



kap11

11. Aktivní serverové stránky (Active server pages)

11.1 Aktivní stránky

Statické stránky

Dosud připravené stránky byly statické. Jejich obsah byl v prohlížeči vždy stejný. Jistou proměnlivost jsme dosáhli časově omezeným vkládáním textu či obrázků, tuto proměnlivost zajišťoval však FrontPage. Aby se stránka změnila, bylo ji nutné znovu uložit či jinak vygenerovat její HTML kód. Nyní chceme, aby se stránky vytvářely v závislosti na datech obsažených v databázi, chceme aby stránky byly aktivní.

Skripty

Kód stránek bude proměnlivý. Proměnlivost bude realizována jednoduchými programy – tzv. skripty. Nejčastěji se k programování skriptů používají jazyky JavaScript a Visual Basic. JavaScript je univerzálnější, neboť mu rozumí prohlížeče Netscape Navigator i Internet Explorer. Visual Basicu rozumí pouze Internet Explorer (nejlépe verze 5.0 a vyšší). Visual Basic (přesněji Microsoft Visual Basic Scripting jako jedna z variant jazyka Microsoft Visual Basic) je bližší uživatelům kancelářského balíku MS Office, neboť je velmi podobný jazyku Microsoft Visual Basic for Applications užívanému ve Wordu, Excelu, PowerPointu, Accessu.

Klient a server

Výsledkem skriptů je generování HTML kódu. Generování HTML kódu probíhá u klienta (prohlížeče kódu předávaného serverem) nebo již na serveru:

- Skripty na straně klienta jsou vhodné malými nároky na server, avšak klient musí skriptům rozumět, tj. musí to být vhodný prohlížeč.
- Skripty generované již na serveru vyžadují znalost jazyka skriptů jen na serveru, klient přebírá vygenerovaný kód HTML. Kód stránky se generuje serverem před odesláním klientovi, což může výrazně zpomalit zobrazování stránek.

11.2 Aktivní stránka realizovaná klientem

Skript realizovaný klientem

Ještě než budeme vytvářet stránky podle obsahu databáze, předvedeme si jednoduchou aktivní stránku realizovanou klientem. Vytvoříme novou stránku s názvem souboru *cas1.htm* a názvem stránky *Aktuální čas*. Do stránky zapíšeme text *Aktuální čas*. (Stránku nezařazujeme do navigační struktury.)



cas1.htm

Klepneme na záložku **HTML**. Do kódu HTML dopíšeme řádky skriptu, které začínají značkou `<script>` a končí značkou `</script>` dle obr. 11-1.¹⁰⁵

OBR. 11-1: TĚLO STRÁNKY CAS1.HTM

```
<body>

<p>Aktuální čas:
<script language=vbscript>
<!--
document.write(now())
-->
</script>

</body>
```

¹⁰⁵ K editaci HTML kódu a skriptů můžeme použít speciální editor dodávaný k FrontPage Microsoft Script Editor. Pokud je nainstalován, spustíme editor příkazem TOOLS, MACRO, MICROSOFT SCRIPT EDITOR nebo kombinací kláves **Shift** **Alt** **F11**.

V editoru je zobrazen HTML kód stránky. Skript realizovaný klientem vložíme v editoru snadno příkazem HTML, SCRIPT BLOCK, CLIENT, který vloží počáteční a koncové značky skriptu. Editor nám pomáhá i při zapisování příkazů, např. když zapíšeme příkaz *document* a tečku editor nám nabízí varianty pokračování příkazu. Po výběru z vyklápěcí nabídky stiskneme tabulátor.

Řada značek HTML je dostupná z levého panelu nástrojů *HTML* (v kartě **Toolbox**). V kartě **HTML Outline** je osnova stránky dle použitých značek HTML. V kartě **Script Outline** je osnova stránky dle vložených skriptů s možností snadno přidat událostní procedury.

V kartě **Quick View** pracovní plochy můžeme v editoru vidět náhled zobrazení stránky v prohlížeči.

Microsoft Script Editor bychom mohli použít k prvotní tvorbě stránek. Na rozdíl od programu FrontPage neposkytuje však podporu pro sledování vazeb mezi stránkami (navigační strukturu).



Poznámky ke skriptu:

- Za úvodní značkou *script* upřesníme jazyk skriptu.
- Značky `<!--` a `-->` obklopují nezobrazovaný text. Může to být komentář ke stránce¹⁰⁶, v našem případě je to skript.
- Příkazem *document.write* zapíšeme na stránku text.
- Funkce *Now()* vrací aktuální datum a čas.

Zatím jsme zobrazili výsledek zvláštní funkce, která nevyžaduje žádné vstupy. Předvedme si interakci čtenáře a výsledného zobrazení stránky. Uživatel zadá text, který se vypíše na stránku. Za příkaz `</script>` doplníme do stránky *cas1.htm* další text a skript (viz obr. 11-2).

OBR. 11-2: DALŠÍ SKRIPT VE STRÁNKĚ CAS1.HTM

```
<p>Aktuální místo:  
<script language=vbscript>  
<!--  
Misto=InputBox("Zadejte místo, kde jste!")  
document.write(Misto)  
-->  
</script>
```

Poznámky ke skriptu:

- Funkcí *InputBox* otevřeme dialogové okno, v němž uživatel stránky vyplní text. Skript uloží aktuální místo do proměnné *Misto*, kterou vypíše na stránku.

Zobrazíme stránku *cas1.htm* v prohlížeči Internet Explorer a zadáme příkaz ZOBRAZIT, ZDROJOVÝ KÓD. V kódu vidíme oba naše skripty, realizují se až v prohlížeči, kterým musí být Internet Explorer, neboť Netscape Navigator neumí pracovat se skripty ve Visual Basicu.

Vrátíme se do FrontPage, kde klepneme na záložku **Normal**. Vidíme, že v normálním stránkovém zobrazení jsou skripty nahrazeny ikonami.¹⁰⁷ Můžeme kombinovat editaci v normálním stránkovém zobrazení, kde píšeme běžný text stránky, a editaci HTML kódu, v němž zapisujeme skripty.

11.3 Aktivní stránka realizovaná serverem



cas2.asp

Upravíme nyní stránku *cas1.htm* na stránku *cas2.asp* tak, aby byla realizovatelná na serveru. Nejprve stránku zkopírujeme, např. klepnutím označíme stránku v seznamu složek, stiskneme kombinaci kláves **Ctrl** **C** a potom kombinaci kláves **Ctrl** **V**. Vznikne stránka *cas1_copy(1).htm*, kterou přejmenujeme na *cas2.asp*. FrontPage nás upozorní, že změnou přípony může být stránka nepoužitelná, své rozhodnutí potvrdíme klepnutím do tlačítka **Ano**.

Tělo stránky upravíme dle obr. 11-3.

OBR. 11-3: ÚPRAVA TĚLA STRÁNKY CAS2.HTM

```
<body>  
<p>Aktuální čas:  
<%=Now( ) %>  
</body>
```

Personal web
server



Server bude sestavovat stránku pro klienta, tj. pro prohlížeč. Abychom mohli testovat stránku, musíme ji číst z vhodného serveru. Součástí Windows 98 je server Personal web server 4.0¹⁰⁸. Po instalaci je nastaveno automatické spuštění serveru po spuštění Windows. Aktivitu serveru poznáme dle ikony zobrazené na pravém konci spodního řádku Windows.

Poklepáním na ikonu serveru můžeme spustit *Správce osobního serveru WWW*, kde mj. v kartě **Hlavní** můžeme zastavit server, v kartě **Upřesnit** můžeme vyjmenovat stránky,

¹⁰⁶ Komentář by musel být uveden v rámci jednoho řádku, např. `<!--komentář-->`.

¹⁰⁷ Pokud je nainstalován Microsoft Script Editor, poklepáním na ikonu v něm otevřeme stránku.

¹⁰⁸ Personal web server se instaluje z instalačního CD Windows 98 spuštěním programu `\adds-ons\pws\setup.exe`.



které budou považovány za výchozí (např. *default.htm* a *default.asp*), tj. stránky, které nebude nutné zapisovat do adresy.

V prohlížeči nemůžeme zadat zobrazení souboru *cas2.asp* z lokálního umístění (tj. nemůžeme přímo ve FrontPage klepnout do tlačítka **Preview in browser**), musíme číst stránku ze serveru, tj. zadat adresu *localhost/encian/cas2.asp*.¹⁰⁹ Stránku nyní správně zobrazí Internet Explorer, Netscape Navigator či jiné prohlížeče. Když v Internet Exploreru zadáme příkaz ZOBRAZIT, ZDROJOVÝ KÓD, neuvidíme skript, ale jeho výsledný HTML kód vygenerovaný serverem.

V aktivní stránce realizované serverem nemůžeme požadovat vstup prostřednictvím dialogového okna jako ve stránce *cas1.htm*, neboť skript se realizuje již na serveru.

11.4 ODBC

Na další stránce chceme vypsát data z databáze faktur připravené v MS Accessu. Databáze obsahuje jedinou tabulku *Deník*, v níž je 165 záznamů za faktury firmy Encián z jednoho pololetí. O každé faktuře jsou uvedeny údaje:

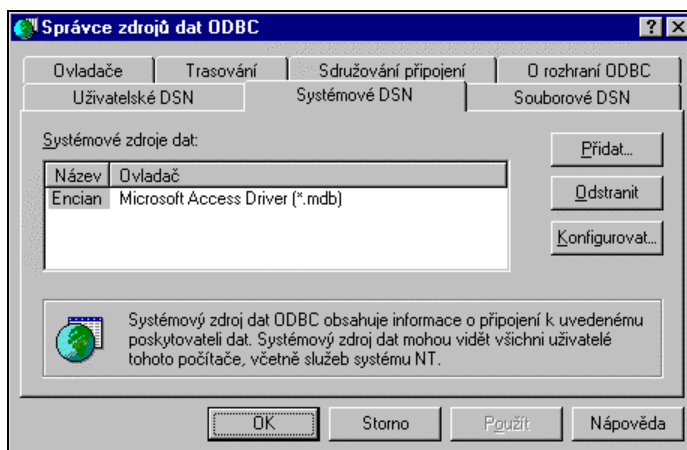
- *Datum*: datum prodeje,
- *Program*: značka programu (BON, FIN, KAU, INV, POD, SAF)
- *Počet*: počet licencí programu prodaných na jednu fakturu,
- *Typ*: typ zákazníka (*P* = průmyslové podniky, *F* = finanční firmy, *B* = banky, *J* = jiné subjekty).

ODBC

Server pracuje s daty prostřednictvím technologie ODBC. ODBC (*Open Database Connectivity*) je technologie pro přístup k datům nezávislá na typu databáze. Díky rozhraní ODBC můžeme k interakci s databází využívat syntaxi jazyka SQL (*Structured Query Language*).

Abychom mohli s databází prostřednictvím ODBC komunikovat, musíme nejdříve nadefinovat datový zdroj ODBC. Ve Windows zadáme příkaz START, NASTAVENÍ, OVLÁDACÍ PANELY a poklepáme na ikonu **Zdroje dat ODBC (32bitové)**. Zobrazí se dialogové okno **Správce zdrojů dat ODBC** (viz obr. 11-4).

OBR. 11-4: DIALOGOVÉ OKNO SPRÁVCE ZDROJŮ DAT ODBC



Nabízí se tři typy datových zdrojů:

- uživatelský: datový zdroj dostupný uživateli, který ho vytvoří, zdroj je dostupný pouze z počítače, na němž byl definován,
- systémový: datový zdroj dostupný všem uživatelům na počítači, na němž byl definován,
- souborový: datový zdroj dostupný všem uživatelům, kteří mají na svém počítači nainstalovány potřebné ODBC.

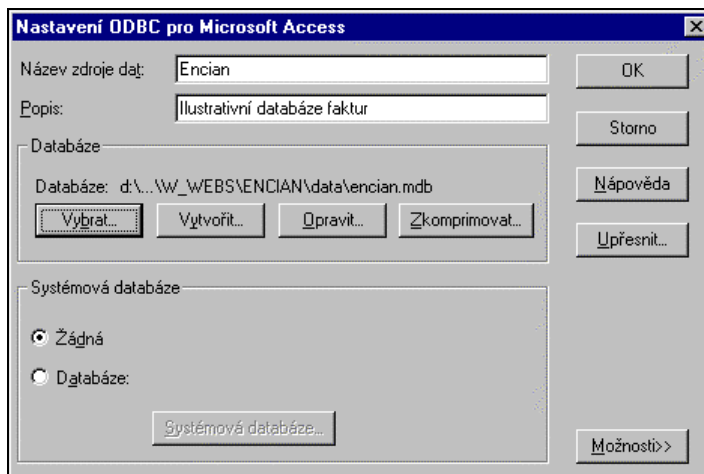
Pro zpřístupnění dat na webu budeme využívat systémový zdroj dat. Klepneme na záložku **Systémové DSN** a v kartě **Systémové DSN** klepneme do tlačítka **Přidat**.

¹⁰⁹ Výběr lokálního adresáře, z něhož čerpá server weby pro *localhost*, lze nastavit ve Správci osobního serveru WWW v kartě **Upřesnit**, když klepneme do tlačítka **Upravit vlastnosti**. (Adresář *encian* bude podadresářem vybraného lokálního adresáře.)



V dialogovém okně **Vytvoření nového zdroje dat** vybereme ovladač *Microsoft Access Driver (*.mdb)* a klepneme do tlačítka **Dokončit**.

OBR. 11-5: DIALOGOVÉ OKNO NASTAVENÍ ODBC PRO MICROSOFT ACCESS



Podle obr. 11-5 vyplníme dialogové okno **Nastavení ODBC pro Microsoft Access**:

- Vyplníme pole *Název*. Jméno bude využíváno ve skriptech pro přístup k datům z databáze.
- Vyplníme pole *Popis*, v němž můžeme stručně popsat databázi.
- Klepneme do tlačítka **Vybrat** a dohledáme umístění databáze. (Databázi před tím importujeme do podadresáře *data* našeho webu.)
- Práci s dialogovým oknem ukončíme.

11.5 Čerpání dat z databáze



vypis.asp

Vytvoříme novou stránku s názvem souboru *vypis.asp* a názvem stránky *Výpis faktur*. Do těla stránky zapíšeme HTML kód dle obr. 11-6.

OBR. 11-6: TĚLO STRÁNKY VYPIS.ASP

```
<body>
<%Set Spojeni = Server.CreateObject("ADODB.Connection")
Spojeni.Open "encian", "guest"
Set Faktury=Spojeni.Execute("Select * From Deník")%>
<table>
  <%Do While Not Faktury.EOF%>
    <tr>
      <td><%=Faktury("Datum")%> </td>
      <td><%=Faktury("Program")%> </td>
      <td><%=Faktury("Počet")%> </td>
      <td><%=Faktury("Typ")%></td>
    </tr>
    <%Faktury.MoveNext
  Loop%>
</table>
</body>
```

Komentář k tělu stránky *vypis.asp*:

CreateObject


- Pro usnadnění přístupu k datům a dalších operací nabízí Visual Basic tzv. komponenty, především komponentu přístupu k datům *ActiveX Data Objects* (ADO). Komponentu vytváříme metodou *CreateObject* objektu *Server*¹¹⁰. Po vytvoření komponenty můžeme pracovat s jejími vlastnostmi a metodami a provádět na serveru různé operace.

¹¹⁰ Pojmy *objekt*, *metoda*, *vlastnost*, *argument* jsou vysvětleny podrobněji např. ve skriptech Kubálek, T. – Luňáček, M. – Littová, J.: *Manažerská informatika. Tabulkový program Microsoft Excel verze 97 CZ. VŠE, Praha 1998. 2. vydání. ISBN 80-7079-309-0* (kap. 11.3, str. 130 – 131).



- Connection* – V našem příkladu vytváříme metodou *CreateObject* objekt *Connection* komponenty *ADODB* nazvaný *Spojeni*.
- Open* – Metoda *Open* otevírá spojení se zdrojem dat, tj. připojíme se k databázi. Má tři argumenty: jméno zdroje, identifikaci uživatele a heslo. V případě uživatele *guest* heslo není nutné. Jméno zdroje jsme definovali prostřednictvím ODBC (viz kap. 11.4).
- Execute* – V rámci vytvořeného objektu *Spojeni* metodou *Execute* realizujeme jednoduchý SQL dotaz, v němž vybíráme všechna pole ze všech vět (záznamů) tabulky *Deník*. Výsledek dotazu je datová sada, nazveme ji *Faktury*.
- <table>* – Na stránce zahájíme generování tabulky značkou *<table>*.
- Do While* – Ukazovátko je nastaveno po vytvoření objektu *Faktury* na první větě. Řádky příkazu *Do While* – *Loop* provádíme opakovaně, dokud ukazovátko není za poslední větou datové sady *Faktury* (*Faktury.EOF*, EOF = end of file).
- V rámci příkazu *While* – *Loop* pro aktuální větu (na níž je ukazovátko, tj. zpočátku pro první větu) objektu *Faktury* vypisujeme její pole.
- Pole objektu *Faktury* zapisujeme ve tvaru název objektu a v závorce název pole (může být vyjádřen proměnnou, v našem případě je vyjádřen konstantou, proto je v uvozovkách), např. *Faktury("Datum")* zastupuje pole *Datum* z aktuální (zpočátku první) věty).
- =* – Příkazem *<%=proměnná%>* vypisujeme na stránku proměnnou.
- <tr>* – Značky *<tr>* a *</tr>* vyznačují začátek a konec řádku tabulky (tr = table row).
- <td>* – Značky *<td>* a *</td>* vyznačují začátek a konec buňky v rámci řádku tabulky (td = table data).
- MoveNext* – Nakonec posuneme ukazovátko v datové sadě *Faktury* o jednu větu dále (*Faktury.MoveNext*).
- Loop* – Skupinu opakovaně prováděných příkazů ukončujeme příkazem *Loop*. Z příkazu *Loop* se program vrátí na příkaz *Do While*, kde otestuje podmínku.
- </table>* – Po přesunutí ukazovátko za poslední větu v objektu *Faktury* ukončíme tvorbu tabulky (*</table>*).
- Začátek stránky v prohlížeči je uveden v obr. 11-7.

OBR. 11-7: UKÁZKA VYPIS.ASP

Adresa  http://localhost/encian/vypis.asp	
3.1.2000	BON 1 P
3.1.2000	FIP 1 P
4.1.2000	BON 1 B
4.1.2000	FIN 1 B
4.1.2000	ESO 1 B
5.1.2000	MPS 1 P

11.6 Nabídka z databáze



program.asp

Vytvoříme novou stránku s názvem souboru *program.asp* a názvem stránky *Výběr programu*. Ve stránce chceme vypsát z tabulky *Deník* pouze seznam programů. Řádky se zkratkami programů chceme očíslovat. Do těla stránky zapíšeme HTML kód dle obr. 11-8.

OBR. 11-8: TĚLO STRÁNKY PROGRAM.ASP

```
<body>
<%Set Spoj = Server.CreateObject("ADODB.Connection")
Spoj.Open "encian", "guest"
Set Programy=Spoj.Execute("Select Distinct Program From Deník")%>
<table>
  <tr>
    <td><p align="center"><strong>Pořadí</strong></td>
    <td><strong>Program</strong></td>
  </tr>
  <%I=1
  Do While Not Programy.EOF%>
    <tr>
      <td><p align="center"><%=I%> </td>
      <td><%=Programy("Program") %></td>
    </tr>
    <%Programy.MoveNext
    I=I+1
  Loop%>
</table>
</body>
```



Nabídka
z databáze

Komentář k tělu stránky *program.asp*:

- SQL dotaz jsme připravili v MS Accessu, kde jsme vybrali dotazem z tabulky *Deník* pole *Program* a zadali jsme vlastnost dotazu *Jedinečné hodnoty ano*. Prostřednictvím schránky jsme přenesli SQL formu dotazu (příkaz Accessu ZOBRAZIT, ZOBRAZIT SQL).¹¹¹
- Tabulku (bez orámování) nadepisujeme tučným písmem ().
- Nadpis *Pořadí* zarovnáваме na střed (*align="center"*).
- V proměnné *I* počítáme vypisované řádky, proměnnou *I* vypisujeme do prvního sloupce.

OBR. 11-9

Pořadí	Program
1	BON
2	ESO
3	FIN
4	FIP
5	MPS
6	POD



fak_prg.asp

Vytvoříme novou stránku s názvem souboru *fak_prg.asp* a názvem stránky *Faktury za zvolený program*. Vypišeme v něm údaje o fakturách za program *BON*. Do těla stránky zapíšeme HTML kód dle obr. 11-10.

OBR. 11-10: TĚLO STRÁNKY FAK_PRG.ASP

```
<body>
<%Set Spoj = Server.CreateObject("ADODB.Connection")
Spoj.Open "encian", "guest"
Set Faktury=Spoj.Execute("Select * From Deník Where Program='BON'")%>
<table>
<tr>
<td><p align="center"><strong>Pořadí</strong></td>
<td><strong>Datum</strong></td>
<td><strong>Program</strong></td>
<td><strong>Počet</strong></td>
<td><strong>Typ</strong></td>
</tr>
<%I=1
Do While Not Faktury.EOF%>
<tr>
<td><p align="center"><%I%> </td>
<td><%=Faktury("Datum")%> </td>
<td><%=Faktury("Program")%> </td>
<td><%=Faktury("Počet")%> </td>
<td><%=Faktury("Typ")%></td>
</tr>
<%Faktury.MoveNext
I=I+1
Loop%>
</table>
</body>
```

Komentář k tělu stránky *fak_prg.asp*:

- Konstanta *BON* je v příkazu SQL v uvozovkách. Vnitřní uvozovky se v kódu HTML zapisují jako apostrofy.

Pokusme se nyní stránky *program.asp* a *fak_prg.asp* dát do souvislosti. Chceme, aby stránka *program.asp* sloužila jako nabídka programů. Klepnutím do zkratky programu chceme zobrazit faktury za vybraný program.

Ve stránce *program.asp* upravíme řádek vyplňující zkratku programu do buňky tabulky. Řádek nahradíme třemi řádky:

```
<%Program2=Programy("Program")
Bunka="<a href='Fak_prg.asp?Program3=" & Program2 & "'>" & Program2 & "</a>"%>
<td><%=Bunka%></td>
```

Do proměnné *Program2* vložíme hodnotu pole *Program*. Do proměnné *Bunka* přichystáme obsah buňky tabulky, např. pro program *BON* zde chystáme příkaz:

```
<a href="Fak_prg.asp?Program3=BON">BON</a>
```

¹¹¹ Text dotazu SQL jsme mírně upravili:

Pro pole *Program* neupřesňujeme, ze které je tabulky, tj. místo *Deník.Program* ponecháme *Program*.
Text dotazu nepíšeme na dva řádky.
Příkaz SQL neukončujeme středníkem.



Příkaz vypíše *BON* jako hypertextový odkaz. Při klepnutí do odkazu přecházíme na stránku *fak_prg.asp* s doplňujícím parametrem (odděleným otazníkem) *Program3* o hodnotě *BON*.

Ve stránce *fak_prg.asp* upravíme řádek s realizací dotazu. Řádek nahradíme dvěma řádky:

```
Dotaz="Select * From Deník Where Program='" & Request("Program3") & "'"  
Set Faktury=Spoj.Execute(Dotaz)%>
```

Funkce *Request* načte vstupní parametr zadaného jména. SQL dotaz připravujeme do proměnné *Dotaz*, v níž sečteme textově tři části: první je konstantní a končí apostrofem, druhá obecně dosazuje načtený parametr převzatý z jiné stránky či z přímého zadání adresy stránky včetně parametru, třetí část již jen ukončí apostrofem dotaz.

Když nyní prohlédneme stránku *program.asp* a klepneme do zkratky programu, zobrazí se stránka *fak_prg.asp* s fakturami vybraného programu.

Obdobně připravíme dvojici stránek pro výběr typu zákazníka *typ.asp* (viz obr. 11-11) a výpis faktur pro daný typ zákazníků *fak_typ.asp* včetně komunikace mezi těmito stránkami. Můžeme vyjít z kopií stránek *program.asp* a *fak_prg.asp*.



typ.asp

OBR. 11-11: TĚLO STRÁNKY TYP.ASP

```
<body>  
<%Set Spoj = Server.CreateObject("ADODB.Connection")  
Spoj.Open "encian", "guest"  
Set Typy=Spoj.Execute("Select Distinct Typ From Deník")%>  
<table>  
  <tr>  
    <td><p align="center"><strong>Pořadí</strong></td>  
    <td><strong>Typ</strong></td>  
  </tr>  
  <%I=1  
  Do While Not Typy.EOF%>  
    <tr>  
      <td><p align="center"><%=I%> </td>  
      <%Typ2=Typy("Typ")  
      Bunka="<a href='Fak_typ.asp?Typ3=" & Typ2 & "'>" & Typ2 & "</a>"%>  
      <td><%=Bunka%></td>  
    </tr>  
    <%Typy.MoveNext  
    I=I+1  
  Loop%>  
</table>  
</body>
```



fak_typ.asp

Stránka *fak_typ.asp* se liší od stránky *fak_prg.asp* jediným příkazem:

```
Dotaz="Select * From Deník Where Typ='" & Request("Typ3") & "'"
```

Přichystáme ještě přímou komunikaci stránek *fak_prg.asp* a *fak_typ.asp*. Vypisování typu programu ve *fak_prg.asp* a programu ve *fak_typ.asp* definujeme jako odkaz:

– ve stránce *fak_prg.asp* upravíme buňku typu (obdobně jako ve stránce *typ.asp*):

```
<%Typ2=Faktury("Typ")  
Bunka="<a href='Fak_typ.asp?Typ3=" & Typ2 & "'>" & Typ2 & "</a>"%>  
<td><%=Bunka%></td>
```

– ve stránce *fak_typ.asp* upravíme buňku programu (obdobně jako ve stránce *program.asp*):

```
<%Program2=Faktury("Program")  
Bunka="<a href='Fak_prg.asp?Program3=" & Program2 & "'>" & Program2 & "</a>"%>  
<td><%=Bunka%></td>
```



faktury.htm

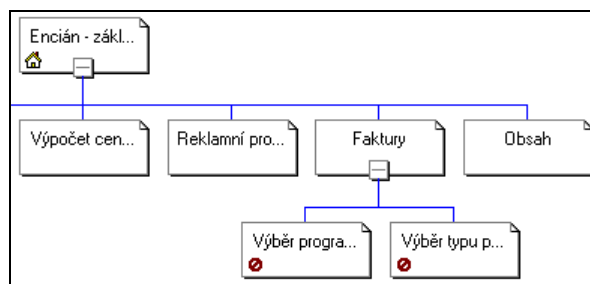
Připravíme novou stránku *faktury.htm* s názvem *Faktury*, do níž vložíme příkazem INSERT, NAVIGATION BAR děti stránky (*Child level*) svisle (*Vertical*) formou textu (*Text*). V navigačním zobrazení zařadíme do struktury našeho webu stránky *faktury.htm*, *program.asp* a *typ.asp* (viz obr. 11-12). Symbol v levém dolním rohu asp stránek signalizuje, že asp stránky nemohou mít dědické stránky v navigačním zobrazení.

Všem novým stránkám doplníme pozadí dle stránky *default.htm* příkazem FORMAT, BACKGROUND nebo úpravou příkazu začínající tělo na tvar:

```
<body stylesrc="default.htm">
```



OBR. 11-12: NAVIGAČNÍ ZOBRAZENÍ WEBU PO ÚPRAVĚ



11.7 Formulář



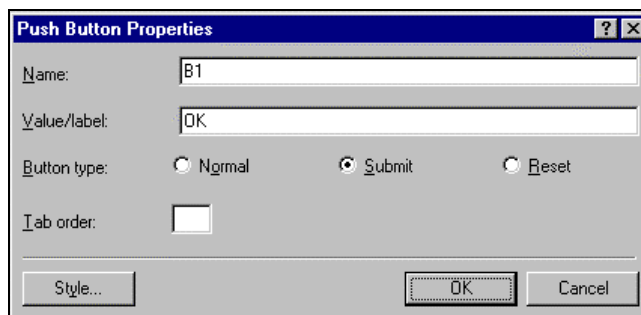
form.htm

Vložení
formuláře

Na další stránce chceme omezení výběru faktur zadat prostřednictvím formuláře. Vytvoříme novou stránku s názvem souboru *form.htm* a názvem stránky *Výběrový formulář*. Příkazem **FORMAT**, **BACKGROUND** doplníme pozadí dle stránky *default.htm*.

Příkazem **INSERT**, **FORM**, **FORM** vložíme do stránky formulář. Ve formuláři jsou zatím pouze dvě tlačítka **Submit** a **Reset**. Mezi tlačítka vložíme mezeru. Poklepáním na tlačítko **Submit** zobrazíme dialogové okno jeho vlastností (viz obr. 11-13).

OBR. 11-13: DIALOGOVÉ OKNO PUSH BUTTON PROPERTIES



Význam polí dialogového okna **Push Button Properties**:

- **Name**: Název tlačítka. Ponecháme dosazený název *B1* (button = tlačítko).
- **Value/label**: Text zobrazený na tlačítku upravíme ze *Submit* na *OK*.
- **Button type**: Typ tlačítka. Typ *Normal* neprovádí přímo datové akce, používá se k spuštění akcí naprogramovaných autorem. Typ *Submit* ukončí stránku a data zadaná na jejím formuláři předá ve vlastnostech stránky zadané stránce či jinému objektu (např. do souboru). Typ *Reset* vrátí hodnoty zadané na formuláři do výchozího stavu.
- **Tab order**: Pořadí tlačítka ve formuláři při pohybu klávesou **Tab** nebo **Shift Tab**. Pořadí doplníme, až bude formulář kompletní.

Klepnutím do tlačítka **OK** ukončíme práci s oknem. Obdobně upravíme vlastnost **Value/label** tlačítka **Reset** na *Storno*.

Vyklápěcí
menu

Klepnutím do tlačítka **Center** umístíme tlačítka ve formuláři doprostřed. Před tlačítky odesláním klávesou **Enter** vložíme nový řádek. Příkazem **TABLE**, **INSERT**, **TABLE** vložíme tabulku 6 řádků a 4 sloupců. Zadáme šířku tabulky 100 % šířky okna.

V tabulce vyplníme texty prvního a třetího sloupce dle obr. 11-17.

Kurzor přichystáme do prvního řádku druhého sloupce. Zadáme příkaz **INSERT**, **FORM**, **DROP-DOWN MENU**. Poklepáme na vložený objekt vyklápěcího menu a upravíme parametry dle obr. 11-14:

- V poli **Name** nazveme vkládanou proměnnou *Program*. Pod tímto názvem bude předávána další stránce (v rámci výsledků vyplnění formuláře).



OBR. 11-14: DIALOGOVÉ OKNO DROP-DOWN MENU PROPERTIES

Choice	Selected	Value
všechny	No	*
Bonita	No	BON
ESO	No	
Finanční analýza	No	FIN
Finanční plán	No	FIP
Mezip. srovnávání	No	MPS
Podnikatel	No	POD

- Tlačítkem **Add** postupně přidáváme jednotlivé řádky budoucí vyklápěcí nabídky.

OBR. 11-15: DIALOGOVÉ OKNO ADD CHOICE

Zobrazí se dialogové okno **Add Choice** (viz obr. 11-15). Hodnotu, kterou chceme při výběru řádku odesílat (v proměnné *Program*), vyplníme v poli *Specify Value*. Pokud hodnotu (*Specify Value*) ne zadáme (viz pravá část obr. 11-15), odesílá se z formuláře hodnota stejná jako text řádku nabídky (*Choice*). Výběrem *Selected* bychom považovali hodnotu za výchozí pro vyklápěcí nabídku.

- Tlačítkem **Modify** můžeme později modifikovat informace o řádcích nabídky.
- Tlačítkem **Remove** můžeme libovolný řádek vyklápěcí nabídky odstranit.
- Tlačítka **Move Up** a **Move Down** můžeme měnit pořadí řádků nabídky.
- V poli *Height* můžeme zvýšit počet řádků, na nichž se bude nabídka zobrazovat. Při větším počtu než 1, nebude aplikovaná vyklápěcí nabídka, ale nabídka se svislým posuvníkem.
- V poli *Tab order* určujeme pořadí pole ve formuláři pro pohyb klávesou **Tab**.
- V poli *Allow multiple selections* můžeme povolit výběr více voleb z nabídky.
- Klepnutím do tlačítka **OK** ukončíme práci s oknem **Drop-Down Menu Properties**.

Zaškrtnávací
pole

Kurzor přichystáme do třetího řádku druhého sloupce. Zadáme příkaz INSERT, FORM, CHECK BOX. Poklepáme na vložený objekt zaškrtnávacího pole a upravíme parametry dle obr. 11-16.

Pole *Value* obsahuje hodnotu, která bude v proměnné nazvané dle pole *Name* (*P*) předána další stránce. Pole *Initial state* definuje výchozí stav zaškrtnutí (*Checked* – zaškrtnuto). V poli *Tab order* definujeme pořadí pole.

OBR. 11-16: OKNO CHECK BOX PROPERTIES



Obdobně přichystáme zaškrťovací tlačítka s hodnotami *B*, *F*, *J* a pořadími 3, 4, 5 (viz obr. 11-17).

OBR. 11-17: VÝSLEDNÝ TVAR FORMULÁŘE

Program:	všechny	Měsíc faktury:	
Typ zákazníka		Počet licencí	
průmysl podniky:	<input type="checkbox"/>	1	<input checked="" type="radio"/>
banky:	<input type="checkbox"/>	2	<input type="radio"/>
finanční firmy:	<input type="checkbox"/>	1 nebo 2	<input type="radio"/>
jiné firmy:	<input type="checkbox"/>	libovolný	<input type="radio"/>
<input type="button" value="OK"/> <input type="button" value="Storno"/>			

Textové pole

Kurzor přichystáme do prvního řádku čtvrtého sloupce. Zadáme příkaz INSERT, FORM, ONE-LINE TEXT BOX. Poklepáme na vložený objekt textového pole a upravíme parametry dle obr. 11-18.

Poznámky k polím dialogového okna **Text Box Properties**:

- V názvech polí formuláře nepoužíváme háčky a čárky.
- Do pole *Initial value* bychom mohli zapsat výchozí hodnotu nabízenou ve formuláři.
- V poli *Width in characters* zadáváme maximální délku vstupního textu ve znacích (v našem případě 1, protože vyplňujeme pouze měsíce 1. pololetí). Délka ovlivní šířku textového pole.
- Jestliže v poli *Password field* vybereme hodnotu *Yes*, budou se ve vstupním poli při vyplňování zobrazovat hvězdičky, tj. nebude se zobrazovat vkládaná hodnota.
- Klepnutím do tlačítka **Validate** vymezíme podmínky pro vstupní hodnoty (viz obr. 11-19):

OBR. 11-18: OKNO TEXT BOX PROPERTIES

OBR. 11-19: DIALOGOVÉ OKNO TEXT BOX VALIDATION

- V poli *Data type* upřesňujeme typ vstupu (*Text*, *Integer*, *Number*).



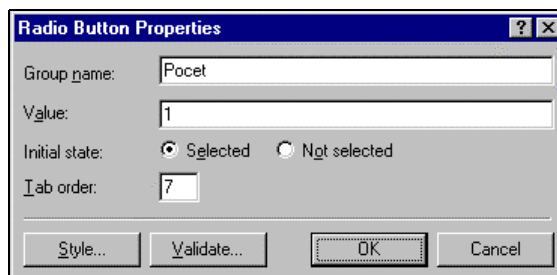
- V poli *Display name* zapíšeme název pole, který bude zobrazen při případném chybném vstupu ve formuláři.
- V případě textového typu můžeme v sekci *Text format* povolit písmena, číslice, mezery či jiné znaky.
- V případě čísel (celých – *Integer* nebo reálných – *Number*) je možné upřesnit formát čísla, jeho délku a vstupní hodnoty.
- V našem případě omezíme hodnotu na celé číslo z intervalu <1;6>.

Přepínací pole

Kurzor přichystáme do třetího řádku, čtvrtého sloupce. Zadáme příkaz INSERT, FORM, RADIO BUTTON. Poklepáme na objekt přepínacího pole a upravíme parametry dle obr. 11-20:

OBR. 11-20: OKNO RADIO BUTTONS PROPERTIES

- Název *Group name* bude (na rozdíl od zaškrťovacího tlačítka) společný pro skupinu přepínacích tlačítek. Uživatel bude moci vybrat jen jedno z tlačítek skupiny. Výstupem bude proměnná nazvaná dle pole *Group name* o hodnotě z pole *Value* pro příslušné tlačítko.



- V poli *Initial state* můžeme (pro jedno z tlačítek skupiny *Pocet*) zadat zaškrtnutí jako výchozí hodnotu.
- V poli *Tab order* definujeme pořadí tlačítka ve formuláři.
- Klepnutím do tlačítka **Validate** bychom mohli zadat nutnost výběru hodnoty ve skupině.

Obdobně doplníme tlačítka do dalších tří řádků se stejnou hodnotou pole *Group name*, s různými hodnotami 2, 3, 4 v poli *Value* a s různými hodnotami 8, 9, 10 v poli *Tab order*.

Akce formuláře

Nyní musíme pro formulář upřesnit akci, která se provede při ukončení formuláře tlačítkem **OK** (tj. tlačítkem typu *Submit*). Klepneme pravým tlačítkem do formuláře a z místní nabídky vybereme FORM PROPERTIES. V dialogovém okně **Form Properties** vyplníme název formuláře (*Form name*) *Formular1*, zvolíme pole *Send to other* a klepneme do tlačítka **Options**. V dialogovém okně **Options for Custom Form Handler** vyplníme do akce (*Action*) název stránky *odpoved.asp*, kterou vytvoříme v dalším výkladu. Opětovným klepnutím do tlačítka **OK** uzavřeme obě okna.



odpoved.asp

Stránku *form.htm* uložíme. Vytvoříme novou stránku s názvem souboru *odpoved.asp* a názvem stránky *Vybrané faktury*. Příkazem FORMAT, BACKGROUND doplníme pozadí dle stránky *default.htm*. Do těla stránky zapíšeme HTML kód dle obr. 11-21.

Komentář k tělu stránky *odpoved.asp*:

- V úvodu jsou uloženy do stejnojmenných proměnných parametry předané z formuláře. Vidíme, že předaných parametrů může být více (*Program, P, B, F, J, Mesic, Pocet*)¹¹²
- V dalších dvou řádcích se připojíme k databázi.
- Složitě je sestavování SQL příkazu do proměnné *Dotaz*:
 - Začátek *Select * From Denik Where* je společný pro všechny varianty vyplnění formuláře.
 - Jestliže uživatel vyplnil do pole *Program* volbu všechny, byla předána hodnota parametru *Program **. V tom případě hledáme programy se zkratkou v abecedě po písmenu *A*, což jsou všechny programy. Jinak hledáme pouze ty programy, jejichž zkratka se shoduje s předaným parametrem *Program*.
 - Jestliže uživatel nezaškrtně žádný typ zákazníka, doplní se do dotazu *And (False)* a nevypíší se tak žádné faktury.
 - Jestliže uživatel zaškrtně typy zákazníka, doplní se do dotazu *And (False or Typ='P' or Typ='B'....)* dle zaškrtnutých typů zákazníků. Vypíší se faktury pro vybrané typy zákazníků.
 - Jestliže uživatel nevyplní ve formuláři měsíc (délka zadání měsíce je nulová), není datum faktury omezeno.
 - Jestliže uživatel vyplní ve formuláři měsíc, doplní se do dotazu *And Month(Datum)=?*, kde ? je hodnota měsíce, funkce *Month* vrací z data měsíc.

¹¹² Při zobrazování stránky *odpoved.asp* nejsou parametry zobrazeny v řádku adresy, stránku *odpoved.asp* bychom nemuseli zobrazovat z formuláře, ale přímo např. zadáním adresy:
<http://localhost/encian/odpoved.asp?program3=BON&P=ano&B=ne&F=ne&J=ne&Mesic=2&Pocet=1>



OBR. 11-21: TĚLO STRÁNKY ODPOVED.ASP

```
<body stylesrc="default.htm">
<%Program4=Request("Program")
P4=Request("P")
B4=Request("B")
F4=Request("F")
J4=Request("J")
Mesic4=Request("Mesic")
Pocet4=Request("Pocet")
Set Spoj = Server.CreateObject("ADODB.Connection")
Spoj.Open "encian", "guest"
Dotaz="Select * From Deník Where "
If Program4="*" Then
    Dotaz=Dotaz & "Program>'A' "
Else
    Dotaz=Dotaz & "Program=' ' & Program4 & ' ' "
End If
Dotaz = Dotaz & "And (False "
If P4="ano" Then Dotaz = Dotaz & "Or Typ = 'P' "
If B4="ano" Then Dotaz = Dotaz & "Or Typ = 'B' "
If F4="ano" Then Dotaz = Dotaz & "Or Typ = 'F' "
If J4="ano" Then Dotaz = Dotaz & "Or Typ = 'J' "
Dotaz = Dotaz & ") "
If Len(Mesic4)<>0 Then
    Dotaz = Dotaz & "And (Month(Datum)=" & Mesic4 & ") "
End If
Select Case Pocet4
    Case 1
        Dotaz=Dotaz & "And Počet=1 "
    Case 2
        Dotaz=Dotaz & "And Počet=2 "
    Case 3
        Dotaz=Dotaz & "And (Počet=1 or Počet=2) "
End Select
Set Faktury=Spoj.Execute(Dotaz)%>
<table>
<tr>
<td><p align="center"><strong>Pořadí</strong></td>
<td><strong>Datum</strong></td>
<td><strong>Program</strong></td>
<td><strong>Počet</strong></td>
<td><strong>Typ</strong></td>
</tr>
<%I=1
Do While Not Faktury.EOF
    Datum2 =Faktury("Datum")
    Program2=Faktury("Program")
    Pocet2 =Faktury("Počet")
    Typ2    =Faktury("Typ")%>
<tr>
<td><p align="center"><%=I%> </td>
<td><p align="right"><%=Datum%> </td>
<%
    Bunka="<a href='Fak_prg.asp?Program3=" & Program2 & "'>" & Program2
    Bunka=Bunka & "</a>"%>
<td><%=Bunka%></td>
<td><%=Pocet2%> </td>
<%
    Bunka="<a href='Fak_typ.asp?Typ3=" & Typ2 & "'>" & Typ2 & "</a>"%>
<td><%=Bunka%></td>
</tr>
<%Faktury.MoveNext
    I=I+1
Loop%>
</table>
</body>
```



- Jestliže uživatel zaškrtnul přepínač počet licencí 1 (popř. 2, 1 nebo 2, libovolný), je v proměnné *Pocet* uložena hodnota 1 (popř. 2, 3, 4). V případě hodnot 1, 2 nebo 3 jsou vypisovány pouze faktury s příslušným počtem licencí.
- Po realizaci dotazu je nejdříve zahájeno vypisování tabulky (`<table>`) a vypsáno tučně (``) záhlaví tabulky (slovo *Pořadí* vycentrováno).
- Další příkazy jsou prováděny pro jednotlivé vybrané faktury:
 - Do proměnných *Datum*, *Program*, *Pocet*, *Typ* jsou připraveny hodnoty z aktuální věty.
 - Do buněk řádku jsou tyto proměnné vypsány.
 - Proměnné *Program* a *Typ* jsou vypsány formou odkazu na výpis faktur příslušného programu či typu.
 - Na závěr vypisování věty je zvýšeno počítadlo vět *I* o jedničku.
- Po ukončení vypisování vět je ukončena tabulka (`</table>`).



form_ram.htm

Pokud do formuláře chceme zadávat různé parametry a prohlížet výsledky, musíme se opakovaně k formuláři vracet. Navíc ve výsledku nejsou zobrazeny vstupní parametry. Proto vytvoříme sestavu rámu *form_ram.htm*, v níž budou ve svislém uspořádání zobrazeny stránky *form.htm* a *odpoved.asp*.

Zadáme příkaz FILE, NEW, PAGE a v kartě **Frames Pages** vybereme *Vertical Split*. Příkazem FILE, SAVE uložíme sestavu rámu pod názvem stránky *Vybrané faktury* do souboru *form_ram.htm*. Tlačítkem **Set Initial Page** vybereme do levého rámu stránku *form.htm* a do pravého rámu stránku *odpoved.asp*. V levém rámu klepneme pravým tlačítkem do formuláře a z místní nabídky vybereme FORM PROPERTIES. V dialogovém okně vlastností formuláře klepneme do tlačítka na konci řádku *Target Frame* a vybereme pravý rám (*right*). Tlačítkem **OK** postupně zavřeme všechna okna. Rám i jednotlivé stránky rámu uložíme.

V navigačním zobrazení umístíme stránku *form_ram.htm* do větve *Faktury* vedle stránky *Výběr typu podniku*. Když z výchozí stránky webu zvolíme *Faktury* a *Vybrané faktury*, můžeme v levém rámu zadávat výběry a v pravém rámu prohlížet výsledky (viz obr. 11-22).¹¹³

OBR. 11-22: PROHLÍŽENÍ VÝBĚRU FAKTUR V RÁMECH

The screenshot shows a web browser window with the address bar displaying `http://localhost/encian/form_ram.htm`. The page is divided into two main sections. The left section contains a form with the following fields:

- Program:** A dropdown menu with the value "všechny".
- Měsíc faktury:** A text input field with the value "5".
- Typ zákazníka:** A label for the customer type.
- Poččet licencí:** A label for the number of licenses.
- průmysl. podniky:** A checkbox that is unchecked.
- banky:** A checkbox that is checked.
- finanční firmy:** A checkbox that is checked.
- jiné firmy:** A checkbox that is unchecked.
- 1** and **2** are radio buttons next to the license count field.
- libovolný** is a radio button next to the license count field.

At the bottom of the form are two buttons: **OK** and **Storno**.

The right section displays a table of invoices with the following columns: **Pořadí**, **Datum**, **Program**, **Poččet**, and **Typ**. The table contains 7 rows of data:

Pořadí	Datum	Program	Poččet	Typ
1	5.5.2000	BON	1	B
2	5.5.2000	FIN	1	B
3	15.5.2000	BON	1	F
4	15.5.2000	MPS	1	F
5	15.5.2000	ESO	1	F
6	22.5.2000	BON	1	B
7	22.5.2000	FIN	1	B

At the bottom right of the page, there is a text label: **Poslední aktualizace stránky: 16.06.2000**.

11.8 Dynamická stránka

Dynamická stránka

Na závěr vytvoříme ukázkou dynamické stránky. Dynamická stránka se mění v průběhu zobrazení. Program musí být realizován na straně klienta. Vytvoříme novou stránku s názvem souboru *cas3.htm* a názvem stránky *Aktuální čas*.



cas3.htm

Do stránky vložíme komponentu ActiveX příkazem INSERT, ADVANCED, ACTIVE X CONTROL¹¹⁴, MICROSOFT FORMS 2.0 LABEL¹¹⁵. Poklepáním na komponentu zobrazíme její vlastnosti a upravíme název (*Name*) na *Cas*. Modifikujeme HTML kód dle obr. 11-23.

¹¹³ Pokud bychom nyní stránku *form.htm* vyplňovali samostatně, stránka *odpoved.asp* by se zobrazovala do samostatného okna, neboť by nenalezla rám *right*.

¹¹⁴ Pokud není příkaz ACTIVE X CONTROL k dispozici, zkontrolujeme nastavení po příkazu TOOLS, PAGE OPTIONS v kartě **Compatibility** (viz kap. 9.10).

¹¹⁵ Pokud není komponenta k dispozici, přidáme ji výběrem ze seznamu po klepnutí do tlačítka **Customize**.



OBR. 11-23: KÓD STRÁNKY CAS3.HTM

```
<head>
... (řádky připravené FrontPage)
<script LANGUAGE="VBScript">
  <!--
Sub Window_OnLoad( )
  Cas.Caption=Now
  Window.SetTimeout "Casomira", 2000
End Sub
Sub Casomira
  Cas.Caption=Now
  Window.SetTimeout "Casomira", 2000
End Sub
-->
</script>
</head>
<body>
<p align="center">
<object classid="clsid:978C9E23-D4B0-11CE-BF2D-00AA003F40D0" id="Cas"
  width="483" height="58">
  <param name="BackColor" value="16777215">
  <param name="FontName" value="MS Times New Roman">
  <param name="FontHeight" value="825">
</object>
</body>
```

Po načtení stránky se sama spustí procedura *Window_OnLoad*, která textu komponenty *Cas* přiřadí aktuální čas. Zároveň nastaví spuštění procedury *Casomira* po dvou sekundách (2 sekundy = 2000 milisekund). Procedura *Casomira* provádí opakovaně totéž. V parametrech komponenty jsou upraveny šířka a výška komponenty, písmo a velikost textu (času).

Shrnutí

1. Pro programování stránek se nejčastěji používají *jazyky* JavaScript a Microsoft Visual Basic Scripting.
2. *Generování kódu* stránek se realizuje na straně klienta nebo na straně serveru.
3. *Generování klientem* vyžaduje, aby klient (prohlížeč) znal využitý skriptovací (programovací) jazyk.
4. Při *generování serverem* se kód stránky generuje serverem před odesláním klientovi, což může výrazně zpomalit zobrazování stránek, klient (prohlížeč) však nemusí znát použitý jazyk.
5. Aktivní stránky realizované serverem (ASP) musí mít *příponu asp*.
6. *ODBC* je technologie pro přístup k datům, která je nezávislá na typu databáze.
7. Pro usnadnění přístupu k datům a dalších operací nabízí Visual Basic *komponenty*.
8. Data lze mezi stránkami předávat formou *parametrů*. Parametry doplňuje v odkazech výchozí stránka nebo je zadává uživatel ve formuláři. Parametrů v jednom odkazu může být více.
9. Ve *formuláři* lze využít řadu objektů, např. vyklápecí menu, textové pole, zaškrtnávací pole, přepínací pole. Odeslání dat je nutné parametrizovat ve vlastnostech formuláře. Formulář a stránku s odpovědí je vhodné zobrazit pomocí rámců v jedné stránce.
10. *Dynamické stránky* mění svůj vzhled v průběhu zobrazení.

Literatura

- [1] VORÁČEK, KAREL: FrontPage 2000. Tvorba dokonalých WWW stránek. Podrobný průvodce začínajícího uživatele. Praha, Grada Publishing, 1999. ISBN 80-7169-877-6.
- [2] HILLIER, SCOT – MEZICK, DANIEL: Programování Active Server Pages. Brno, Computer Press 1998. ISBN 80-7226-118-5.
- [3] CORNELL, GARY: Microsoft Visual Basic Scripting. Příručka programátora. Praha, Computer Press 1999. ISBN 80-7226-144-4.