

**Pagination pro formuláře ASP.NET**  
**Pagination for ASP.NET forms**

**Bakalářská práce**  
**Mgr. Petr Jelínek**  
**Vedoucí práce: Ing. Václav Novák, CSc.**  
**Jihočeská univerzita v Českých Budějovicích**  
**Pedagogická fakulta**  
**Katedra Informatiky**  
**Rok 2010**

## **Prohlášení**

Prohlašuji, že svoji bakalářskou práci jsem vypracoval samostatně pouze s použitím pramenů a literatury uvedených v seznamu citované literatury.

Prohlašuji, že v souladu s § 47b zákona č. 111/1998 Sb. v platném znění souhlasím se zveřejněním své bakalářské/diplomové práce, a to v nezkrácené podobě pedagogickou fakultou elektronickou cestou ve veřejně přístupné části databáze STAG provozované Jihočeskou univerzitou v Českých Budějovicích na jejích internetových stránkách.

V Českých Budějovicích dne

## **Anotace**

Tato bakalářská práce se zabývá stránkováním pro formuláře ASP.NET použitím aplikačního rámce ve verzi 3.5. Práce obsahuje 2 hlavní části. V první provedu průzkum oblíbenosti 5 různých druhů stránkování, vzhledem k jejich funkcím a vyhodnotím je. Ve druhé části ukáži, jak vytvořit stránkování v praxi pomocí Visual Studia různými způsoby. Zároveň upozorním na jednotlivá úskalí při použití tohoto prvku ve webových prezentacích.

## **Abstract**

The Bachelor thesis deals with pagination for the forms ASP.NET using application frame in version 3.5. The thesis consists of two main parts. Firstly, I am going to do the research about popularity of five different kinds of pagination according to their functions and then I am going to evaluate them. Secondly, I am going to show how to make pagination in practice using Visual Studio in different ways. I am going to emphasize some problems with using this part in web presentations.

## **Poděkování**

Rád bych poděkoval vedoucímu mé bakalářské práce Ing. Václavu Novákovi, CSc. za pomoc a cenné rady při postupu v mé práci a umožnil mi tak, zabývat se touto problematikou.

Také bych chtěl poděkovat všem přátelům, známým i neznámým, kteří se zabývali mým průzkumem a věnovali mi trochu času.

## Obsah

<b>1</b>	<b>ÚVOD .....</b>	<b>7</b>
<b>2</b>	<b>CO JE ASP.NET .....</b>	<b>8</b>
2.1	VÝHODY A NEVÝHODY .....	9
2.2	HISTORIE .....	10
2.2.1	ASP.NET 1.0 .....	11
2.2.2	ASP.NET 1.1 .....	11
2.2.3	ASP.NET 2.0 .....	11
2.2.4	ASP.NET 3.5 .....	12
2.3	PHP .....	17
2.4	DALŠÍ VERZE .....	17
<b>3</b>	<b>PAGINATION .....</b>	<b>18</b>
3.1	VYSVĚTLENÍ POJMU .....	18
3.2	PRŮZKUM .....	19
3.2.1	<i>První stránkování</i> .....	20
3.2.2	<i>Druhé stránkování</i> .....	22
3.2.3	<i>Třetí stránkování</i> .....	24
3.2.4	<i>Čtvrté stránkování</i> .....	26
3.2.5	<i>Páté stránkování</i> .....	28
3.2.6	<i>Celkové hodnocení</i> .....	30
3.3	JAK NA STRÁNKOVÁNÍ V PRAXI .....	32
3.3.1	<i>GridView</i> .....	32
3.3.2	<i>ListView</i> .....	39

3.3.3	<i>DetailsView</i> .....	42
3.3.4	<i>FormView</i> .....	44
<b>4</b>	<b>ZÁVĚR</b> .....	<b>45</b>
<b>5</b>	<b>PŘEHLED POUŽITÉ LITERATURY</b>	
<b>6</b>	<b>SEZNAM PŘÍLOH</b>	

## 1 Úvod

Počítačový průmysl patří v současné době mezi nejproduktivnější a nejdůležitější odvětví v našem světě. Pokud bychom šli do hloubky, tak celosvětová ekonomika, i když si to mnozí neuvědomují, je závislá na počítačích a s tím i souvisejícího internetu a webových stránkách.

Na internetu najdeme úplně všechno a je radost si prohlížet originální webové stránky s různým zaměřením a úchvatnou grafikou. Otázkou je, zda se v tom chumlu odkazů a „nekonečně“ dlouhých, textem popsaných, stránkách dokážeme orientovat a najít, co potřebujeme. Tento problém řeší, pokud ho autor stránek využije, stránkování (Pagination), které je hlavním tématem této bakalářské práce.

Pro tento účel jsem vytvořil webové stránky, založené na technologii ASP.NET od Microsoftu. Zde najdete 5 vybraných druhů stránkování, které mají sloužit jako hlavní opěrný bod pro průzkum, který je první ze dvou částí této práce. Ve druhé části bych chtěl dále zdůraznit různá úskalí tohoto členění textu na stránky a seznámit Vás s tvorbou stránkování v praxi pomocí různých ovládacích prvků ve vývojovém prostředí Microsoft Visual Studio s pomocí programovacího jazyku C#.

## **2 Co je ASP.NET**

Technologie ASP.NET, která pochází od společnosti Microsoft, navazuje na svého předchůdce pro tvorbu webových aplikací ASP (Active Server Pages). I když se jedná o následníka, je ASP.NET dosti odlišná od ASP a dalo by se říci, že i přelomová technologie. Je totiž založena na CLR (Common Language Runtime), která je sdílena aplikacemi postavenými na .NET Frameworku a programátoři již nejsou omezovali pouze na skriptovací jazyky, ale mohou si vybrat jakýkoliv objektově orientovaný programovací jazyk, který podporuje CLR – Visual C#, Visual Basic, Visual C++ a další.

Běžnému uživateli jsou asi z ASP.NET nejbližší webové formuláře (WebForms). Webové stránky se skládají z objektů nebo lépe řečeno z ovládacích prvků zvaných Controls, které dobře známe z klasického operačního systému Windows. Najdete zde tlačítka (Button), textová pole (TextBox), tabulky (Table), popisky (Label) a další. Každému takovému objektu je možné nastavit individuálně jejich vlastnosti, události, atd.



## **2.1 Výhody a nevýhody**

Ačkoli je ASP.NET vcelku mladá technologie, přináší sebou kromě značných výhod i některé nevýhody. Asi největší výhodou aplikací je jejich rychlost, s jakou přistupují ke klientům. Stránky se totiž již neinterpretují, ale kompilují (většina ovládacích prvků na straně serveru), a tak si programátor může vytvořit například jakýkoliv kus kódu, třeba v poznámkovém bloku, ten následně vloží na server, který ho při požadavku klienta překompiluje a uloží do cache paměti. Zde zůstane uložen pro další požadavky klientů nebo do té doby, dokud se nevloží upravený kód, který se překompiluje.

### ***Výhody nejen oproti staršímu ASP***

- vyšší rychlost díky kompilaci a ukládání do cache paměti serveru
- výběr z více programovacích jazyků
- redukce duplicitního kódu při použití šablon (ovládací prvky definované uživatelem)
- programátoři mají k dispozici bohatý výběr z knihoven tříd a velké množství ovládacích prvků, což vede k rychlejšímu vytváření aplikací
- díky kompilaci lze zachytit velké množství chyb již v počátečním vývoji

- od verze 2.0 je generován v aplikacích validní kód HTML a JavaScript
- ASP.NET může běžet na různých webových serverech s různým operačním systémem – IIS (Windows), Apache (Windows, Linux)
- vysoká míra zabezpečení
- programování designu je odděleno od kódu

### ***Nevýhody ASP.NET***

- robustní a složité (spojení .NETu, programovacího jazyka a ASP.NETu)
- není zatím tolik rozsáhlé jako například konkurenční PHP.
- menší počet hostingů

## **2.2 Historie**

I když je ASP.NET řekněme „novinkou“, prodělal již celkem výrazný rozvoj. Jak jsem již zmiňoval, celá technologie se váže na samotný .NET Framework a Microsoft už stačil vydat několik verzí, které přináší programátorům nové možnosti a snaží se jim usnadnit práci.

### **2.2.1 ASP.NET 1.0**

Již první verze zaznamenala obrovský úspěch a přinesla revoluční systém pro programování. Nárůst výkonu v aplikacích byl hlavním plusem a se vznikem .NETu se objevily i nové programovací jazyky – C#. Velice příjemným překvapením byly různé předpřipravené ovládací prvky zvané Controls a abstraktní přístup k programování webových prezentací.

### **2.2.2 ASP.NET 1.1**

Další verze 1.1, je spíše braná jako service pack, než jako plnohodnotná verze.

### **2.2.3 ASP.NET 2.0**

Obrovský skok kupředu přináší až verze 2.0, která představuje i novou verzi programovacího jazyka .NET C# 2.0 a Visual Studio 2005, které je považováno za jedno z nejlepších vývojových prostředí. U nové verze je kladen velký důraz na ulehčení práce programátorům a zeštíhlení kódu až o 70 procent. Součástí je také vestavěné API (Application Programming Interface) pro přihlašování – dosud se musel přihlašovací modul programovat prakticky pokaždé od začátku. To obsahuje i správu uživatelských rolí (Role Provider), správu profilů (Profile Provider) a kompletní strukturu pro přihlašování (Membership Provider). Pokud programátorovi defaultní API nevyhovuje, může třídu

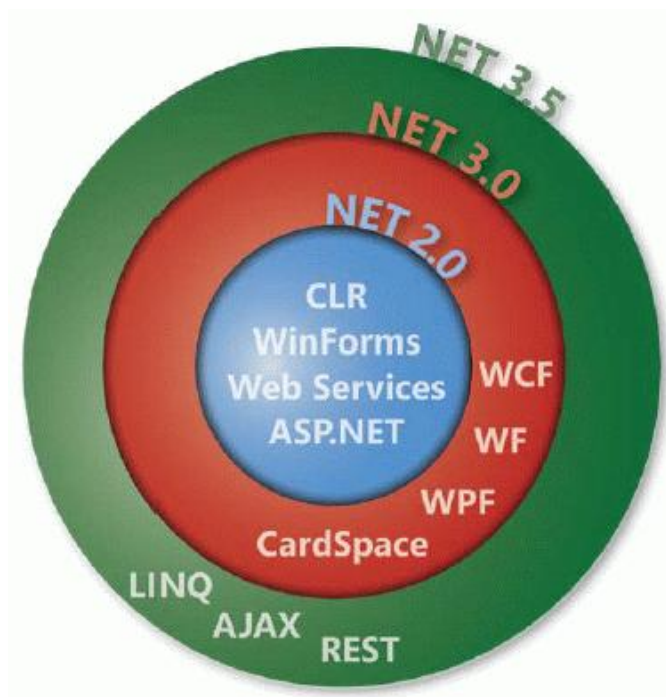
podědit a změnit vlastnosti podle svých přání. Mezi další velké inovace patří jistě kompatibilita s XHTML, bezpečnost, motivy pro změnu vzhledu webových ovládacích prvků, vzory stránek, používání MasterPage (hlavní stránka, která funguje jako „šablona“ webové prezentace, ve které jsou vnořené jednotlivé stránky, Master stránek může být na webu samozřejmě více) a mnohé další.

#### **2.2.4 ASP.NET 3.5**

Teď se možná tváříte trochu rozpačitě a přemýšlíte, kam zmizel ASP.NET 3.0. Microsoft totiž použil pouze .NET Framework 3.0 k vydání nových technologií a samotné ASP.NET bylo zařazeno až do nejnovějšího .NET Frameworku 3.5. I když .NET Framework 3.0 pracuje se stávajícím CLR (Common Language Runtime) a BCL (Base Class Library) z verze 2.0, byly do něj implementovány nové technologie:

- WPF - Windows Presentation Foundation (dříve známý Avalon, zaměřen na bohaté grafické aplikace a sloučení 2D a 3D grafiky)
- WCF - Windows Communication Foundation (slouží k budování služeb, které se orientují na zprávy)
- WF - Windows Workflow Foundation (modelování obchodních procesů pomocí diagramů)

- Windows CardSpace (technologie, která vytváří vztahy mezi weby a online službami)



obrázek<sup>1</sup>

Také podporuje již operační systémy Windows Vista a nelze ho provozovat na starých Windows 9x ani Windows 2000.

---

<sup>1</sup> schéma jednotlivých verzí .NET Frameworku, přežato z <http://aspnet.4guysfromrolla.com/articles/112107-1.aspx>

A co je tedy v .NET Frameworku 3.5 nového? Všechny novinky by se daly zařadit do následujících kategorií, které nyní podrobněji popíši.

### **LINQ**

LINQ (Language Integrated Query) je jednou z nejzásadnějších a nejdiskutovanějších novinek nového Frameworku. Díky této technologii programátoři manipulují s daty v paměti a využívají dotazů do databáze velice podobně, jak je tomu u jazyka SQL. Veškeré operace jsou prováděny pomocí jazyků C# a Visual Basic a to mnohem jednodušším a efektivnějším způsobem, než tomu bylo doposud. V podstatě tato technologie nabízí tyto 4 implementace:

- **LINQ to Objects** – manipulování s daty v paměti
- **LINQ to DataSet** – manipulování s daty, která jsou uložena v datasetech ADO.NET
- **LINQ to XML** – manipulování s daty ve formátu XML
- **LINQ to SQL** – manipulování s daty v databázích na serveru MS SQL

První 3 implementace jsou pouze funkcemi, které rozšiřují a doplňují ASP.NET, ale nejsou s ním nijak vázané. Poslední jmenovaný LINQ to SQL umožňuje vykonat dotaz do databáze a

výsledky následně vázat na nějaký ovládací prvek, aniž byste potřebovali nějaký kód navíc. Vývoj webových aplikací, tak zabere mnohem méně času a zdrojový kód je menší a přehlednější. LINQ je, podle mého názoru, největším plusem, se kterým přichází APS.NET 3.5

### **ASP.NET AJAX**

AJAX (Asynchronous JavaScript and XML) je vývojářská technika, která pracuje na principu skriptování na straně klienta. V praxi to funguje tak, že webová stránka s AJAXem vykoná asynchronní požadavek, ten odešle na server, který ho zpracuje a vrátí data zpět stránce (většinou to bývá kus kódu v XML). Kód klientské stránky obdrží tato data a provede se aktualizace pouze části stránky, které se požadavek týkal. V jednoduchosti lze říci, že webové prezentace s AJAXem zavolají server a aktualizují svůj obsah, aniž by se vše odesílalo zpět na server. I když je tento způsob tvorby webových stránek jednoduchý, dokáže s ní programátor vytvářet dynamické aplikace, které fungují plynuleji a opravdu nepřerušovaně. Bohužel, každá technologie má své úskalí a i když AJAX může někdy velice pomoci, najdou se situace, kde se jeho použití nedoporučuje a může uživatelům nebo aplikaci samotné spíše uškodit.

### **Silverlight**

Silverlight je zbrusu nová technologie, která je zcela nezávislá na .NET Frameworku a díky tomu může pracovat v různých webových prohlížečích a pod různými operačními systémy. Šířen je pomocí paginu, který si musí každý nainstalovat do prohlížeče, podobně jako Flash player. Kód Silverlightu je zpracováván na straně klienta a programátor s ním dokáže vytvořit poutavé a bohatě vybavené stránky, na které už ani kombinace HTML, DHTML a Javascriptu nestačí. Pokud využijete možností, které v sobě ukrývá, dokážete vytvořit poutavou dvourozměrnou grafiku, animované scény, přehrávat video, zvuk, vytvářet originální animované fotogalerie a mnohé další dynamické věci do webových stránek.

O Silverlightu by se dalo říci, že je podmnožinou WPF (Windows Presentation Foundation) a dříve jste ho mohli najít pod kódovým označením WPF/e (Windows Presentation Foundation Everywhere). S tím také souvisí tvorba v něm, protože i Silverlight využívá plně programovacího jazyka XAML (eXtensible Application Markup Language), který se vyznačuje nejen čistotou kódu, ale i jeho jednoduchostí a rychlostí při psaní. Velkou výhodou je také možnost oddělení designu od aplikační logiky. (v jednom souboru je design webové aplikace v XAMLu a ve druhém je logika napsána v jednom z .NET programovacích jazyků).



## **2.3 PHP**

I když sem PHP vůbec nepatří, nelze ho nezmínit jako hlavního konkurenta ASP.NET. PHP je jistě jednodušší programovací jazyk, snáze se učí a stačí trochu a máte naprogramovaný kus stránky. Rozšířen je také více a běhá bez problému na linuxovém serveru, nicméně pokud budete chtít vytvářet složitější internetové aplikace, ztratíte se u PHP v nepřehledném a zdlouhavém kódu. Jistě, u ASP.NET se budete muset toho více ze začátku naučit a není to tak jednoduché, ale vynaložené úsilí se Vám v pozdější fázi tvorby webu, jak sami poznáte, určitě vrátí.

## **2.4 Další verze**

ASP.NET 3.5 je zatím poslední vydanou verzí, ale Microsoft už připravuje nový .NET Framework 4.0, který přinese i novou verzi ASP.NET a nové vývojářské prostředí Visual Studio 2010. Hlavní důraz při tvorbě nového frameworku byl, aby programátoři napsali co nejméně kódu a tím i udělali méně chyb. Co všechno nám nová verze přinese, tak na to si budeme muset chvíli počkat, ale ze současných informací, které jsou již známé, lze jmenovat - nové uživatelské prostředí, nový programovací jazyk F#, doplňování kódů IntelliSense, lepší podpora databází Oracle a IBM DB2 a efektivnější tvorba webových aplikací s podporou technologie Silverlight.

### **3 Pagination**

Pagination, u nás známé pod pojmem stránkování, je hlavním tématem mé bakalářské práce. Mnozí uživatelé na internetu ho neznají, někteří neví, že se tak nazývá, hrstka z nich ví o čem je řeč, ale všichni ho plně využívají, aniž by věděli, že jim usnadňuje spousta práce a času při každodenním hledání na internetu.

#### **3.1 Vysvětlení pojmu**

Stránkování je způsob dělení dlouhých stránek na stejně dlouhé části podle určitých pravidel. Stránkování vzniklo primárně, jako dělení velice dlouhého textu na více stránek a listování v nich podobně, jak je tomu v knize. Později se tato technologie začala objevovat v dalších oblastech. Nejvíce se s ním v současnosti setkáte na webových stránkách typu e-shop nebo na diskusních fórech, kde stránky načítají z databáze stovky i tisíce záznamů a je nelogické, je všechny zobrazovat najednou.

Protože je hlavním cílem programátorů vytvářet uživatelsky přívětivé webové aplikace s intuitivním ovládáním pro všechny věkové kategorie, setkáváme se na internetu se stránkováním stále častěji. Nejznámějším příkladem jsou asi vyhledávače (seznam, google, centrum). Pagination tak zastává, vedle šikovné navigace na stránkách, neméně důležitou roli při vyhledávání.

### 3.2 Průzkum

Na internetu se vyskytuje spousta druhů stránkování, ale většina se liší především různými styly - barvou, typem fontu, rozmístěním prvků na stránce a jiné. Bohužel, začátečník i mírně pokročilý programátor je při tvorbě zmatený a neví, jaký druh použít, aby se zavděčil, co možná největšímu počtu návštěvníků stránek. S tímto by mu měla pomoci moje práce, jejíž hlavním cílem je, právě tuto mezeru zaplnit.

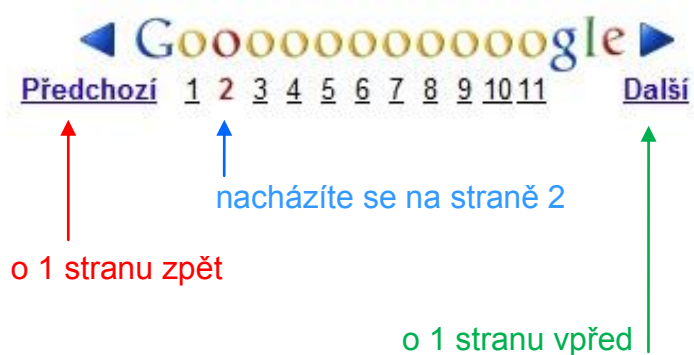
Pro tento účel jsem vytvořil v ASP.NET internetový dotazník ( <http://www.umelevne.cz/bakalarka/default.aspx> ), který byl předkládán řadě respondentů v různých věkových kategoriích. Každý uživatel měl možnost si vyzkoušet 5 různých druhů stránkování a následně je ohodnotit. Respondenti měli za úkol, nevnímat si jednotlivých stylů nebo-li vzhledů, ale především se soustředit na tato kritéria stránkování:

- funkce - jaké možnosti přeskokování uživatel má
- rychlost
- přehlednost
- umístění na stránce
- celkové hodnocení známkou od 1 do 5

Jednotlivé druhy stránkování v dotazníku s jednoduchým popisem najdete dále.

### 3.2.1 První stránkování

Na začátek jsem vybral jednoduché stránkování, které využívá nejrozšířenější vyhledávač - Google.



Stránkování se nachází na webu pouze dole uprostřed stránky a představuje jednoduché procházení stránkami po jedné stránce nebo skok na danou stránku, která je viditelná.

### ***Výsledky průzkumu***

**Průměrná známka:** 1,94

#### **Výhody:**

- rychlost
- jednoduchost
- kompromis
- přehlednost
- přesnost
- dobře se listuje
- viditelnost

#### **Nevýhody:**

- stránkování pouze dole
- není vidět počet stránek
- chybí pokročilé funkce
- chybí skok na konec
- zdlouhavé

### 3.2.2 Druhé stránkování

Druhé stránkování vychází ze stejného principu jako to první, ale je rozšířené o nové funkce, které pomáhají při vyhledávání u velkého počtu stránek.



Tento druh stránkování využívá jeden z největších internetových obchodů v České republice. Oproti prvnímu druhu, zde můžete navíc přeskakovat již po 10 stránkách najednou nebo skočit až na úplný konec, respektive začátek. To je velice efektivní, právě pro případ, když jsou produkty rozděleny na desítky, ne-li stovky stránek. Další nesmírnou výhodou je možnost filtrování obsahu podle kritérií - cena, název, datum a také možnost určit si, kolik položek se bude zobrazovat na jedné stránce (9, 18, 27, ... 99).

### ***Výsledky průzkumu***

**Průměrná známka: 2,34**

#### **Výhody:**

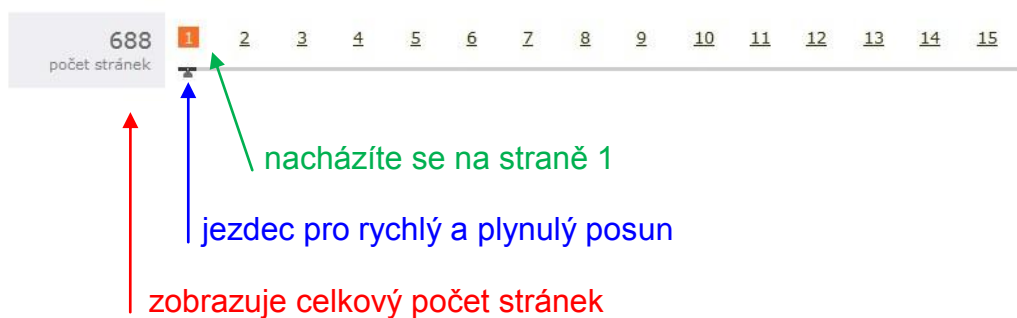
- rychlost
- přehlednost
- sofistikované
- skok na začátek a konec
- hodně funkcí
- dobré listování

#### **Nevýhody:**

- dlouho trvá dojet na konec
- nefunguje řazení
- pár lidí nepochopilo funkci šipek
- schází celkový počet stránek

### 3.2.3 Třetí stránkování

Třetí typ je dosti nestandardní a nachází se na zahraničních stránkách. Najdete ho "pouze" dole, což vzhledem k délce jednotlivých stránek nepovažuji za příliš vydařené.



Tento způsob dělení textu do stránek zobrazuje uživateli celkový počet stránek, na kolik se webová prezentace rozděluje. Podle mého názoru, je však největší výhodou posuvný jezdec, kterým si rychle nalistujete na stránky, o které máte zájem a pak jen stačí na jednu z 15 nalistovaných stránek kliknout.



### ***Výsledky průzkumu***

**Průměrná známka: 2,57**

#### **Výhody:**

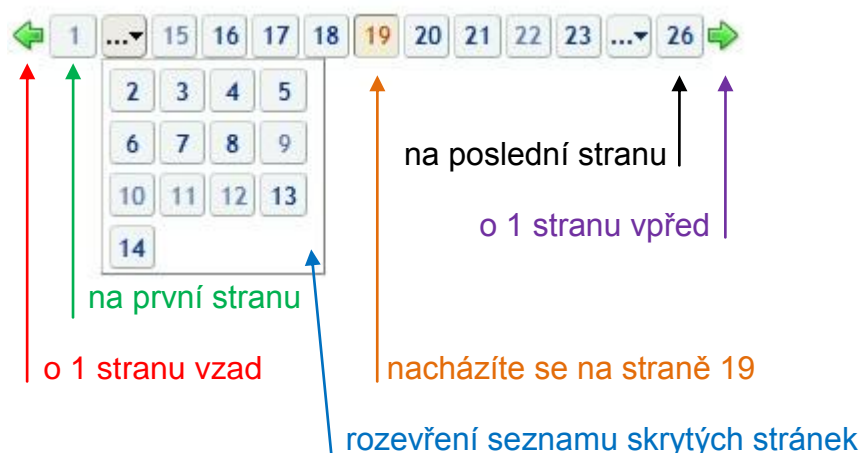
- jednoduchost
- rychlost
- přehlednost
- ukazatel celkového počtu stránek
- originální
- intuitivní posuvník

#### **Nevýhody:**

- nelze navolit počet záznamů
- nelze skočit na první a poslední stránku
- nejasné, na co se má kliknout
- chaotické

### 3.2.4 Čtvrté stránkování

Čtvrtý typ stránkování je také nestandardní řešení. Ovládací prvky však umístil programátor šikovně na začátek i na konec.



Tento typ kombinuje výhody klasického jednoduchého stránkování s novými funkcemi. Mezi ně patří takto označená  políčka, která Vám nabídnou náhledy čísel zbývajících stránek v mřížce s možností, na onu stránku kdykoliv přejít. Velkou výhodou, zprvu nepostřehnutelnou, jsou políčka  a , která v tomto případě ukazují první a poslední stránku a zůstávají neustále uživateli k dispozici a viditelná, ať se nachází na jakékoliv stránce.

### ***Výsledky průzkumu***

**Průměrná známka: 2,61**

#### **Výhody:**

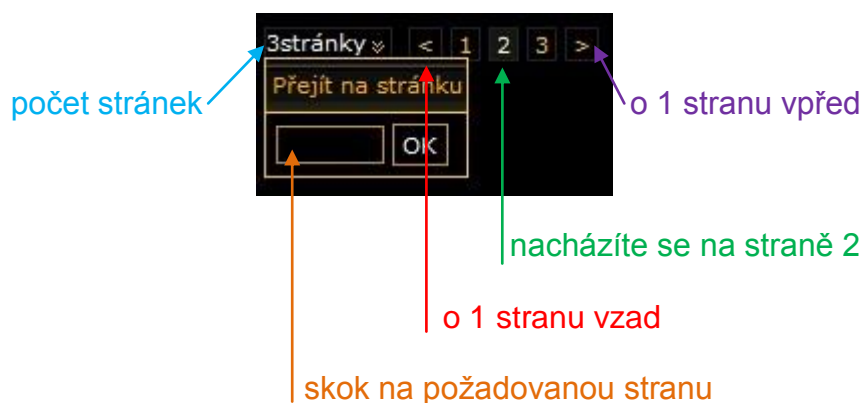
- jednoduchost
- přehlednost
- rychlost
- okamžitý výběr stránky
- nemusí se překlíkávat
- snadný přechod na první a poslední stránku

#### **Nevýhody:**

- moc tlačítek
- nepřehlednost
- pomalost
- nefungují některé stránky
- člověk loví jak v kalkulačce
- zabírá moc místa

### 3.2.5 Páté stránkování

Poslední typ stránkování je používán zahraničním diskusním fórem a také je najdete, jak na začátku stránky, tak i na konci.



Kromě klasického listování po stránkách o jednu stranu vpřed či vzad a zobrazení celkového počtu stránek, nabízí stránkování jednu unikátní funkci. Ta dává návštěvníkovi stránek možnost vložit přímo číslo stránky, na kterou chce vstoupit a po kliknutí se ihned zobrazí.

### ***Výsledky průzkumu***

**Průměrná známka: 3,12**

#### **Výhody:**

- možnost zvolit stránku
- jednoduchost
- přehlednost
- rychlost

#### **Nevýhody:**

- zadání stránky zbytečné
- nutnost použít klávesnici
- pomalé
- nepraktické
- nestandardní

### 3.2.6 Celkové hodnocení

Všech 5 druhů bylo otestováno pomocí výše zmiňovaného internetového dotazníku, který byl rozeslán několika desítkám respondentů, zveřejněn na diskusní fóra a další. Ti ho vyplnili a nadále rozesílali důvěryhodným osobám. Každé stránkování si mohli respondenti vyzkoušet bez časového limitu.

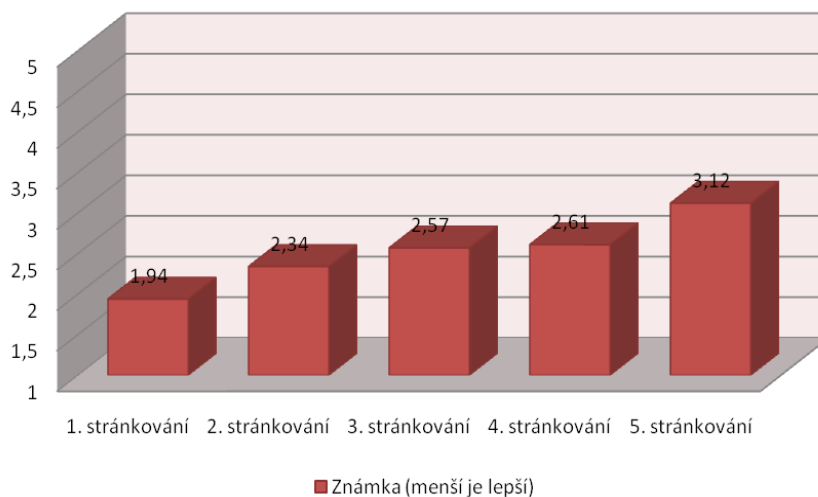
Hodnocení probíhalo ve 2 rovinách:

- známka 1 - 5 (menší je lepší)
- textový popis
  - jaké funkce se líbily
  - jaké funkce se nelíbily
  - doporučení, změna...

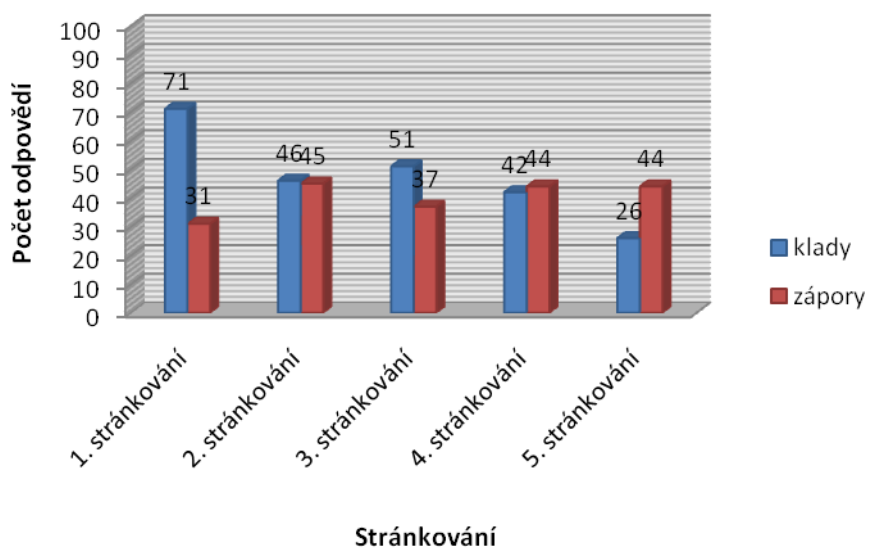
Všechny výsledky byly odeslány na můj email a následně vyhodnoceny. Podrobné výsledky najdete vždy u každého typu stránkování a porovnání Vám přiblíží následující grafy.

K hodnocení bych chtěl říci, že velkou úlohu také hraje to, pro jakou webovou prezentaci (e-shop, vyhledávač, diskusní fórum a jiné) je stránkování určeno. Každý by zvolil jistě jiné řešení. Průzkum měl ukázat, jak jsou lidé v současné době na stránkování zvyklí, jak jej používají a které funkce jim vyhovují a které ne. A to průzkum s úspěchem zjistil.

### Průměrné známky



### Četnost kladných a záporných funkcí



### **3.3 Jak na stránkování v praxi**

Jak prakticky vytvořit stránkování pomocí nějakého vývojového prostředí Vám ukáži nyní. Jako software jsem použil Visual Web Developer 2008 Express od společnosti Microsoft s programovacím jazykem C#.

Visual Studio nabízí programátorovi k dispozici hned několik předpřipravených datových ovládacích prvků, které mohou pomoci se stránkováním. Liší se však v mnoha směrech a každý se hodí pro jinou situaci. Na přiloženém CD najdete všechny vypracované příklady, které zde budu popisovat.

#### **3.3.1 GridView**

Tento ovládací prvek se nachází ve Visual Studiu již od verze ASP.NET 2.0 (ve verzi 1.0 to byl `DataGrid`) a přináší velice flexibilní možnosti nastavení - pro editaci, stránkování, řazení nebo editování. Data zobrazuje pomocí mřížky, která se dělí na řádky a sloupce. Díky zabudované podpoře pro stránkování je jeho vytvoření celkem jednoduché. Programátor má na výběr ze dvou způsobů tvorby stránkování, kde využije různé datové zdroje (`SqlDataSource`, `ObjectDataSource`).



### 3.3.1.1 Automatické stránkování

Nastavení automatického stránkování v `GridView` je velice jednoduché. Pro jeho aktivování stačí pouze povolit vlastnost `AllowPaging` na `true` a zvolit počet položek, kolik se bude zobrazovat na jedné stránce, pomocí vlastnosti `PageSize` (viz obr 3.3.1 - 1), defaultně je nastavena na hodnotu 10.

```
<asp:GridView ID="GridView1" runat="server"
              AllowPaging="true" PageSize="5">
  ...
</asp:GridView>
```

obr. 3.3.1 - 1

Datový zdroj pro `GridView` lze použít jakýkoliv, za předpokladu, že implementuje `ICollection`. Já jsem zvolil pro jednoduchost `SqlDataSource`, kde se pomocí průvodce jednoduše připojíte k databázi a provedete následný `SELECT`. Datový zdroj by měl vypadat podobně, jako na obrázku 3.3.1 - 2. Nyní už stačí si jen prohlédnout výsledek. viz obr 3.3.1 - 3, který je mírně upravený pomocí stylů.

```
<asp:SqlDataSource ID="SqlDataSource1" runat="server"
    ConnectionString="<%$
    ConnectionStrings:ConnectionString %>"
    SelectCommand="SELECT [jmeno], [prijmeni],
    [ulice], [mesto], [psc], [telefon], [email]
    FROM [JmenaDB] ORDER BY [prijmeni]">
</asp:SqlDataSource>
```

obr. 3.3.1 - 2

Příjmení	Jméno	Ulice	Město	PSC	Telefon	Email
Antoň	Štěpán	Branišovská 774	České Budějovice	55485	385556854	stepy@centrum.cz
Bosáková	Lenka	F.A.Brabce 5	Lišov	36658	606336585	boss@email.cz
Černý	Vit'a	Květinová 3	Kostelec u Jihlavy	58856	608365463	mujmail@gmail.com
Dvořáková	Jana	Trojská 18	Jindřichův Hradec	35845	737564582	dvorak@gmail.com
Horálek	Jakub	Benešovská 66	Velký Pěčín	32545	777658565	ford@seznam.cz
						1 2 3 4 5

obr. 3.3.1 - 3

Nevýhodou automatického stránkování je obrovská databázová zátěž. Pokud je totiž stránka rozdělena na např. 10 stránek, tak při jejich procházení se pokaždé načítají všechna data ze všech 10 stránek. Jediné zmírnění této zátěže spočívá v používání ukládání do cache paměti serveru. Bohužel, toto se hodí pouze pro méně rozsáhlé dotazy, protože cache paměť využívá operační paměti serveru, která není nekonečná.

### 3.3.1.2 Vlastní stránkování (custom paging)

Vlastní stránkování je sice složitější než automatické, ale zajistí, že `GridView` bude pracovat pouze se stránkou, která je vybraná uživatelem a nebude zahlcovat tolik paměť serveru. Jako datový zdroj podporuje pouze `ObjectDataSource`, který se musí nastavit.

Abyste mohli se stránkováním pracovat, je nutné nejprve povolit `EnabledPaging` na `true`. Stránkování se pak již implementuje pomocí těchto 3 vlastností:

- `SelectCountMethod`
- `StartRowIndexParameterName`
- `MaximumRowsParameterName`.

`GridView` u automatického stránkování dokázal spočítat počet záznamů v databázi sám, zde musíme vytvořit metodu `CountNames()` nacházející se ve třídě `NameDatabase`, která to vypočítá za něj. Tu pak přiřadíme právě oné vlastnosti `SelectCountMethod` (viz obr 3.3.1 - 4)

```
<asp:ObjectDataSource ID="ObjectDataSource1"
    runat="server" SelectCountMethod="CountNames"
    EnablePaging="True" ... >
</asp:ObjectDataSource>
```

obr. 3.3.1 - 4

Pro efektivnější chod tohoto stránkování, aby se nemusely ukládat všechny záznamy z tabulky, ale pouze ty, které jsou na aktuální stránce, jsem vytvořil proceduru (viz obr 3.3.1 - 5). Ta vytvoří přechodnou tabulku, do které se načtou všechna jména. Při otevření určité stránky se pomocí parametrů vyberou ze přechodné tabulky pouze ty záznamy, které mají být viditelné.

Naposled vytvoříme metodu `GetNames()` (viz obr. 3.3.1 - 6), která pracuje s výše zmiňovanými vlastnostmi datového zdroje `ObjectDataSource` (`MaximumRowsParameterName` a `StartRowIndexParameterName`), jako parametry. Ty jsou defaultně nastavené na hodnoty `startRowIndex` a `maximumRows` a vytáhnou si potřebné informace z procedury z přechodné tabulky. Pokud si příklad spustíte, bude `GridView` vypadat stejně, jako u automatického stránkování, jen bude pracovat efektivněji.

```

ALTER PROCEDURE dbo.GetNamePage

    @Start int, @Count int

AS
CREATE TABLE #TempName
(
    ID int IDENTITY PRIMARY KEY,
    jmeno nvarchar(50),
)

INSERT INTO #TempName
(
    jmeno
)
SELECT jmeno FROM Prvni ORDER BY jmeno ASC

DECLARE @FromID int
DECLARE @ToID int

SET @FromID = @Start
SET @ToID = @Start + @Count - 1

SELECT * FROM #TempName

WHERE ID >= @FromID AND ID <= @ToID
    
```

obr. 3.3.1 - 5

```

public List<NameDetails> GetNames(int startRowIndex, int
maximumRows)
{
    SqlConnection conn = new
    SqlConnection(connectionString);
    SqlCommand cmd = new SqlCommand("GetNamePage",
conn);
    cmd.CommandType = CommandType.StoredProcedure;
    cmd.Parameters.Add(new SqlParameter("@Start",
SqlDbType.Int, 4));
    cmd.Parameters["@Start"].Value = startRowIndex +
1;
    cmd.Parameters.Add(new SqlParameter("@Count",
    
```

```
SqlDbType.Int, 4));  
cmd.Parameters["@Count"].Value = maximumRows;  
  
List<NameDetails> namesList = new  
List<NameDetails>();  
  
try  
{  
    conn.Open();  
    SqlDataReader reader = cmd.ExecuteReader();  
  
    while (reader.Read())  
    {  
        NameDetails nam = new  
        NameDetails((string) reader["jmeno"]);  
        namesList.Add(nam);  
    }  
    reader.Close();  
    return namesList;  
}  
catch (SqlException ex)  
{  
    throw new ApplicationException("Chyba v načítání  
dat.");  
}  
finally  
{ conn.Close(); }  
}
```

obr. 3.3.1 - 6

Výběr stránkování není jednoduchý a zpravidla to závisí na konkrétní situaci, ale jako nepsané pravidlo mohu říci, že pokud `GridView` zobrazuje velké množství dat, je lepší použít vlastní stránkování, na druhou stranu, pokud je Váš databázový server pomalejší, bude lepší ukládat data do cache paměti serveru a použít tak automatické stránkování.

### 3.3.2 ListView

Tento datový ovládací prvek patří také mezi novinky ASP.NET verze 3.5. Jeho cílem je nahradit dosud používaný Repeater, který tu je již od verze ASP.NET 1.0. ListView nabízí také celou škálu možností editací. Na rozdíl od předchozího GridView zde však nenajdeme sloupce, ale pouze řádky, jejichž tvorba a úpravy jsou založené na velmi sofistikovaném používání rozsáhlé sady šablon (templates). Obecně mohou říci, že ListView se hodí hlavně na zobrazení mnoha řádků, kde každý řádek má více různorodých prvků.

Pokud budeme chtít zobrazit data pomocí tohoto ovládacího prvku, musíme připojit datový zdroj (já použil SqlDataSource), vytvořit šablonu LayoutTemplate zobrazující obecnou strukturu a pak šablonu ItemTemplate, která je jakýmsi obsahem prvků (viz obr. 3.3.2 - 1).

```
<asp:ListView ID="ListView1" runat="server"
    DataSourceID="SqlDataSource1">
    <LayoutTemplate>
        <span id="itemPlaceholder" runat="server"></span>
    </LayoutTemplate>

    <ItemTemplate>
        <%# Eval("jmeno") %>
    </ItemTemplate>
</asp:ListView>
```

obr. 3.3.2 - 1

Ačkoliv je `ListView` nový ovládací prvek, neobsahuje možnost sám o sobě stránkovat data. K tomuto účelu byl vytvořen nový ovládací prvek `DataPager` (ASP.NET 3.5), který poskytuje stránkování pro více ovládacích prvků, ale v současné době jej zatím podporuje právě jen `ListView`. `DataPager` se vkládá do šablony `LayoutTemplate` spolu s elementem `Fields`, který slouží k formátování stránkování. Na obrázcích 3.3.2 - 2 a 3.3.2 - 3 ukazují jednodušší a složitější způsob.

```
<LayoutTemplate>
...
  <asp:DataPager ID="DataPager1" runat="server"
    PageSize="5">
    <Fields>
      <asp:NumericPagerField />
    </Fields>
  </asp:DataPager>
</LayoutTemplate>
```

1 2 3 4 5

obr. 3.3.2 - 2 - jednodušší

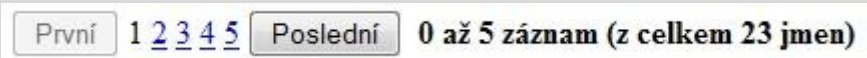
```
<Fields>
  <asp:NextPreviousPagerField ButtonType="Button"
    ShowFirstPageButton="True" ShowNextPageButton="False"
    ShowPreviousPageButton="False" />
  <asp:NumericPagerField />
  <asp:NextPreviousPagerField ButtonType="Button"
    ShowLastPageButton="True" ShowNextPageButton="False"
    ShowPreviousPageButton="False" />
```



```

<asp:TemplatePagerField>
  <PagerTemplate><b>
    <asp:Label runat="server" ID="CurrentPageLabel"
      Text="<%# Container.StartRowIndex %>" />
    až
    <asp:Label runat="server" ID="TotalPagesLabel"
      Text="<%#
      Container.StartRowIndex+Container.PageSize
      %>" />
    záznam (z celkem
    <asp:Label runat="server" ID="TotalItemsLabel"
      Text="<%# Container.TotalRowCount %>" />
      jmen)
    <br /></b>
  </PagerTemplate>
</asp:TemplatePagerField>
</Fields>

```



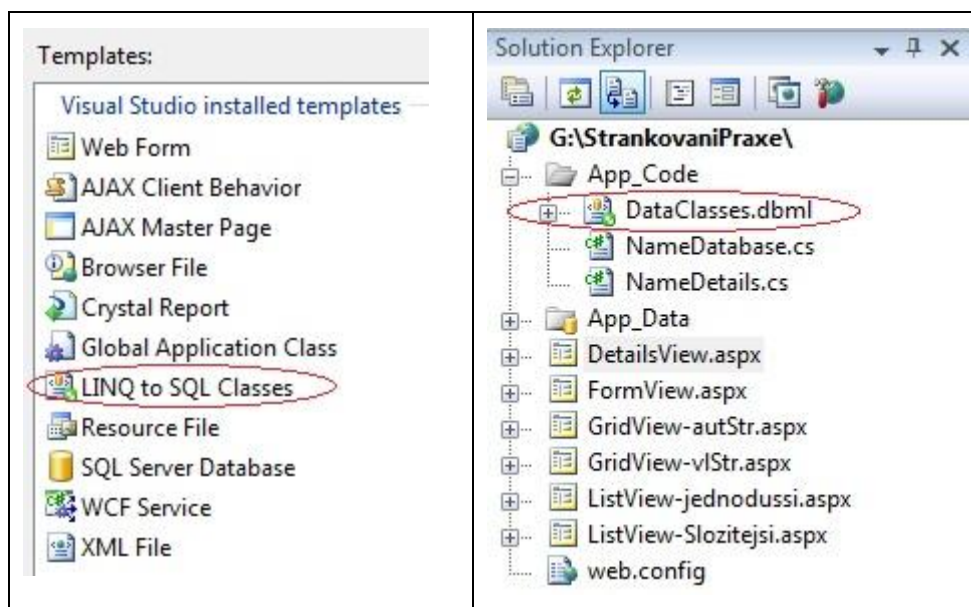
obr. 3.3.2 - 3 - složitější

Nejen ovládací prvek `ListView`, ale hlavně `DataPager` má jistě velice slibnou budoucnost a skrývá se v něm obrovský potenciál pro tvorbu různých druhů stránkování. Nehledě na to, až přibudou další ovládací prvky, které ho budou podporovat.

Nyní se podíváme na dva velmi podobné datové ovládací prvky `DetailsView` a `FormView`. Oba mají společný pohled vždy na jeden jediný záznam s možností stránkování mezi nimi. Jejich malé rozdíly si probereme nyní.

### 3.3.3 DetailsView

Tento ovládací prvek zobrazuje záznamy po jednom uvnitř tabulky. Jako datový zdroj si můžete vybrat, který se Vám hodí (SqlDataSource, ObjectDataSource, LinqDataSource). Já jsem si tentokrát vybral LinqDataSource, který jsem ještě nepoužíval a také z důvodu, že je to novinka .NET Frameworku 3.5. Pro tento účel je nutné přidat **LINQ to SQL Classes**, která se vloží do složky **App\_Code** v Solution Exploreru. (viz obr. 3.3.3 - 1). Na ní již pak stačí přetáhnout vytvořené tabulky z databáze ze Server Exploreru a databáze je připravená k připojení k datovému zdroji (viz obr. 3.3.3 - 2).



obr. 3.3.3 - 1

```
<asp:LinqDataSource ID="LinqDataSource1" runat="server"
    ContextTypeName="DataClassesDataContext"
    GroupBy="jmeno" OrderGroupsBy="key"
    Select="new (key as jmeno, it as JmenaDBs)"
    TableName="JmenaDBs">
</asp:LinqDataSource>
```

obr. 3.3.3 - 2

Mezi záznamy lze procházet opět pomocí stránkování, které se aktivuje pomocí vlastnosti `AllowPaging`, kterou nastavíme na hodnotu `true` (viz obr. 3.3.3 - 3). Další konfigurace pak probíhá pomocí vlastností `PagingStyle` a `PagingSettings`, jako u `GridView`.

```
<asp:DetailsView ID="DetailsView1" runat="server"
    DataSourceID="LinqDataSource1"
    AllowPaging="True">
</asp:DetailsView>
```

obr. 3.3.3 - 3

Rozdíl oproti `GridView` je v tom, že nemáte možnost použít vlastní stránkování a datový objekt se načítá pokaždé celý. Mezi klady mohou naproti tomu uvést, že pokud pracujete v editačním režimu jednoho záznamu a přejdete na stránku jinou, `DetailsView` zůstává defaultně v editačním režimu. Zabránit se tomu samozřejmě dá zavoláním metody `ChangeMode()`.

### 3.3.4 FormView

FormView je také ovládací prvek zobrazující pouze jediný záznam. Oproti DetailsView však nepoužívá tabulku, ale je založen na šablonách a přímo je vyžaduje. Pokud bych ho měl srovnat s GridView a DetailsView, tak všechny 3 prvky podporují vkládání, čtení a editaci. FormView se od nich pouze odlišuje tím, že nepodporuje třídu CommandFile(), která vytváří tlačítka pro editaci. Zde si je musíte vytvořit sami.

Co se týká stránkování, tak jeho podpora je samozřejmostí. Opět stačí povolit AllowPaging (viz obr. 3.3.4 - 1) zvolit jeden z datových zdrojů a můžeme směle stránkovat po jednotlivých záznamech.

```
<asp:FormView ID="FormView2" runat="server"
    AllowPaging="True"
    DataSourceID="SqlDataSource1">
</asp:FormView>
```

obr. 3.3.4 - 1

## 4 Závěr

Úkolem mé bakalářské práce bylo seznámit se s .NET Frameworkem 3.5, hlavně s jeho novinkami, provést srovnání různých druhů stránkování pomocí průzkumu a provést vyhodnocení. Dále jsem se měl seznámit s tvorbou stránkování v praxi pomocí různých metod ve vývojovém prostředí Microsoft Visual Studio. Tyto úkoly jsem se snažil splnit v plné rozsahu.

Vytvořil jsem webové stránky, kde jsem vybral 5 různých druhů stránkování z hlediska jejich funkcí, které byly předloženy respondentům různých věkových kategorií. Ty si je vyzkoušeli a zhodnotili, jak známku, tak slovním hodnocením. Výsledky jsem zpracoval v programu Excel a předložil ve své práci.

Dále jsem vytvořil projekt se sadou příkladů, kde ukazují, jak vytvořit stránkování pomocí různých ovládacích prvků a datových zdrojů a upozorňuji v nich na výhody, nevýhody a různá úskalí této technologie. Všechny příklady popisované v této práci jsou na přiloženém CD.

Stránkování (Pagination), i když se zdá nepatrnou součástí webových prezentací, dostává se, podle mého názoru, do čím dál většího povědomí běžných uživatelů a programátoři jej využívají se stále větší oblibou.

## 5 Přehled použité literatury

- [1] MACDONALD, Matthew, SZPUSZTA, Mario. *ASP.NET 3.5 a C# 2008 : tvorba dynamických stránek profesionálně*. Brno : Zoner Press, 2008. 1584 s. ISBN 978-80-7413-008-3.
- [2] *.NET Framework Developer Center* [online]. 2008 [cit. 2009-01-25]. Dostupný z WWW: <<http://msdn.microsoft.com/en-us/netframework/default.aspx>>.
- [3] *.NET Framework 3.0 a 3.5, ASP.NET* [online]. 2002-2008 [cit. 2009-01-25]. Dostupný z WWW: <<http://www.aspnet.sk/Novinky-v-.NET-Framework-3.5-a-C-3.0-VIDEO-122.aspx>>.
- [4] RŮŽIČKA, Pavel. *Pagination pro formuláře ASP.NET* [online]. 2007 [cit. 2009-01-25]. Dostupný z WWW: <<http://interval.cz/clanky/strankovaci-prvek-pagination-pro-formulare-asp-net-pouziti/>>.
- [5] *Stránkování ASP.NET* [online]. 2007 [cit. 2008-12-10]. Dostupný z WWW: <<http://blog.jakubmaly.cz/programovani/csharp-aspnet/strankovani-v-asp-net-a-ms-sql-serveru-univerzalne.aspx>>.

- [6] MITCHELL, Scott. *An Overview of ASP.NET 3.5 and Visual Studio 2008* [online]. 2007 [cit. 2009-01-20]. Dostupný z WWW: <<http://aspnet.4guysfromrolla.com/articles/112107-1.aspx>>.
- [7] BAŠTÝŘ, Jan. *Programování v ASP.NET* [online]. 2007 [cit. 2009-12-25]. Dostupný z WWW: <<http://nidzo.cz/programovani/aspnet/programovani-v-asp-net-referat.aspx>>.
- [8] VALÁŠEK , Michal. *Visual Studio 2008 je venku a co z toho vyplývá* [online]. 2007 [cit. 2009-12-28]. Dostupný z WWW: <<http://www.aspnet.cz/Articles/173-visual-studio-2008-je-venku-a-co-z-toho-vyplyva.aspx>>.
- [9] JANOŠÍK, Dušan. *Něco málo k technologiím WPF a Silverlight* [online]. 2007 , 2007 [cit. 2010-01-02]. Dostupný z WWW: <<http://www.vyvojar.cz/Articles/473-neco-malo-k-technologieim-wpf-a-silverlight.aspx>>.
- [10] *LinqDataSource with ASP.NET data controls* [online]. 2008 [cit. 2010-01-03]. Dostupný z WWW: <<http://www.codeproject.com/KB/aspnet/LinqDataSourcebasics.aspx>>.

## **6 Seznam příloh**

- [1] Na přiloženém CD najdete veškeré zdrojové kódy a příklady popisované v této práci.