

Microsoft

Visual Studio .NET

praktické programování
krok za krokem

František Šíma, David Vilímek

- Vytvoření nové aplikace
- Návrh databáze
- DataGrid a validace
- Propojení tabulek
- Příprava počítače
- Slovník pojmů



Upozornění pro čtenáře a uživatele této knihy

Všechna práva vyhrazena. Žádná část této tištěné či elektronické knihy nesmí být reprodukována a šířena v papírové, elektronické či jiné podobě bez předchozího písemného souhlasu nakladatele. Neoprávněné užití této knihy bude **trestně stíháno**.

Používání elektronické verze knihy je umožněno jen osobě, která ji legálně nabyla a jen pro její osobní a vnitřní potřeby v rozsahu stanoveném autorským zákonem. Elektronická kniha je datový soubor, který lze užívat pouze v takové formě, v jaké jej lze stáhnout s portálu. Jakékoliv neoprávněné užití elektronické knihy nebo její části, spočívající např. v kopírování, úpravách, prodeji, pronajímání, půjčování, sdělování veřejnosti nebo jakémkoliv druhu obchodování nebo neobchodního šíření je zakázáno! Zejména je zakázána jakákoliv konverze datového souboru nebo extrakce části nebo celého textu, umístování textu na servery, ze kterých je možno tento soubor dále stahovat, přitom není rozhodující, kdo takovéto sdílení umožnil. Je zakázáno sdělování údajů o uživatelském účtu jiným osobám, zasahování do technických prostředků, které chrání elektronickou knihu, případně omezují rozsah jejího užití. Uživatel také není oprávněn jakkoliv testovat, zkoušet či obcházet technické zabezpečení elektronické knihy.





Copyright © Grada Publishing, a.s.



Copyright © Grada Publishing, a.s.

Obsah

Úvod.....	9
1. Nová aplikace (projekt).....	13
2. První funkční aplikace – volání stránek	23
3. První databázová aplikace.....	33
3.1 Definování databáze	34
3.2 Vytvoření první tabulky (tabulka ORGanizace)	36
3.3 Plnění tabulky vzorkem dat	40
3.4 Navigace na stránkách .aspx	45
3.5 Aktualizace na stránkách .aspx	63
3.5.1 SQL procedury	63
3.5.2 Přidání prvků na stránku	67
3.5.3 Volání SQL procedur z Visual Studia	69
Založení nového řádku (volání procedury pORG_Insert).....	69
Oprava řádku (volání procedury pORG_Update).....	74
Metoda Page_Load	80
Metoda ButtonOprava_Click.....	81
Metoda ListBoxORG_SelectedIndexChanged	81
Rušení řádku (volání procedury pORG_Delete).....	81
3.6 Vylepšování programu.....	83
3.7 Přidání filtrů do navigačního seznamu.....	88
3.8 Import – export dat	93
4. Návrh databáze	99
4.1 Definice tabulek	99
4.1.1 Tabulka FAKtury	101
4.1.2 Tabulka DPH	103
4.1.3 Tabulka RADky	104
4.1.4 Tabulka PARametry	106

4.2 Definice vazeb	107
4.3 Vytvoření vzorku dat	111
5. DataGrid a validace (číselník DPH)	115
6. Tabulka parametrů	133
7. Propojení tabulek zdola nahoru (zaevidování faktury).....	139
7.1 Ovládání transakcí v SQL procedurách	142
7.2 Ovládání transakcí ve Visual Studiu	148
8. Propojení tabulek shora dolů (rozúčtování faktury)	155
9. Předávání parametrů mezi stránkami	175
10. Tisky	179
10.1 HTML	180
10.2 SqlDataReader (tisk faktur v intervalu).....	183
10.3 DataView (výpis DPH)	188
10.4 Odstránkování (styly)	195
11. Příprava počítače	197
11.1 Instalace MS SQL 2000	197
11.1.1 Vlastní instalace MSDE.....	198
11.1.2 Vlastní instalace MS SQL 2000 DE	199
11.2 Instalace IIS.....	204
11.2.1 Vlastní instalace IIS.....	204
11.3 Instalace .NET Framework	207
11.4 Instalace Visual Studia .NET (VS.NET)	208
11.4.1 Vlastní instalace VS.NET	208

12. Instalace webové aplikace u zákazníka	221
12.1 Základní administrace IIS.....	221
12.2 Základní administrace MS SQL 2000	228
Příloha: Slovník pojmů.....	249
Rejstřík	251

Úvod

Začínáte-li pracovat s nějakým produktem, musíte předem vědět, zda je vhodný pro cíl, kterého chcete dosáhnout. **Co** všechno od něj můžete očekávat.

Vyhovuje-li nabízený produkt vašim představám, začne vás zajímat způsob jeho použití. Potřebujete znát, **jak** s tímto nástrojem zacházet, aby se váš záměr povedl. Během času budete zlepšovat svoje dovednosti v dané oblasti, a pak mnohé z vás napadne seznámit se s používaným produktem hlouběji. Budete hledat odpovědi na otázky začínající slovem **proč**.

Tento úvod, spolu s kapitolou 4. Návrh databáze, by vám měl sdělit, **co** produkt umožňuje. Zbytek publikace je zaměřen hlavně na otázku, **jak** toho dosáhnout. Zájemci v něm najdou i odpovědi na svá **proč**.

Zaměřili jsme se na vytváření databázových aplikací v prostředí tenkého klienta. Tato část totiž bývá v různých knihách zmiňována pouze okrajově a programátor, který se z nejrůznějších důvodů rozhodne vytvářet aplikace v tomto prostředí, je nucen naučit se i klienta tlustého.

Aplikace vytvářené pomocí programu Visual Studio jsou typu klient-server. Server je počítač, na kterém je uložena databáze a který představuje datovou vrstvu. Jeho jediným úkolem je efektivní přístup k datům uloženým v databázi. Klient je počítač, který obsahuje vlastní aplikaci (předává příkazy pro server) a který, v případě tlustého klienta, představuje aplikační vrstvu.

Tenký klient je další, třetí vrstvou. Došlo totiž k rozdělení aplikační vrstvy na funkční a prezentační. Jedná se o vizuální část s minimální funkcionalitou omezenou na rychlé ošetření vstupů a na vizualizaci dat. Vlastní funkcionalita je na aplikačním serveru a její část může být až na databázovém serveru. Jak je vidět, pojem tenkého klienta je úzce spjat s trojúrovňovou architekturou. V našem konkrétním případě se jedná o webový prohlížeč na straně tenkého klienta.

Při práci s tenkým klientem jste omezeni možnostmi webového prohlížeče, tj. možnostmi HTML. Ten neumožňuje využití všech ovládacích prvků Windows. Jaký smysl má vlastně použití tenkého klienta, když jeho možnosti z pohledu uživatele jsou menší? Hlavní výhoda spočívá v údržbě a v instalaci u uživatelů. U klientských počítačů stačí, když mají nainstalován webový prohlížeč. Instalovat aplikaci tedy znamená, že uživatel obdrží pouze webovou adresu stránky, ze které se má aplikace vyvolat, a ta je tím nainstalována. Údržbu aplikace pak stačí provádět pouze na aplikačním serveru. Nároky na počítače u uživatelů mohou být též menší.

V této publikaci budeme vytvářet typickou databázovou aplikaci, která by vás v průběhu její tvorby měla seznámit se všemi podstatnými částmi týkajícími se Visual Studia při hromadném zpracování dat. Po jejím přečtení byste měli být schopni vytvářet databázové aplikace, pracující s několika vzájemně provázanými tabulkami.

I když se budeme snažit své programy neustále zdokonalovat podle toho, jak porostou naše znalosti, neznámá to, že následující program bude vždy zahrnovat všechny dosud získané poznatky. Na stránce **DPH.aspx** se například seznámíme se způsobem, jak ověřovat data dříve, než se je pokusíme zapsat do databáze. V následujících programech se k nim ale už nevrátíme. Chceme totiž aby program zůstal přehledný a aby obsahoval pouze probíranou funkčnost. Při psaní skutečné aplikace byste ovšem měli data ověřovat vždy.

Simulovali jsme situaci, kdy na začátku je pouze vhodný počítač a na konci je aplikace „Fakturace“, kterou jste schopni distribuovat. Aby tvrzení, obsažená v této knize, odpovídala realitě, nainstalovali jsme na prázdný počítač program Visual Studio .NET a začali jsme vytvářet ukázkovou aplikaci, která by vás měla průběžně seznamovat se všemi podstatnými částmi tohoto produktu, jež jsou nutné při vyvíjení robustních databázových aplikací. Současně vznikala tato publikace. Zaznamenávali jsme do ní všechny situace, se kterými jsme se v průběhu vývoje naší úlohy střetli, i když se netýkaly přímo Visual Studia .NET. V případě, že bylo třeba rozhodnout se mezi efektivitou pro uživatele a jednoduchostí z hlediska programátora, volili jsme jednoduchost. Protože jednoduchost je spojena s vizuálním programováním, dáváme přednost i vizuálnímu programování. Toto slovo je ostatně obsaženo i v názvu produktu. Efektivní programy mají tu nevýhodu, že jsou náročné na údržbu.

Jednotlivé příkazy, prvky apod. jsme uváděli tak, aby nebyly použity samostatně, ale v kontextu s vytvářenou aplikací.

Vzorová databáze „Fakturace“ obsahuje pouze 5 tabulek (z nichž jedna představuje tabulku parametrů), na kterých jsme se snažili ukázat základní vazby, se kterými byste měli uspět i při vytváření vazeb složitějších.

Ukázková aplikace provádí evidenci došlých faktur s vazbou na číselník organizací. U faktur evidujeme i řádky, které mají vazbu na číselník přijatých zdanitelných plnění. Faktury jsou

číslovány vzestupnou řadou s tím, že poslední číslo faktury udržujeme v parametrickém souboru. Mezi fakturami a číselníkem organizací je použita neidentifikační vazba, mezi fakturami a řádky vazba identifikační. Tato aplikace by mohla sloužit jako evidence DPH nebo jako vstup do účetnictví.

V první části publikace se ještě nepředpokládá, že jste seznámeni s logikou práce ve Visual Studiu. Proto jsme zvolili strategii „přilepování“ funkčnosti. Výsledný program vznikne po několika průchodech programováním. Při prvním průchodu se spokojíme s pouhým překladem zatím nefunkčního programu, v dalším pak se zobrazením dat a po dalším rozvíjení programu se dostaneme ke konečnému kroku s hotovou funkční aplikací pro aktualizaci. Vzhledem k tomu, že jednu a tutéž věc lze mnohdy řešit několika způsoby, snažíme se např. aktualizaci provádět pokaždé jinak. Při tvorbě profesionální aplikace by měly mít všechny programy stejnou formu.

V další části, kdy již předpokládáme, že základy práce ovládáte, budeme postupovat přesně obráceně. Ukážeme si výsledný program a potom rozebereme jeho jednotlivé části.

Publikace je určena pro všechny programátory, kteří již pracovali s nějakým jazykem (včetně programování v DOS) nebo alespoň tuší co to je podmínka, cyklus či sekvence, kteří mají představu co je SQL a kteří zběžně ovládají Windows. Po přečtení byste měli být schopni napsat robustní databázovou aplikaci s využitím transakcí.

Vzhledem k tomu, že tato kniha má formu učebnice, čtete ji souvisle tak jak je napsána. V každé kapitole se totiž seznámíte s určitými pojmy, znalostmi a postupy, na něž pak budeme navazovat. Vysvětlíme-li si například postup jak „Nakonfigurovat SqlDataAdapter“, nebudeme tento postup znova vysvětlovat, pouze uvedeme „Nakonfigurujte SqlDataAdapter“. Na konci každé kapitoly se budeme snažit vás na nové pojmy, znalosti a postupy upozornit.

Doporučujeme též, abyste si po každé části zkusili samostatně vytvořit svůj vlastní příklad. Naučíme-li se například jak vytvořit tabulku, zkuste si podle uvedeného postupu zhotovit svojí vlastní tabulku.

1.

Nová aplikace (projekt)

V této kapitole se naučíme, jak vytvořit novou aplikaci, přeložit ji a vzápětí vyvolat. Vzhledem k tomu, že tato publikace je zaměřena na tenkého klienta, bude naše aplikace prezentována webovou stránkou. Její snadné rutinní volání umožníme prostřednictvím programu Microsoft Internet Explorer tím, že ji zahrneme mezi oblíbené položky. I když aplikace v této kapitole nebude mít zatím žádnou funkčnost, uvedený postup využijeme i při vytváření složitějších aplikací. V prostředí Visual Studia se aplikace nazývá projekt, a proto budeme používat i toto označení. Pojem aplikace budeme používat v uživatelském smyslu, pojem projekt ve smyslu vývojářském.

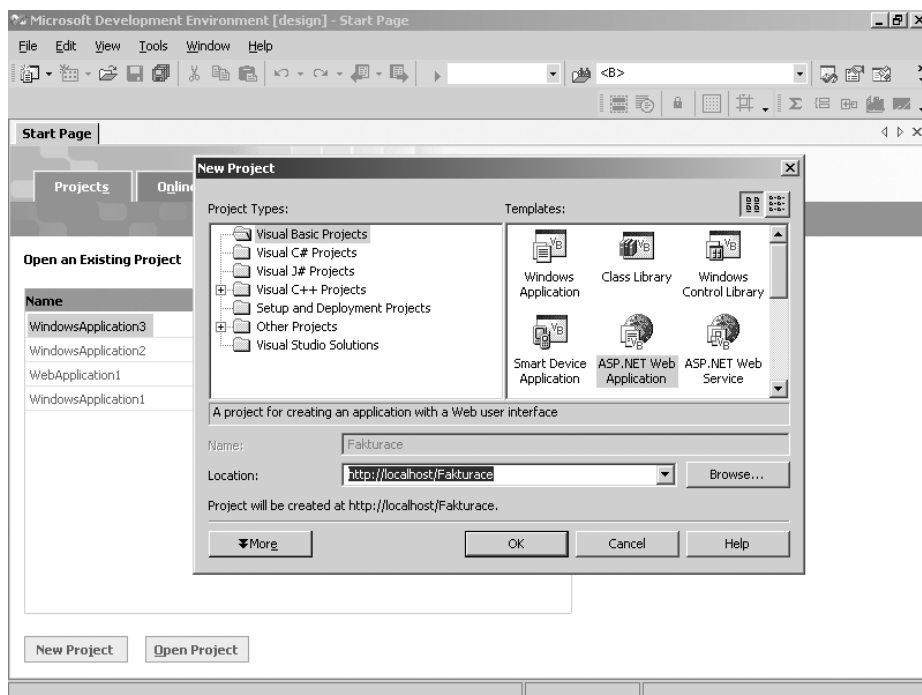
Projekty ve Visual Studiu mohou být různých typů. V následujících kapitolách jsme vybrali jako typ projektu Visual Basic Projects, neboť se nám zdál nejsrozumitelnější pro všechny ty, kteří mají zkušenost s nějakým jiným jazykem. Programátoři C si pravděpodobně vyberou typ projektu Visual C#.

Po vyvolání Visual Studia se zobrazí posledně použité 4 projekty (jestliže existují). Programátor má možnost vyvolat již existující projekt nebo vytvořit projekt nový. Protože právě začínáme a budeme vytvářet nový projekt, stiskneme tlačítko **New Project**. Dále, po stisknutí tlačítka **OK**, se zobrazí dialogové okno pro výběr již zmíněného typu

projektu (viz obrázek 1.1). V závislosti na typu projektu lze vybrat i typ zobrazení (šablonu), kterým se budou prezentovat data na obrazovce – **Template**. Protože vytváříme projekt ve Visual Basicu, vybereme **Visual Basic Projects** a pro již zmíněného tenkého klienta vybereme následně aplikaci **ASP.NET Web Application**. V rozvíracím seznamu je zobrazen navrhovaný název projektu spolu s umístěním. Obsluha má možnost tento název i umístění změnit. Rozhodli jsme se změnit pouze název. Protože vytváříme aplikaci pro evidenci přijatých faktur, přepsali jsme navrhovaný název `http://localhost/WebApplication1` na `http://localhost/fakturace`.



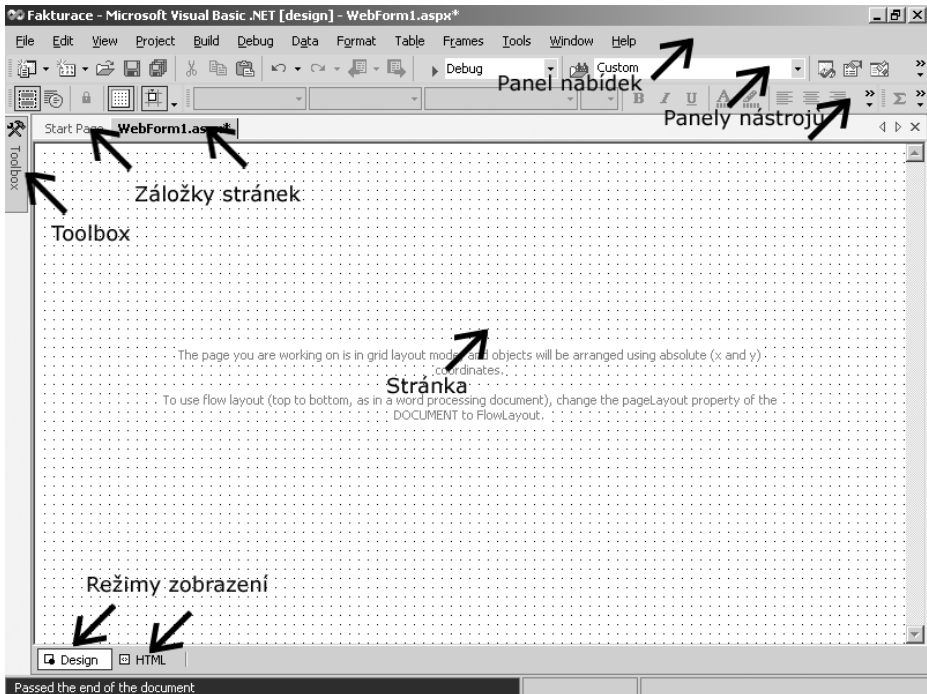
*Označení projekt je v tomto případě poněkud nepřesné. Nejedná se totiž o projekt, ale o řešení (solution). Obecně může totiž řešení obsahovat více projektů, i když v případě této publikace bude obsahovat vždy pouze jediný. Toto „nepřesné“ označení jsme použili proto, že je použito na úvodní obrazovce ve Visual Studiu. Jak ukazuje obrázek 1.1, používají se termíny **New Project**, **Open Project**, **Open an Existing Project**, ale ve skutečnosti se jedná o řešení.*



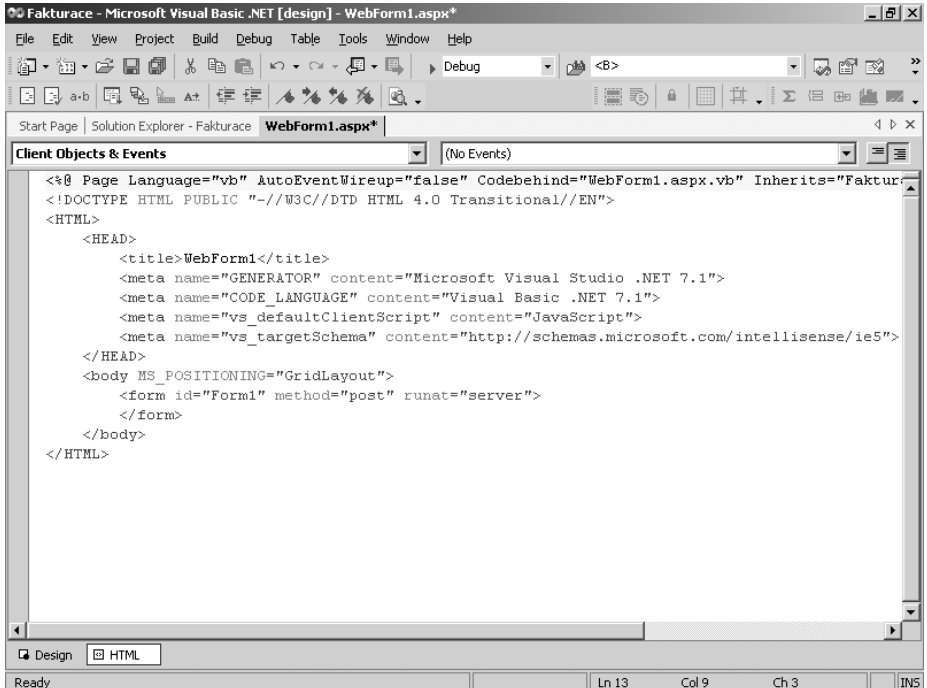
Obr. 1.1: Výběr typu projektu

Po stisknutí tlačítka **OK** se zobrazí pracovní plocha pro návrh první webové stránky (viz obrázek 1.2). V horní části (tak jako u všech produktů Microsoft) jsou pod **panelem nabídek (MenuBar)** umístěny **panely nástrojů (Toolbars)**. Pod panely nástrojů jsou umístěny záložky jednotlivých stránek (v našem případě zatím pouze již zmiňovaná pracovní plocha navrhované webové stránky s názvem **WebForm1.aspx** a úvodní stránka Visual Studia **Start Page**). Po levé straně je minimalizované okno **Toolbox** a ve spodní části je záložka s dotazem na způsob zobrazení pracovní plochy stránky. Pracovní plocha každé navrhované webové stránky může být totiž zobrazena ve dvou režimech (modech), a to buď v režimu **Design**, nebo v režimu **HTML**. Změny provedené na jedné záložce se

1. Nová aplikace (projekt)



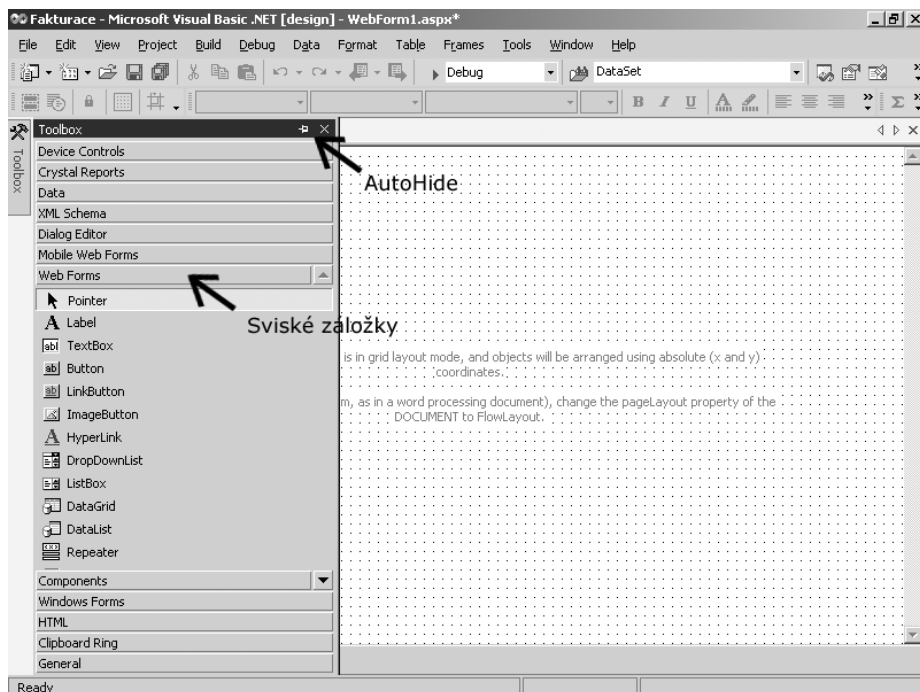
Obr. 1.2: Pracovní plocha pro návrh webové stránky v režimu Design



Obr. 1.3: Pracovní plocha pro návrh webové stránky v režimu HTML

okamžitě promítají do druhé záložky a vy si můžete vybrat, ve kterém režimu si přejete pracovat. My budeme pracovat v návrhovém režimu (**Design**), který zobrazuje pracovní plochu v takové podobě, jaká bude po vyvolání stránky uživatelem, nicméně pro ilustraci si zobrazíme i režim **HTML** (viz obrázek 1.3). Měl-li by vás zobrazený režim **HTML** odradit od dalšího čtení, nemusíte se ničeho bát. Můžete se bez něj obejít. Zvláště nyní, když s Visual Studiem teprve začínáme.

Ukážeme-li myší nad již zmíněný **Toolbox**, „vyjede“ tento do strany, zvětší se a zobrazí se nám jeho nabídka (viz obrázek 1.4). Pomocí okna **Toolbox** můžeme umístit na pracovní plochu různé prvky, jako jsou například pole pro vstup údajů **TextBox**, políčka pro zaškrtnutí **CheckBox** sloužící pro zadávání voleb, seznamy **ListBox**, přepínače **RadioButton** pro alternativní zadávání voleb apod. Komu nevyhovuje tento způsob práce s oknem **Toolbox**, může okno **Toolbox** přetáhnout na pracovní plochu, na které bude otevřené stále. Pomocí myši lze upravit jeho velikost a umístění.

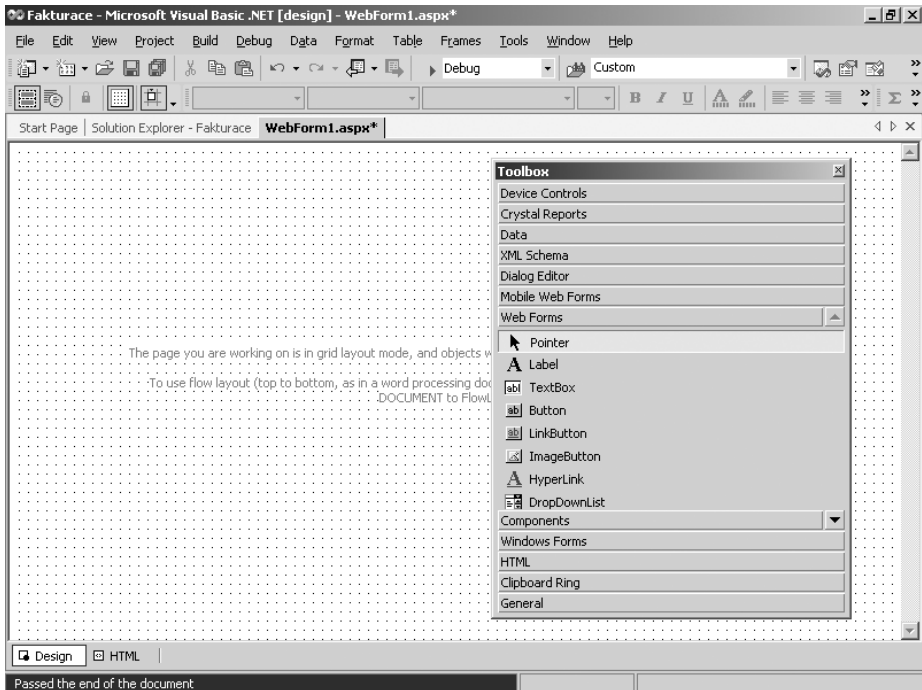


Obr. 1.4: Okno Toolbox na pracovní ploše v režimu AutoHide

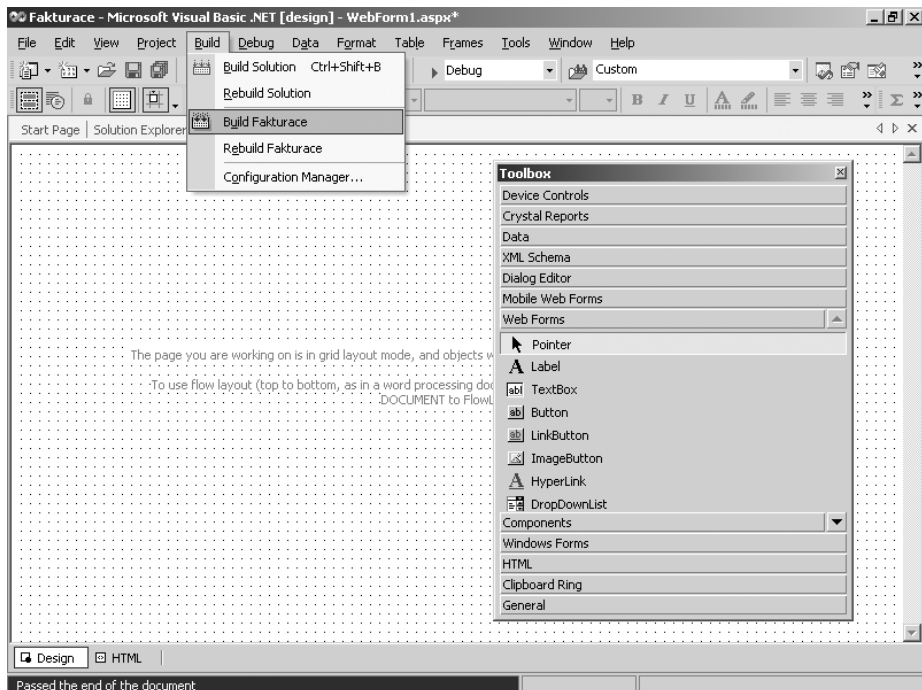
Okno **Toolbox** přetáhnete na plochu tak, že po jeho „vyjetí“ stisknete tlačítko **AutoHide**, které se nachází vedle křížku v záhlaví okna **Toolbox**, a pak jej uchopením přetáhnete na zvolené místo (viz obrázek 1.5). Tím se dostane do režimu **Float**. V záhlaví tohoto okna můžete nastavit i další vlastnosti. Podaří-li se vám okno **Toolbox** zavřít nebo pokud se vám na pracovní ploše neobjeví, vyvolejte ho zadáním příkazu **View** → **Toolbox**.

Projekt přeložíme a spustíme, i když zatím ještě nic neumí. Překlad provedeme tak, že klepneme na příkaz **Build** (pátá volba) a potom na libovolnou 1. až 4. položku z nabídky. My jsme jej vyvolali příkazem **Build** → **Build Fakturace** (viz obrázek 1.6).

1. Nová aplikace (projekt)



Obr. 1.5: Okno Toolbox na pracovní ploše v režimu Float



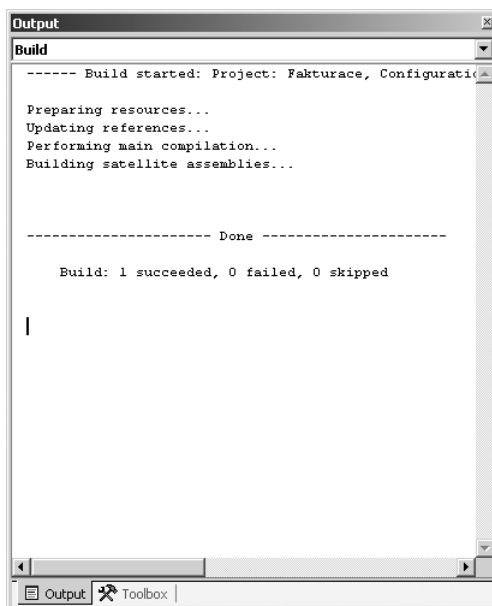
Obr. 1.6: Překlad projektu

1. Nová aplikace (projekt)



Při překladu se projekt uloží i v tom případě, že obsahuje chyby. Neukončíte-li Visual Studio, lze použít tlačítek **Undo/Redo**.

V okně **Output** (viz obrázek 1.7), které se při vyvolání překladu objeví, se zobrazí průběh a výsledek překladu. (Všimněte si, že se k oknu **Toolbox** automaticky vytvořila nová záložka.) Bude-li projekt obsahovat nějaké syntaktické chyby, zobrazí se jejich seznam právě v tomto okně. Máte-li okna **Toolbox** a **Output** umístěna vzájemně nezávisle, uchopte jedno okno pomocí myši a pokuste se ho umístit nad druhé okno. Při troše štěstí se vám hned napoprvé podaří obě okna sloučit a pro jejich vyvolávání se vytvoří záložky. Sloučí-li se vám jinak, okno opět vytáhněte stranou, upravte jeho velikost a zkuste to znovu. Podobným způsobem můžete nastavit zobrazení i dalších oken sloužících vývojáři, které je možné vyvolávat z panelu nabídek a se kterými se později setkáme. Nepodaří-li se vám požadované okno umístit do záložky, zkontrolujte zda nemá ve vlastnostech nastaveno **Floating**. Je-li tomu tak, zaškrtněte volbu **Dockable**. Vlastnost okna nastavíte tím, že umístíte kurzor do záhlaví okna a stisknete pravé tlačítko.



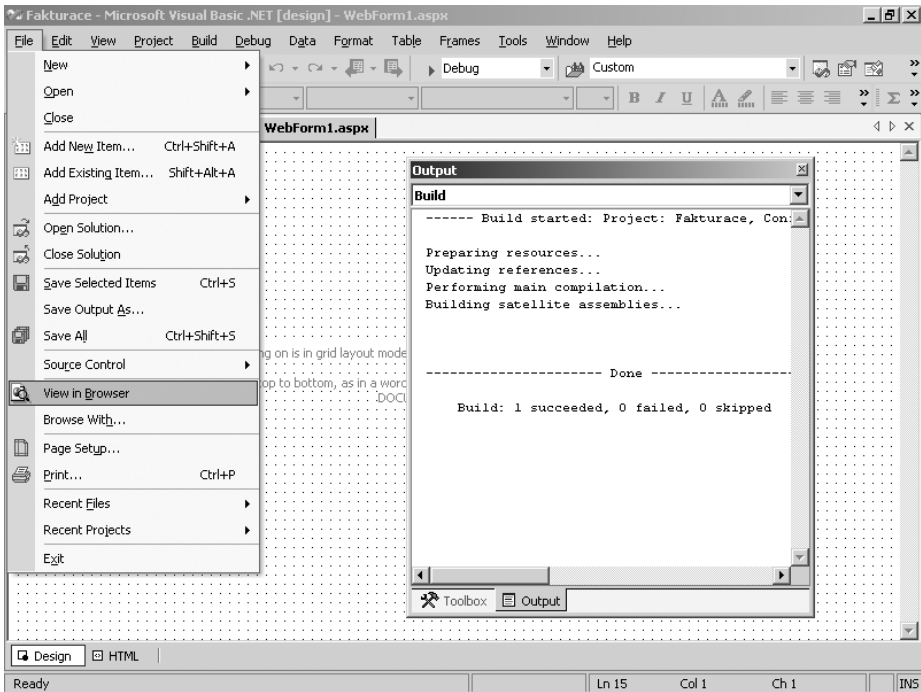
Obr. 1.7: Okno Output

Jak ukazuje obrázek 1.7, jsou okna **Toolbox** i **Output** sloučena. Obě se dají zavřít klepnutím na křížek v pravém horním rohu. Potřebujete-li zavřít pouze jedno z nich, klepněte na záložku tohoto okna, aby se stalo aktivní, uchopte toto okno a potáhněte jej stranou. Okno by nyní mělo být na stránce umístěno samostatně. Poté ho můžete zavřít. Později, až se seznámíme s dalšími okny (v panelu nabídek pod volbou **View**), budete postupovat obdobně.

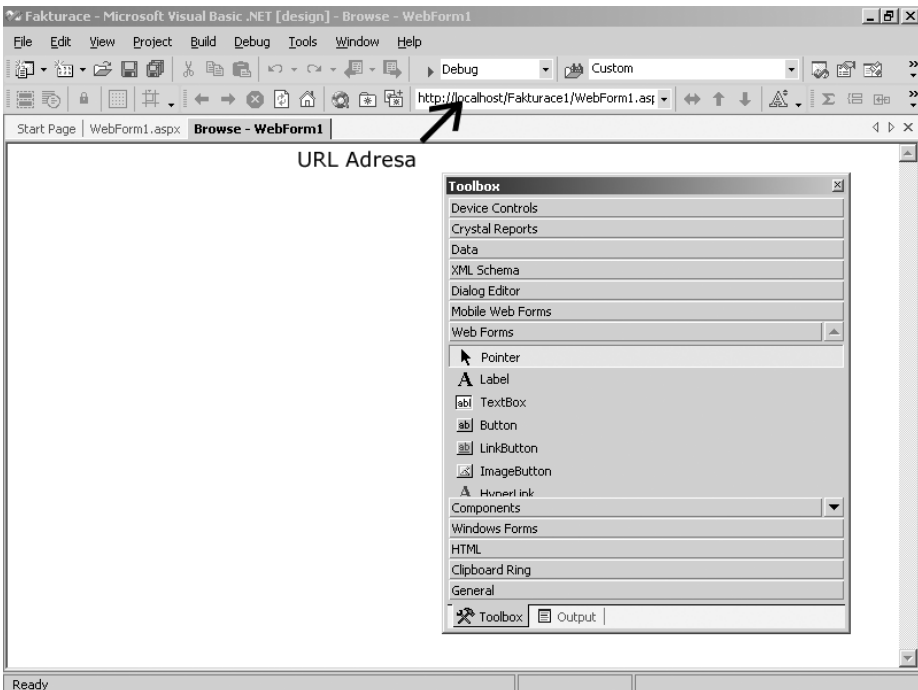
Po úspěšném překladu můžete projekt vyvolat. To provedete příkazem **File** → **View in Browser** (viz obrázek 1.8). Ve skutečnosti můžete předcházející krok (tj. **Build**) přeskóčit a vyvolat přímo **View in Browser**. Překlad se vyvolá sám. Měla by se vám zobrazit následující stránka (viz obrázek 1.9). Naše aplikace sice zatím nic neprovádí, ale to, co je nejdůležitější (překlad a vyvolání stránky), již umíme.

Jestliže jste nezavřeli okna **Toolbox** nebo **Output**, budou v případě spuštění tímto způsobem též zobrazena. Kromě této stránky jsou vidět i záložky všech otevřených stránek ve Visual Studiu. V našem případě zatím jen záložky stránek **Start Page** a **WebForm1.aspx**. Zapamatujte-li si adresu webové stránky (URL adresu), která by měla být vidět v rozvíracím seznamu ve spodní řadě panelu nástrojů, nebo ji stiskem klávesové zkratky **CTRL+C** dáte do paměti, můžete tuto webovou stránku přidat do oblíbených položek a volat ji přímo z Internet Exploreru. Zavřete tedy stránku nebo celé Visual Studio (podle toho, který křížek stisknete), vyvolejte Internet Explorer, jako adresu napište URL adresu (nebo ji vložte pomocí stisku klávesové zkratky **CTRL+V**). V našem případě se jedná o adresu `http://localhost/Fakturace/WebForm1.aspx`. Po stisknutí klávesy **ENTER** se zobrazí následující stránka. Přidejte ji do oblíbených položek (viz obrázek 1.10).

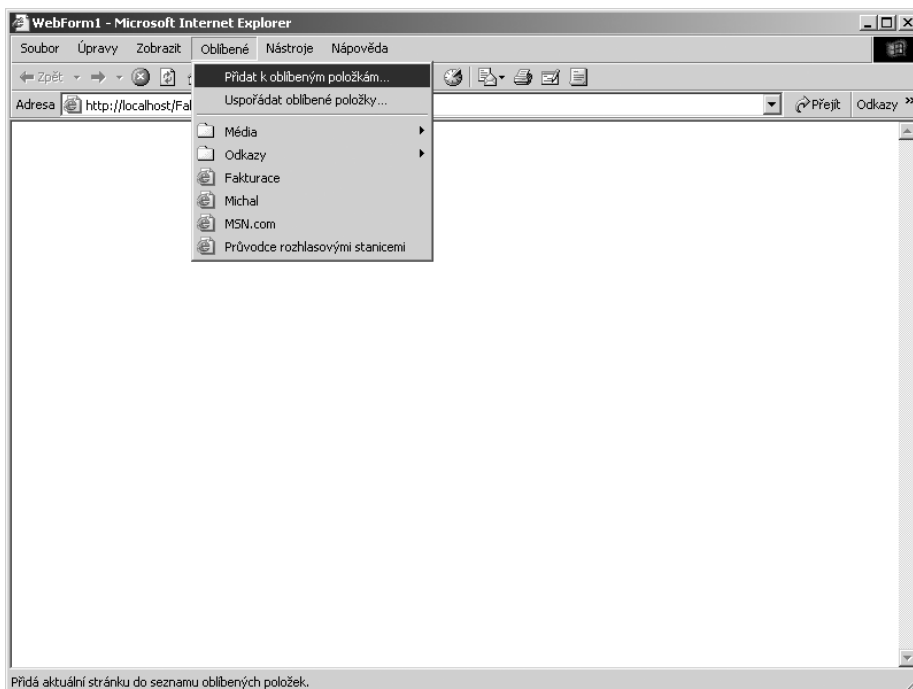
1. Nová aplikace (projekt)



Obr. 1.8: Vytvoření projektu



Obr. 1.9: Zobrazení stránky projektu



Obr. 1.10: Zařazení stránky projektu mezi oblíbené položky



Není-li řádek s URL adresou viditelný, je to způsobeno nastavením Visual Studia. Pravděpodobně je v nabídce **View** nastavena volba **Full Screen**.



Čtenáři obvykle očekávají, že první kapitola obsahuje program, který dokáže napsat „Hello, world!“. Náš první program, resp. webová stránka, to nevědí. Je to dáno koncepcí této knihy, neboť tato webová stránka bude v další kapitole použita. V našem případě by se tedy jednalo o „slepou uličku“. Naučili jste se však, jak se vytváří nový projekt, jakým způsobem se překládá a vyvolává přeložená webová stránka, jakým způsobem ji lze zahrnout do internetového prohlížeče a jak ji z něj vyvolávat.

Budete-li příště otevírat již existující projekt, a ten se bude vyskytovat mezi posledními čtyřmi nabízenými projekty, klepnete pouze na jeho nabízené jméno. Nebude-li v nabídce obsažen, klepnete na tlačítko **Open Project** a standardním způsobem, prostřednictvím dialogového okna, projekt vyhledáte.

Ještě než přejdete k další kapitole, zkuste tento projekt otevřít, znovu přeložit a vyvolat. Zároveň zkuste naši nově vytvořenou stránku z tohoto projektu vyvolat i z oblíbených položek.

Při překladač jste si možná všimli, že kromě příkazu **Build** → **Rebuild** projektu **Fakturace** se ve stejné nabídce objevuje i příkaz **Build** →

1. Nová aplikace (projekt)

Rebuild Solution. Řekli jsme si, že každé řešení bude obsahovat jediný projekt. Proto lze překlad vyvolat libovolnou volbou. O řešení zatím stačí vědět, že je zde uložen poslední stav zobrazení vašeho projektu (případně vašich projektů). Po otevření řešení se tedy zobrazí všechny stránky, okna i záložky tak, jako když jste projekt naposled zavírali. Jak vytvořit projekt.

- ✓ Jak přeložit projekt.
- ✓ Jak vyvolat stránku projektu.
- ✓ Jak zahrnout projekt mezi oblíbené položky.
- ✓ Co je to okno Toolbox.
- ✓ Co je to okno Output.
- ✓ V jakém režimu může být stránka zobrazena.



2.

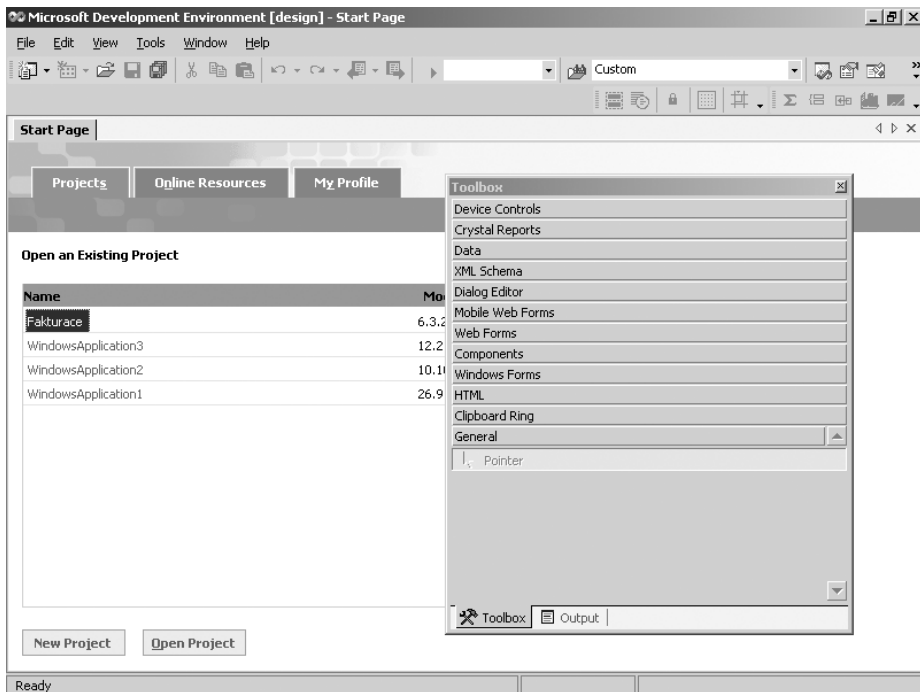
První funkční aplikace – volání stránek

Stránka, kterou jsme vytvořili v předcházející kapitole, bude v naší aplikaci sloužit jako menu, ze kterého se budou prostřednictvím hypertextových odkazů volat další stránky. Bude představovat hlavní stránku. Z ní se budou vyvolávat například stránky pro aktualizaci číselníku organizací, pro DPH, pro zaevidování faktur apod. V této kapitole vytvoříme další novou stránku, která bude v následujících kapitolách sloužit k aktualizaci číselníku organizací, a obě stránky mezi sebou propojíme. Ani v této kapitole se neseznámíme s žádným programovým kódem.

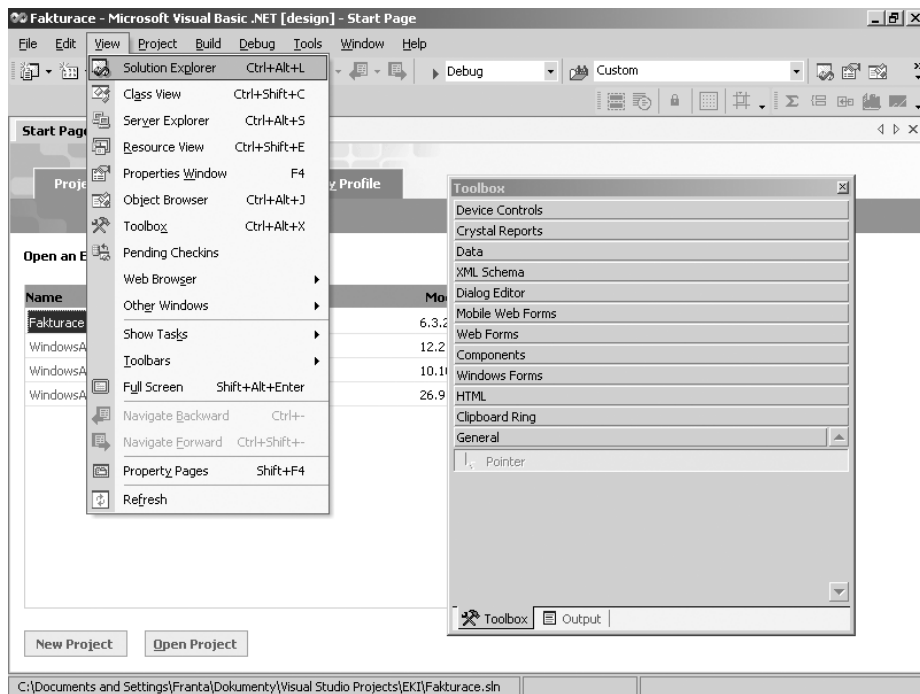
Vzhledem k tomu, že jsme Visual Studio zavřeli, musíme jej znovu otevřít. Ve výčtu posledních čtyř projektů by se měl náš projekt již nacházet, proto na něj přímo klepneme. Protože při ukončení práce s Visual Studií jsme měli otevřené záložky oken **Output** a **Toolbox**, zobrazí se nám tato okna i při otevření projektu (viz obrázek 2.1).

Po klepnutí na projekt **Fakturace** by se vám měla objevit webová stránka **WebForm1.aspx** (tj. stránka, kterou jsme měli v projektu **Fakturace** naposled otevřenou).

Není-li tomu tak (stránku jste předtím třeba zavřeli) musíte ji znovu otevřít. Po vyvolání příkazu **View** → **Solution Explorer** (viz obrázek 2.2) se objeví další okno (pravděpodobně se samo zařadí do záložek k již zobrazeným oknům **Toolbox** a **Output**), ve kterém je



Obr. 2.1: Otevření existujícího projektu

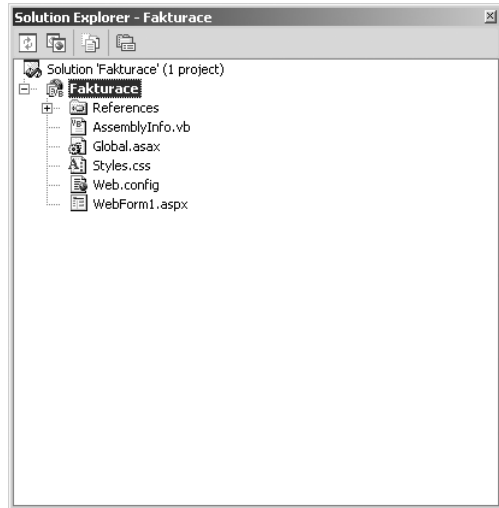


Obr. 2.2: Otevření okna Solution Explorer

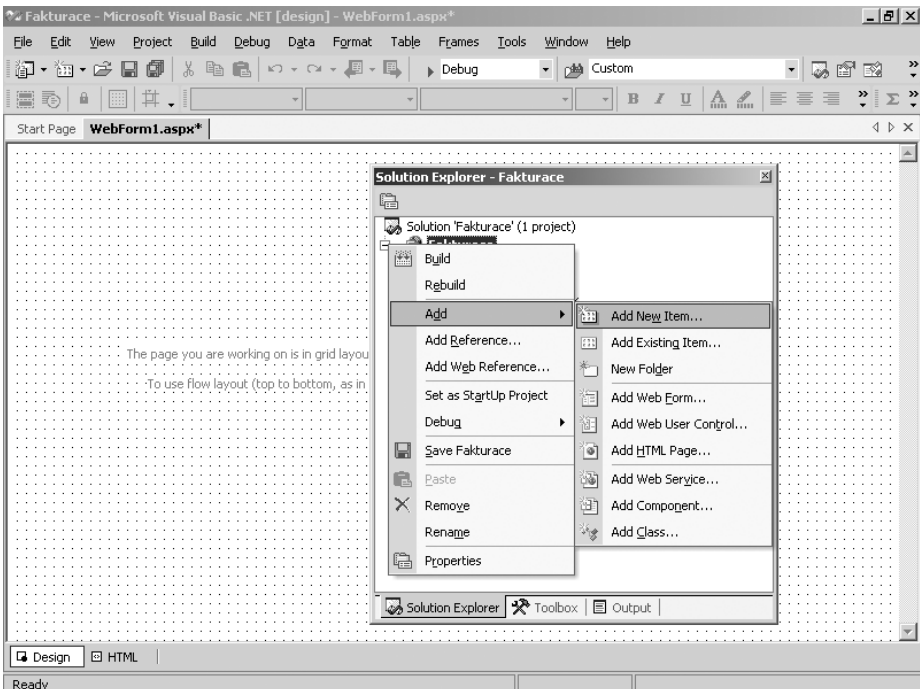
2. První funkční aplikace – volání stránek

zobrazena struktura vaší aplikace (viz obrázek 2.3). Poklepáním na požadované stránce (v našem případě tedy na stránce **WebForm1**) se zobrazí webová stránka v režimu **Design**. Ostatní stránky byly vygenerovány Visual Studiem při vytvoření nového projektu a my si jich zatím nebudeme všimnout (a raději ani v budoucnu). I když je stránka **WebForm1.aspx** otevřená, vyvolejte okno **Solution Explorer** stejně, neboť ho budeme potřebovat.

Jak jsme uvedli na začátku této kapitoly, vytvoříme nyní novou stránku. To lze provést např. tak, že v okně **Solution Explorer** klepneme pravým tlačítkem na složku **Fakturace** a z nabídek vybereme **Add** → **Add New Item** (viz obrázek 2.4). Musíte dbát na to, aby kurzor byl skutečně na složce **Fakturace**. Pokud byste myší klepli např. na stránku **WebForm1**, následná nabídka se nezobrazí, resp. zobrazí se s jiným obsahem.



Obr. 2.3: Okno Solution Explorer

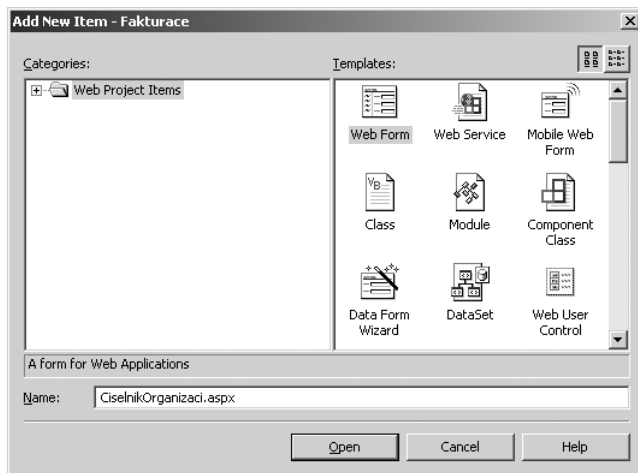


Obr. 2.4: Vytvoření nové stránky

Pomocí dialogového okna vyberte ikonu **WebForm**, nabízený název **WebForm2** přepište na **CiselnikOrganizaci.aspx** a stiskněte tlačítko **Open** (viz obrázek 2.5). Novou stránku lze vytvořit i rychleji, vyberete-li místo **Add new item** přímo **Add Web Form**.

2. První funkční aplikace – volání stránek

Pro vytváření stránek lze použít průvodce WebFormWizard, který umožňuje vytvářet stránku pomocí sekvence dialogových oken. Domníváme se, že pro profesionální programování jsou tito průvodci nepoužitelní, a tak se o nich ani nebudeme zmiňovat.



Obr. 2.5: Výběr stránky typu WebForm

Po stisknutí tlačítka **Open** se vytvoří nová stránka, která je zcela nezávislá na stránce **WebForm1**. Není ani definována, která stránka je hlavní (viz obrázek 2.6). Tuto stránku můžeme překládat a spouštět stejně jako stránku vytvořenou v předcházející kapitole, můžeme jí zařadit do oblíbených položek, přestože jsou obě stránky součástí stejného projektu. Po překladu a vyvolání se vždy zobrazí ta stránka, kterou jste měli otevřenou naposled. Můžete si to vyzkoušet, přidáte-li ji do oblíbených položek. Po odzkoušení ji z oblíbených položek zase vyjměte.

Nyní se pokusíme obě stránky spolu provázat. Zkusíme to pomocí hypertextových odkazů. Klepneme na záložku **Toolbox**. Ne zobrazí-li se **Toolbox** v následující podobě, vyberte ze svislých záložek okna **Toolbox** záložku **WebForms** a klepněte na ni. Vyberte prvek **HyperLink**, stiskněte levé tlačítko, přetáhněte ho někam do levého horního rohu pracovní plochy a opět tlačítko uvolněte (viz obrázek 2.7). Vytvořili jste nový objekt, který byl pojmenován jako **HyperLink1**. Je odvozen od prvku **HyperLink**. Objekt **HyperLink1** umožňuje volat námi požadovanou stránku. Aby tak mohl učinit, musíme mu nastavit určité vlastnosti. Vlastnosti libovolného objektu, a tedy i **HyperLink1**, se nastaví např. tak, že umístíte kurzor myši na vybraný objekt a stisknete pravé tlačítko. Tím se vyvolá kontextová nabídka (viz obrázek 2.7). Vy-



Obr. 2.6: Okno Solution Explorer po přidání nové stránky