretiCislo**.
4. Opět se vraťte do formuláře a tentokrát klepněte na objekt typu *PictureBox*. V okně Vlastnosti (Properties) pak vyhledejte vlastnost *Name* a zadejte do ní **ObrazekMinci**.
5. V tuto chvíli jsou všechny potřebné vlastnosti nastaveny. Pokud stále pracujete s plovoucím oknem Vlastnosti (Properties), stiskněte klávesu Ctrl a poklepáním na záhlaví tohoto okna zajistíte jeho návrat do ukotvené pozice.

Příprava kódu

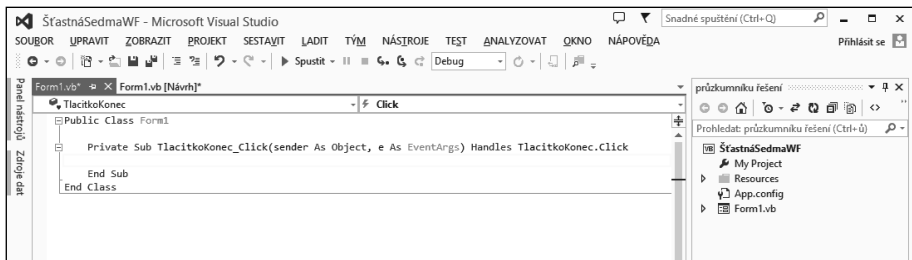
Nyní jste připraveni začít psát kód aplikace Šťastná sedma určené pro Windows desktop. V kapitole 3 jste se dozvěděli, že kód jazyka Visual Basic se zadává a upravuje v okně Editor kódu (Code Editor). Ačkoliv ovládací prvky, které jsme použili pro vytvoření uživatelského rozhraní této aplikace využívající technologii Windows Forms, se liší od těch, které jsme použili při vytváření podobné aplikace pro Windows Store, samotný kód programu je do značné míry podobný. To proto, že v obou případech je základem Visual Basic, který je pro obě platformy identický. Jediné rozdíly, na které narazíte, se budou týkat názvů objektů a jejich vlastností a v některých případech i tříd, s nimiž v platformě .NET Framework budete pracovat. Nicméně základní prvky jazyka Visual Basic budou v podstatě shodné.

Při zadávání kódu aplikace Šťastná sedma určené pro Windows desktop postupujte podle následujících kroků.

Použití okna Editor kódu (Code Editor)

1. Přejděte do formuláře a poklepejte na tlačítko Konec.

Uprostřed IDE sady Visual Studio se na další kartě zobrazí okno Editor kódu (Code Editor), jehož ukázkou vidíte dále:



Uvnitř tohoto okna vidíte příkazy programu související s aktuálním formulářem. Přitom lze říci, že příkazy programu jsou v aplikacích využívajících technologii Windows Forms, vždy seskupeny v jedné či více procedurách. V kódu jazyka Visual Basic se setkáte s celkem třemi typy procedur: s procedurami typu *Sub*, procedurami typu *Function* a procedurami typu *Property*.

Na obrazovce nyní vidíte proceduru *TlacitkoKonec_Click*, což je procedura typu *Sub*, deklarovaná proto, aby umožnila zpracování události, která vznikne ve chvíli, kdy uživatel v běžícím programu klepne na tlačítko Konec. Platí, že všechny procedury typu *Sub* začínají klíčovým slovem *Sub* a končí kombinací klíčových slov *End Sub*.

Ve vývojovém vzoru aplikací založených na technologii Windows Forms se podobné procedury říká také *zpracovatel událostí (event handler)*. Kód tohoto konkrétního zpracovatele bude proveden ve chvíli vzniku (či též spuštění) události *Click*. Je nutné zdůraznit, že tento zpracovatel nevrací volajícímu kódu žádnou návratovou hodnotu. (Jak uvidíte v dalších částech této knihy, procedury typu *Function* a *Property* velmi často nějaké hodnoty vrací.)

Jakmile jste v IDE sady Visual Studio poklepali na tlačítko Konec, Visual Studio automaticky připravilo první a poslední řádek kódu zpracovatele události *TlacitkoKonec_Click* a přiřadilo tyto řádky k události *Click*, což ostatně dokazuje i následující ukázka. V okně Editor kódu (Code Editor) vidíte možná ještě další kousky kódu (například slova jako *Public a Class*), které sada Visual Studio přidala proto, aby definovala důležité charakteristiky formuláře. Jejich plnému popisu se budu věnovat později.

```
Private Sub TlacitkoKonec_Click(sender As Object, e As EventArgs) Handles
TlacitkoKonec.Click
```

```
End Sub
```

Tělo procedury je nutné zadat mezi tyto dva řádky. Platí, že kód těla procedury bude proveden vždy, když uživatel spustí akci související se zpracovatelem události. V tomto případě se jedná o událost typu klepnutí, nicméně mohlo by se jednat o jakýkoliv jiný typ události.

2. Mezi tyto dva řádky napište **End** a stiskněte klávesu Enter.

Sada Visual Studio rozpozná slovo *End* jako jedno z jedinečných rezervovaných slov neboli *klíčových slov* a zobrazí jej v seznamu majícím karty Obecné (General) a Všechny (All). Společnost Microsoft nazývá tuto funkci interaktivní podpory *IntelliSense*, neboť se vám snaží inteligentním způsobem pomoci při psaní kódu. Kromě toho vám tato funkce umožňuje procházení abecedně seřazeného seznamu různých klíčových slov a objektů jazyka Visual Basic. (Díky tomu je možné se jazyk částečně naučit používáním samotného IDE.)

Po stisku klávesy Enter se barva písmen ve slově *End* změní na modrou. Současně vidíte, že slovo bude odsazeno. To vše znamená, že sada Visual Studio rozpoznala slovo *End* jako jedno z několika stovek jedinečných klíčových slov jazyka Visual Basic. Klíčové slovo *End* umožňuje zastavení běhu programu a uzavření jeho okna. V tomto případě je slovo *End* také úplným příkazem programu neboli samostatnou instrukcí, kterou následně rozpozná i kompilátor jazyka Visual Basic, což je ta část sady Visual Studio, která zpracovává neboli *parsuje* každý řádek zdrojového kódu v jazyce Visual Basic a výsledek kombinuje s dalšími prostředky s cílem sestavení spustitelného souboru.

V tuto chvíli máte dokončený kód tlačítka Konec. Můžeme tedy přejít k tlačítku Vytoč a napsat jeho kód. V jeho případě však platí, že kód bude o něco delší.

Vytvoření kódu tlačítka Vytoč

1. Poblíž horního okraje okna Editor kódu (Code Editor) klepněte na kartu Form1.vb [Návrh] (Form1.vb [Design]). Tímto způsobem se znovu vrátíte do okna s návrhem formuláře.

Je-li viditelné okno Editor kódu (Code Editor), není současně zobrazený formulář, na němž právě pracujete. Pomocí karet, které najdete na horním okraji okna Editor kódu (Code Editor) a okna Návrhář (Designer), je však velice snadné přecházet z jednoho okna do druhého a zase zpět.

2. Poklepejte na tlačítko Vytoč.

Objeví se okno Editor kódu (Code Editor), v němž bude připravený základ kódu zpracovatele událostí souvisejícího s tlačítkem Vytoč.

Ačkoliv jste text tohoto tlačítka změnili na „Vytoč“, nezapomeňte, že název tlačítka v programu je *TlacitkoVytoč*. Proto bude vždy, když uživatel klepne na toto tlačítko, proveden kód zpracovatele událostí *TlacitkoVytoč_Click*.

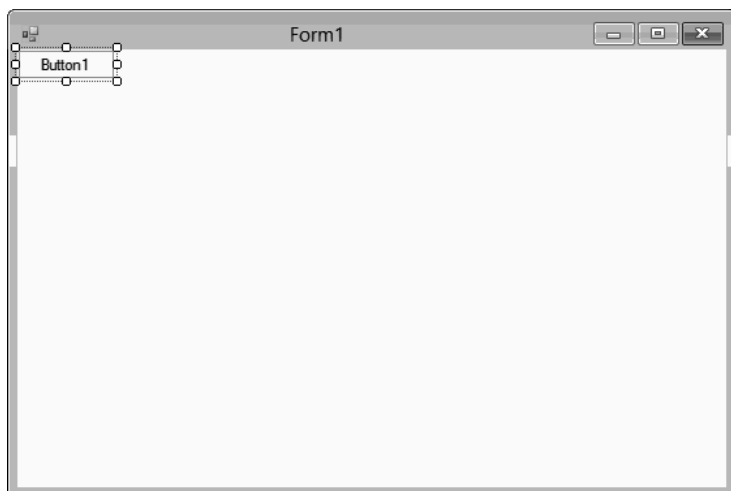
3. Mezi příkazy *Private Sub* a *End Sub* napište následující řádky. Po zadání každého řádku kódu stiskněte klávesu Enter a poté v případě potřeby klávesu Tab, kterou zajistíte odsazení dalšího řádku kódu. Všimněte si, že při zadávání jednotlivých řádků IDE kód formátuje a zobrazuje jeho různé části rozdílnými barvami, čímž vám usnadňuje rozpoznávání těchto částí.

```
Dim generator As New Random
ObrazekMinci.Visible = False
```

```
PrvniCislo.Text = generator.Next(0, 9)
DruheCislo.Text = generator.Next(0, 9)
TretiCislo.Text = generator.Next(0, 9)
```

```
If (PrvniCislo.Text = "7") Or (DruheCislo.Text = "7") Or
(TretiCislo.Text = "7") Then
    ObrazekMinci.Visible = True
    My.Computer.Audio.Play(My.Resources.ArcadeRiff, AudioPlayMode.Background)
End If
```

Po dokončení by okno Editor kódu (Code Editor) mělo vypadat asi tak jako na naší ukázce:



4. Z nabídky Soubor (File) zvolte Uložit vše (Save All) a uložte tak všechny dosud provedené změny programu.

Rozbor kódu zpracovatele události `TlacitkoVytoc_Click`

Kód zpracovatele událostí `TlacitkoVytoc_Click` je prováděn vždy, když uživatel ve formuláři klepne na tlačítko Vytoč. Klíčové slovo `Handles` v prvním řádku procedury `Sub` je tím příkazem, který propojuje tlačítko na formuláři se zpracovatelem událostí.

Lze říci, že tato procedura funguje prakticky stejně jako ten kód jazyka Visual Basic, který jsme v kapitole 3 použili ve hře Šťastná sedma navržené pro Windows Store. Námi navržený zpracovatel událostí v podstatě provádí tyto čtyři základní kroky:

1. Deklaruje generátor náhodných čísel, nazvaný *generator*.
2. Skrývá fotografii mincí.
3. Generuje tři náhodná čísla a zobrazuje je v objektech typu `popisek`.
4. Je-li jedno z vygenerovaných čísel rovno 7, zobrazuje fotografii mincí a přehrává zvuk.

Jak již víte, objekt generátoru náhodných čísel je inicializován následujícím příkazem:

```
Dim generator As New Random
```

Tento řádek kódu deklaruje generátor náhodných čísel, který lze použít k výpočtu jednociferných čísel neodpovídajících žádné číselné řadě či žádnému vzoru a spadajících do zadaného rozsahu. V dalších řádcích kódu procedury je pak používán identifikátor `generator`.

Skrýtí fotografie mincí je zajištěno následujícím řádkem kódu:

```
ObrazekMinci.Visible = False
```

Ačkoliv fotografie mincí není při spuštění programu zobrazena (v jednom z předcházejících cvičení jste změnilí vlastnost `Visible` tohoto obrázku, čímž jste zajistili popisovaný stav), tento řádek kódu řeší stav, kdy uživatel trefí jackpot a při další hře je nutné fotografii opět skrýt.

Následující tři řádky kódu zajišťují výpočty náhodných čísel. Objekt `generator` v každém z následujících řádků kódu vytváří náhodné číslo v rozmezí 0 až 9 – což je přesně to, co potřebujete pro tuto hru. Tyto řádky jsou zcela shodné s těmi, které jste použili i v kapitole 3.

```
PrvniCislo.Text = generator.Next(0, 9)
DruheCislo.Text = generator.Next(0, 9)
TretiCislo.Text = generator.Next(0, 9)
```

Poslední skupina příkazů programu pak provádí kontrolu, zda některé z náhodných čísel není rovno 7. Je-li jedno či více náhodných čísel rovno 7, program zobrazí grafické znázornění výhry a přehraje zvuk, jímž oznamuje výhru.

```
If (PrvniCislo.Text = "7") Or (DruheCislo.Text = "7") Or
  (TretiCislo.Text = "7") Then
  ObrazekMinci.Visible = True
  My.Computer.Audio.Play(My.Resources.ArcadeRiff, AudioPlayMode.Background)
End If
```

Jak vidíte, zvuk výhry je imitován zcela stejným zvukovým souborem typu .wav, který jsme použili i v kapitole 3. Avšak protože Panel nástrojů (Toolbox) technologie Windows Forms neobsahuje žádný ovládací prvek typu *MediaElement*, použil jsem k přehrání zvuku objekt *My.Computer.Audio*, který je standardní součástí platformy .NET Framework. Tento objekt nabízí metodu *Play*, díky níž jsem byl schopen zajistit přehrání zvukového souboru, uloženého ve složce *Resources* projektu. Vlastnost *AudioPlayMode.Background*, kterou jsem použil při volání metody *Play*, umožňuje přehrání zvuku v pozadí, což znamená, že program nečeká na dokončení přehrávání a pokračuje ve svém běhu.

Kdykoliv uživatel klepne na tlačítko Vytoč, je proveden (či též zavolán) kód zpracovatele události *TlacitkoVytoč_Click*. To znamená, že jsou znovu provedeny všechny příkazy tohoto kódu.

Spuštění aplikace Šťastná sedma

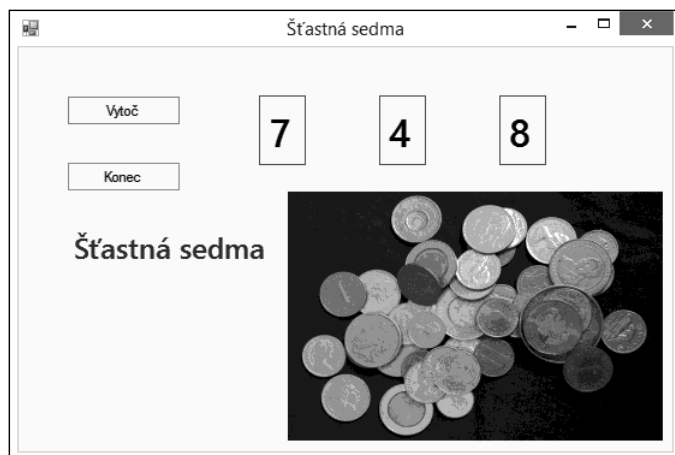
Skvělé – v tuto chvíli máte připravené vše, co potřebujete ke spuštění své nové hry pro Windows desktop. Jak jsme si řekli již v kapitole 2, aplikaci můžete spustit tak, že na panelu nástrojů Standardní (Standard) klepnete na tlačítko Spustit (Start).



Poznámka: Pokud jste projekt *MojeŠťastnáSedmaWF* nevytvářeli od počátku anebo chcete pouze porovnat to, co jste vytvořili, s kódem, který jsem pro vás připravil sám, zkuste si nyní otevřít a spustit dokončený projekt *ŠťastnáSedmaWF*, který najdete na pevném disku svého počítače ve složce *Dokumenty\Visual_Basic_KzK\Kapitola 04* (to je ostatně výchozí umístění pro všechny soubory související s touto kapitolou).

Spuštění programu ŠťastnáSedmaWF

1. Na panelu nástrojů Standardní (Standard) klepnete na tlačítko Spustit (Start). Sada Visual Studio zkompileje program *Šťastná sedma* a spustí jej. Po několika vteřinách se zobrazí uživatelské rozhraní v té podobě, v jaké jste jej navrhli.
2. Několikrát klepnete na tlačítko Vytoč. Pokuste se tak hru vícekrát vyhrát. Vždy, když klepnete na tlačítko Vytoč, program vypočítá další tři náhodná čísla a zobrazí je v popisích, které jsou součástí formuláře. Podaří-li se vám vyhrát, bude vaše obrazovka vypadat například takto:



Jelikož v prvním popisku je zobrazeno číslo 7, program zobrazil fotografii ilustrující výhru a současně přehrál zvuk napodobující zvuk výherního automatu. Vyhrali jste!

3. Klepněte na tlačítko Vytoč ještě alespoň desetkrát. Zkuste přitom sledovat vygenerovaná náhodná čísla, zobrazená v jednotlivých popiscích.

Přesně toto je okamžik, kdy byste měli program sledovat obzvláště pozorně. Funguje program správně? Je zvuk při každé výhře přehrán správně? Zmizí obrázek vždy, když po výhře znovu klepnete na tlačítko Vytoč? Vypadají čísla správně – jinými slovy řečeno, vypadají skutečně *náhodně*?

V tuto chvíli to vypadá tak, že v téměř polovině případů vyhráváte jackpot. Nicméně z čistě matematického hlediska vaše šance opravdu není takto velká. Skutečná pravděpodobnost výhry je totiž okolo 28 %, tj. 2,8x z 10 her.

4. Po skončení veškerého experimentování se svým nejnovějším „výtvořem“ klepněte na tlačítko Konec.

Běh aplikace pro Windows desktop bude ukončen a na vaší obrazovce se znovu objeví IDE sady Visual Studio.

Sestavení spustitelného souboru

Vášim posledním úkolem v této kapitole bude dokončení celého procesu vývoje a vytvoření konečné verze aplikace pro Windows desktop, určené pro váš počítač. Aplikace pro Windows desktop vytvořené pomocí sady Visual Studio mají příponu .exe a lze je spustit na jakémkoliv počítači, na němž je nainstalovaný operační systém Windows spolu se všemi nezbytnými podpůrnými soubory. Přitom platí, že jednou z kriticky důležitých knihoven, vyžadovaných aplikacemi vytvořenými v sadě Visual Studio, je platforma .NET Framework. Nicméně ta je standardně nainstalována prakticky na všech počítačích s operačními systémy Windows 8.1, Windows 8 či Windows 7.

V kapitole 3 jsme si řekli, že v tuto chvíli můžete pro svůj projekt vytvořit dva typy spustitelných souborů: sestavení pro ladění a sestavení pro uvolnění. Rozbalovací seznam Konfigurace řešení (Solution Configurations), který je součástí panelu nástrojů Standardní (Standard), vám říká, zda výsledný spustitelný soubor bude určen pro ladění anebo pro uvolnění.

Zkusme si nyní vytvořit sestavení pro uvolnění, které nazveme ŠťastnáSedmaWF.exe.

Vytvoření spustitelného souboru

1. Na panelu nástrojů Standardní (Standard) rozbalte seznam Konfigurace řešení (Solution Configurations) a vyberte volbu Release.

Tímto připravíte sadu Visual Studio k vytvoření sestavení pro uvolnění, které nebude obsahovat žádné dodatečné informace pro ladění. Sestavení bude uloženo do složky bin\Release vašeho projektu.

2. V nabídce Sestavit (Build) klepněte na volbu Sestavit ŠťastnáSedmaWF (Build ŠťastnáSedmaWF). (Je možné, že v případě svého projektu uvidíte na začátku názvu prefix „Moje“.)

Příkaz Sestavit (Build) zkompile zdrojový kód vašeho projektu. Výsledkem je spustitelný soubor vaší aplikace pro Windows desktop, nazvaný ŠťastnáSedmaWF.exe.

Nyní můžeme zkusit spustit tuto hru mimo prostředí sady Visual Studio, tj. přímo ze systému Windows. (Následující příkaz závisí na tom, jakou verzi systému Windows používáte.)

3. Pracujete-li se systémem Windows 8 či Windows 8.1, panelu Charms bar klepněte na Hledat, do pole pro hledání zadejte **spustit** a stisknutím klávesy Enter otevřete dialogové okno Spustit.

Máte-li na svém počítači systém Windows 7 či Windows Vista, do pole pro hledání zadejte **spustit** a stisknutím klávesy Enter otevřete dialogové okno Spustit.

4. Vpravo dole klepněte na tlačítko Procházet a poté přejděte do složky Dokumenty\Visual_Basic_KzK\Kapitola 04\MojeŠťastnáSedmaWF\bin\Release.

5. Klepněte na soubor MojeŠťastnáSedmaWF.exe. Tento výběr potvrďte klepnutím na tlačítko Otevřít. Poté klepněte na OK.

Program ŠťastnáSedmaWF se načte a spustí na ploše systému Windows.

Jelikož se jedná o jednoduchou testovací aplikaci, jejíž součástí není formální certifikát vydavatele, potvrzující spolehlivost či pravost programu, je možné, že se vám zobrazí hlášení tohoto typu: „Vydavatel nemohl být ověřen. Přejete si skutečně spustit tento program?“

V takovém případě klepnutím na tlačítko Ano potvrďte svůj úmysl spustit tento program. (Problematika vytváření zmíněných certifikátů již překračuje rámec této kapitoly, nicméně vy se můžete spolehnout na to, že tento program je vcelku bezpečný – pokud ovšem neohrozíte svoje zdraví tím, že tuto hru budete hrát příliš dlouho!)

6. Několikrát klepněte na tlačítko Vytoč a ověřte si tak funkčnost programu. Poté klepněte na tlačítko Konec.



Tip: Aplikace pro Windows desktop, mezi něž patří samozřejmě i programy zkompileované v sadě Visual Studio, můžete spouštět i tak, že pomocí Průzkumníka Windows vyhledáte příslušný soubor typu .exe a poklepete na něj.

Publikování aplikace pro Windows desktop

V kapitole 3 jste se dozvěděli, že aplikace pro Windows Store jsou navrženy pro online prodej a distribuci prostřednictvím Windows Store. Ačkoliv aplikace pro Windows desktop nemohou být v současné době tímto způsobem distribuovány, je vcelku jednoduché distribuovat neboli *publikovat* aplikaci pro Windows desktop pomocí technologie nasazení ClickOnce. Tato technologie seskupí soubory vašeho projektu do balíku, který lze umístit na webový server, do

sdílené složky, na lokální počítač (tj. na váš počítač či notebook) anebo na přenosné médium, jako například na CD-ROM či USB.

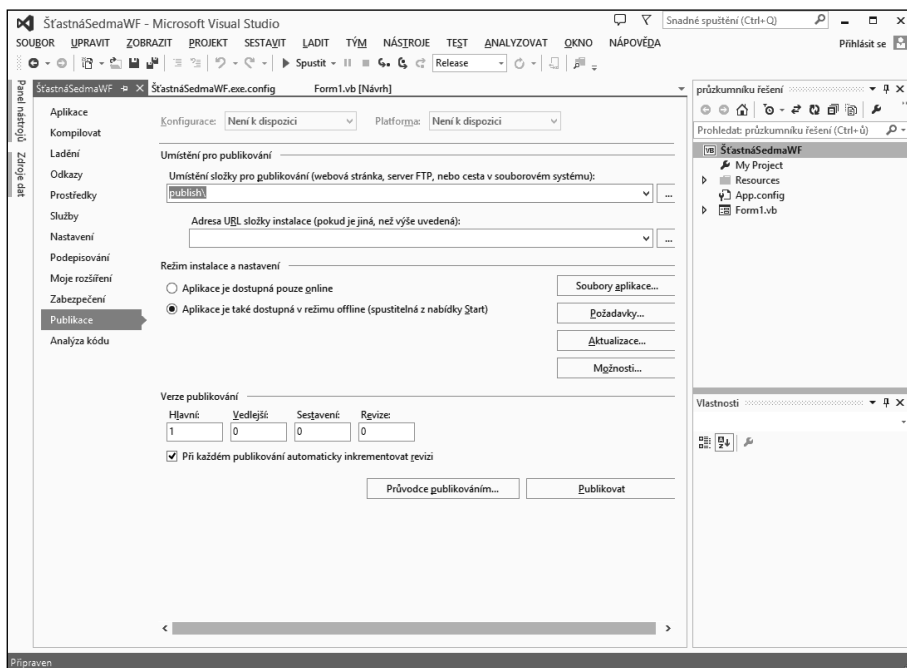
Chcete-li publikovat aplikaci pro Windows desktop, postupujte podle následujícího návodu. (Zkuste si nyní publikovat aplikaci *MojeŠťastnáSedmaWF* anebo si tyto kroky poznamenejte pro nějaké budoucí použití.)

Publikování aplikace pomocí technologie nasazení ClickOnce

1. V nabídce Projekt (Project) zvolte volbu Vlastnosti *Název projektu* (*Název projektu Properties*). V tomto případě se tedy může jednat o volbu Vlastnosti *ŠťastnáSedmaWF* (*ŠťastnáSedmaWF Properties*).

Otevře se okno *Návrhář vlastností projektu* (Project Properties Designer), obsahující mnoho karet a voleb týkajících se aplikace a jejich funkcí.

2. Přejděte na kartu *Publikace* (Publish), obsahující řadu voleb týkajících se nasazení aplikace. Ukázku těchto voleb vidíte níže:



3. Do textového pole *Umístění složky pro publikování* (Publishing Folder Location) zadejte název webového serveru, souborového serveru či cestu ke složce na vašem lokálním počítači, do níž má sada Visual Studio uložit vytvořený instalační balíček.



Poznámka: V této souvislosti je nutné zdůraznit, že se nejedná o konečné umístění nainstalované aplikace; jedná se o umístění instalačních souborů, které bude uživatel používat při instalaci vaší nové aplikace pro Windows desktop. To znamená, že do tohoto pole byste měli zadat cestu, v níž uživatelé naleznou potřebný instalační balíček. (Z tohoto důvodu je velmi rozumné uvažovat o webovém serveru.)

4. Pokud chcete složku, do níž má být instalační balíček uložen, vyhledat pomocí grafického rozhraní, pak můžete vedle tohoto pole klepnout na tlačítko s ikonou výpusky (tři tečky). Klepnete-li na tento velmi užitečný nástroj, otevře se dialog Otevřít web (Open Web Site), který vám umožňuje vyhledání složky v souborovém systému vašeho počítače, na webovém serveru služby IIS či na FTP serveru.
Případně můžete také klepnout na tlačítko Průvodce publikováním (Publish Wizard), čímž se spustí průvodce, který vás krok za krokem – pomocí řady otázek – provede celým procesem publikování. V průběhu tohoto průvodce budete moci říci, co a kde chcete publikovat.
5. Po zadání všech potřebných informací klepnete na tlačítko Publikovat (Publish Now), nacházející se ve spodní části stránky s vlastnostmi.
6. Sada Visual Studio nakopíruje vytvořený instalační balíček aplikace do toho umístění, které jste zadali. Uživatel pak bude schopen si aplikaci nainstalovat na svůj vlastní počítač.



Tip: Zajímají-li vás otázky zabezpečení týkající se nasazení aplikací pro Windows desktop, pak v okně Návrhář vlastností projektu (Project Properties Designer) přejděte na kartu Zabezpečení (Security). Zde najdete volby související s možnostmi zabezpečení technologie ClickOnce.

Gratuluji! Podařilo se vám vytvořit úplně novou aplikaci pro Windows desktop. Aplikaci jste napsali v jazyce Visual Basic a při její tvorbě jste využili technologii Windows Forms. Uložte všechny provedené změny a ukončete Visual Studio.

7. Změny uložte klepnutím na tlačítko Uložit vše (Save All), které najdete na panelu nástrojů Standardní (Standard).
8. Stránku vlastností projektu uzavřete klepnutím na křížek nacházející se v pravé části záložky této stránky.
9. V nabídce Soubor (File) zvolte Konec (Exit). Uzavřete tak jak projekt ŠťastnáSedmaWF, tak i samotnou sadu Visual Studio.

Tímto krokem se uzavře i celé vývojové prostředí sady Visual Studio.

Shrnutí

Tato kapitola názorně ukazuje, jakým způsobem je možné pomocí jazyka Visual Basic a technologie Windows Forms vytvořit aplikaci pro Windows desktop, nazvanou ŠťastnáSedmaWF. Jak vidíte, samotný proces vývoje se velice podobá tomu procesu, který jste použili v kapitole 3 při vývoji aplikace určené pro Windows Store. Zmíněná podobnost je ještě umocněna tím, že jste vytvářeli v podstatě shodnou aplikaci (tj. automatovou hru hledání šťastného čísla), a proto jste mohli poměrně snadno porovnat obě metody vývoje aplikací v sadě Visual Studio. Technologie Windows Forms nadále zůstává velice důležitou při psaní aplikací v jazyce Visual Basic, určených pro systémy Windows. Na trhu totiž existuje velké množství aplikací založených právě na technologii Windows Forms. A kromě toho, že tyto aplikace je nutné i nadále udržovat, je nutné uvážit i to, že Návrhář (Designer) i Panel nástrojů (Toolbox) technologie Windows Forms se snadno používají a zcela určitě dosahují takového stupně zralosti, s jakou se zatím nástroje pro tvorbu aplikací pro Windows Store či nástroje využívající technologii Windows Presentation Foundation (WPF) nemohou srovnávat.

V této kapitole jste se naučili pracovat s ovládacími prvky typu *Button*, *Label* a *PictureBox*. Součástí vaší práce byly i úpravy vlastností těchto prvků a použití původní grafiky. V další části jste se dozvěděli, jak se v okně Editor kódu (Code Editor) píše kód programu a jak se publikuje aplikace pomocí technologie ClickOnce. V kapitole 6 „Práce s ovládacími prvky pro Windows Forms“ pak najdete nejen podrobnější seznámení s ovládacími prvky Panelu nástrojů (Toolbox) technologie Windows Forms, ale i dalšími základními prvky aplikací pro Windows desktop, jako například nabídkami či panely nástrojů.

Základním omezením technologie Windows Forms je to, že její grafické procedury jsou o něco pomalejší než odpovídající procedury technologie WPF či aplikací pro Windows Store, a dále to, že technologie Windows Forms nenabízí plnou podporu přepracovaných operačních systémů Windows 8 a Windows 8.1. Přitom základní výhody těchto platforem spočívají v jejich zabezpečení, spolehlivosti, širší podpoře dotykových zařízení a samozřejmě i novém uživatelském rozhraní. Zjednodušeně řečeno, nacházíme se v období přechodu a profesionálové vyvíjející aplikace v jazyce Visual Basic musí získat dovednosti odpovídající současnému prostředí. Což ovšem znamená, že je nutné zvládat více vývojových platforem a je nutné hledat kreativní a efektivní způsoby migrace kódu z jednoho modelu do druhého.

Naštěstí sada Visual Studio tuto dynamickou výměnu do značné míry usnadňuje. Ostatně ze cvičení v kapitole 3 a v této kapitole zcela jasně vyplývá, že je možné vytvořit dvě různá uživatelská rozhraní, využívající rozdílné nástroje, a přesto v kódu jazyka Visual Basic udržovat prakticky stejnou aplikační logiku. Tuto podobnost budu zdůrazňovat i v dalších částech této knihy, kdy navíc paralelu jazyka Visual Basic rozšířím o webové (HTML) aplikace a aplikace pro Windows Phone.

V následující kapitole se vrátíme k vývoji aplikací pro Windows Store a budeme se podrobně zabývat některými základními ovládacími prvky Panelu nástrojů (Toolbox) pro Windows Store.

Návrh uživatelského rozhraní

V této části:

- Kapitola 5: Práce s ovládacími prvky aplikací pro Windows Store
- Kapitola 6: Práce s ovládacími prvky pro Windows Forms
- Kapitola 7: Značovací jazyk XAML krok za krokem
- Kapitola 8: Používání stylů XAML
- Kapitola 9: Seznámení se základními prvky návrhu pro Windows 8.1: řádek nabídek, informační rámeček, dlaždice a dotyk
- Kapitola 10: Vytváření konzolových aplikací

Toto je pouze náhled elektronické knihy. Zakoupení její plné verze je možné v elektronickém obchodě společnosti eReading.