

Jihočeská univerzita
Pedagogická fakulta
Katedra Fyziky

Zpracování dat s využitím WWW technologií
bakalářská práce

Autor: **Milan Sýkora**
Vedoucí bakalářské práce: **Ing. Michal Šerý**

České Budějovice 2008

Anotace

Tato bakalářská práce se zabývá popisem datových zdrojů na internetu, způsoby a možnostmi jejich zpracování. Její součástí je praktická ukázka, která pomocí programového vybavení PHP ukazuje některé možnosti zpracování dat na běžném PC s operačním systémem Windows XP. Tento počítač může zároveň fungovat jako server, s přístupem ke staženým datům přes webovské prohlížeče. Celá práce, spolu se skripty programu se nacházejí na přiloženém CD.

Synopsis

This bachelor's thesis is aimed at the description a data sources on the internet. The content shows the different way and the possibilities of processing the dates. The specific project made use of the platform PHP, installed on PC with the operating system Windows XP, took advantage of this processing standards. That PC can be used as a webserver. All the program files that create virtual directory of the application can be found on the enclosed CD.

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 práce, a to v nezkrácené podobě elektronickou cestou ve veřejně přístupné části databáze STAG provozované Jihočeskou univerzitou v Českých Budějovicích na jejích internetových stránkách.

24.11.2008

Děkuji vedoucímu mé bakalářské práce Ing. Michalovi Šerému za odborné vedení, cenné rady a připomínky k vypracování mé práce.

ÚVOD	8
1. DATOVÉ ZDROJE	9
1.1 Databáze	9
1.1.1 Souborové a systémové databáze	9
1.1.2 Systémové databáze	9
1.1.3 Objektové a relační databáze.....	10
1.1.4 Relační databáze.....	10
1.2 WWW zdroje	11
1.2.1 Vyjádření struktury	11
1.2.2 Metadata	11
1.2.3 Vyjádření sémantik - RDF	12
1.2.4 Rozšiřitelnost, kompatibilita	12
1.2.5 Přejchod k RDF	12
2. ZPŮSOBY ZPRACOVÁNÍ A UKLÁDÁNÍ DAT	13
2.1 Jak zpracovat data	13
2.1.1 Historie přístupu a tvorby WWW	13
2.2 Rozdělení podle umístění skriptu	14
2.2.1 Klientské WWW technologie	14
2.2.2 Skript na serveru	14
2.3 Přístup k datům - Datová rozhraní	14
2.3.1 DAO	15
2.3.2 ADO	15
2.3.3 RDO	15
2.3.4 ODBC (Open Database Connectivity).....	16
2.4 Java script.....	16
2.5 ASP	17
2.5.1 Popis ASP	17
2.5.2 Program.....	18
2.5.3 Objekty ASP	18
2.5.4 Komponenty ASP	18
2.6 PHP- „Professional Home Pages“	19
2.6.1 Popis PHP.....	19
2.6.2 Datové typy PHP	19
2.6.3 Podpora databází.....	19
2.6.4 Objektové programování	19
2.6.5 Další funkce	19
2.6.6 Zpracování souborů s daty.....	20
2.6.7 Další možnosti PHP pro práci se soubory	20
2.6.8 Parsery – čtení dokumentu XML.....	21
2.6.9 Výstup – nahrání souborů na server	21
2.6.10 Výstup – HTML stránka	21
2.6.11 Výstup – PDF soubor.....	21
2.6.12 Výstup – XML soubor.....	22
2.6.13 Výstup – Export do souboru z databáze	22
2.6.14 Výstup – Graf.....	23
2.7 XAML	24
2.8 Google Office.....	25
2.8.1 Služby.....	25
2.8.2 Výhody práce s daty na serveru	25
2.8.3 Nevýhody Google Office	26
2.8.4 Google Office - zhodnocení	26

3. PREZENTACE DAT	27
3.1 Logické formátování dat - XML	27
3.1.1 DTD	27
3.1.2 Struktura XML dokumentu	28
3.1.3 Využití XML	29
3.1.4 Složení XML dokumentu	30
3.2 XHTML	311
3.3 Grafické formátování stránky	31
3.3.1 CSS	31
3.3.2 XSL	31
3.4 Aktuální data- RSS	32
3.4.1 Využití	32
3.4.2 Struktura RSS dokumentu	32
3.4.3 Čtečky RSS dokumentu	33
3.5 Progres nových multiplikačních vývojových prostředí	34
3.6 Technologie Microsoft zpracování dat a grafiky	34
4. UKÁZKOVÁ APLIKACE	39
4.1 Zadání	39
4.1.1 Požadavky zadání	39
4.1.2 Rozbor	39
4.2 Řešení	40
4.3 Programové vybavení a nastavení	40
4.3.1 PHP 5.2.0	41
4.3.2 Nastavení PHP	41
4.3.3 Web server Apache 2.2.3	41
4.3.4 Nastavení Apache	41
4.3.5 Databáze MySql 5.0.27	41
4.3.6 Nastavení MySql	41
4.3.7 PhpMyAdmin 2.9.1.1	41
4.3.8 SQLiteManager 1.2.0	41
4.3.9 Další pomocný freeware - HeidiSQL	42
4.3.10 JavaScript	42
4.4 Perioda stahování dat	42
4.5 Tvorba aplikace	42
4.5.1 Databáze meteo	42
4.5.2 Tabulky v databázi	43
4.5.3 Tabulka Meteo_url	43
4.5.4 Tabulka Meteo_zdroje	43
4.5.5 Tabulka Meteo_data	44
4.5.6 Diagram struktury databáze	44
4.6 Popis souborů aplikace	45
4.6.1 Config.php	45
4.6.2 Soubor stazeni_dat.php	45
4.6.3 Soubor add1.php	51
4.6.4 Soubor insert.php	55
4.6.5 Soubor del1.php	57
4.6.6 Soubor delete.php	59
4.6.7 Soubor edit_url1.php	61
4.6.8 Soubor edit_url2.php	63
4.6.9 Soubor edit_zdroj1.php	64
4.6.10 Soubor edit_zdroj2.php	66
4.6.11 Soubor konkretni_dotaz.php	68
4.6.12 Soubor zobraz_csv.php	72
4.6.13 Soubor zobraz_web.php	73
4.6.14 Soubor zobraz_xml.php	74
4.6.15 Soubor zobraz_data.php	75
4.6.16 Soubor help.html	77
5. ZHODNOCENÍ	79

SEZNAM POUŽITÉ LITERATURY	80
ZDROJE Z INTERNETU	80
PŘÍLOHA.....	80

Úvod

Aktuální informace, data a přístup k nim jsou v současnosti tou nejdražší devizou každého subjektu, jenž působí na rozsáhlém poli IT i v ostatních oborech. Rozvoj technologií v oblasti zpracování dat se neustále posunuje kupředu. Vývoj však správně od jisté doby podléhá celosvětovým standardům, které zajišťují kompatibilitu starého s novým a východu se západem. Hierarchicky nadřazené standardy spojují oba světy a co se vymyká, nemá už před sebou dlouhou dobu života. Vytvoření vlastní cesty je většinou obtížné a opovrhované a i se skvělým řešením to nějakou dobu trvá, než si prorazí svou cestu nahoru. Lokální, zabezpečené sítě i celosvětová síť se od dob svého vzniku neustále vyvíjí. WWW. World Wide Web je vedle elektronické pošty, pevných médiích jako další z alternativ přenosu souborů a informací nám již od počátku vyvstává otázka jak efektivněji, snadněji, s maximální úsporou času zpracovat data. K zápisu statických stránek-hypertextových dokumentů slouží jazyky HTML, XML. Statickou stránku řídí logický mechanismus logické vrstvy kódu, který zabezpečí jejich změnu, aktualizaci.

1.Datové zdroje

1.1 Databáze

1.1.1 Souborové a systémové databáze

- databáze která je obsažena v jednom nebo několika málo souborech operačního systému
- databázi přesuneme na jiný PC pouhým překopírováním souboru nebo několika souborů
- zástupcem souborových databází jsou databáze dbf (foxpro), SQLite nebo třeba Microsoft Access
- jejich dostupnost v rámci sítě
- různá úroveň podpory souběžné práce více uživatelů
- k práci se souborovou databází je potřeba mít nějaká práva.

1.1.2 Systémové databáze

- slouží jako databázové servery
- mají velmi dobrou podporu souběžné práce více uživatelů
- bývají přístupné pomocí nějakého směrovatelného protokolu (asi TCP)
- mívají obecně složitější instalaci, mívají lepší strukturu a robustnější
- například MySQL, PostgreSQL, Oracle nebo Microsoft SQL server
- můžete se připojit k systémové databázi i v případě, že na systém na němž běží nemáte jinak přístup.

V poslední době se rozdíl mezi souborovými a systémovými databázemi ztrácí. Existuje řada způsobů, jak pomocí software přistupovat k souborovým databázím, jako by to byly databáze systémové. Například pomocí UnixODBC nebo ODBC můžete přistupovat k databázi dbf z počítačové sítě a podobně.

Rozdělení podle filozofie uložených dat v databázi

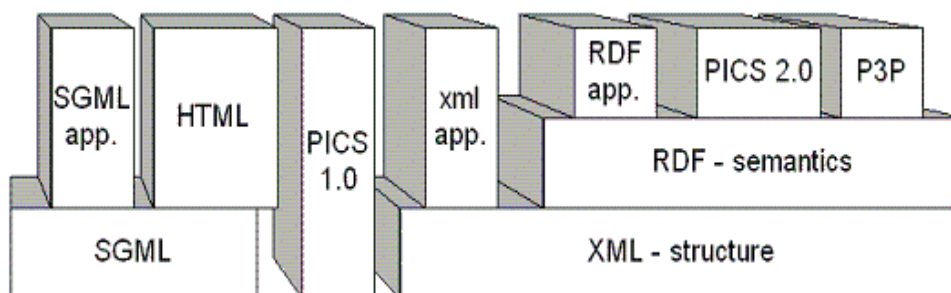
1.1.3 Objektové a relační databáze

- způsob je organizovat data v databázi jako objekty
- objektové databáze nejsou příliš rozšířené
- jsou pomalé.
- zástupcem např. Caché

1.1.4 Relační databáze

- všechny dnešní databáze
- data jsou (nebo alespoň mohou být) uložena tak, že související záznamy můžete jednoduše vyhledat
- relace = vazby mezi jednotlivými tabulkami(tabulkou1 a tabulkou2)

1.2 WWW zdroje



Rozložení aplikací, používaných pro zpracování www stránek

1.2.1 Vyjádření struktury

Standard **SGML** dává obecný způsob vyjádření struktury dat aplikacím zpracovávajícím text .

HyperText Markup Language (**HTML**) je konkrétní aplikace dána jedním DTD (viz. níže) vytvořeným z SGML, používaný k definování a formátování dat v síti WWW.

ISO(International Organization for Standardization) je mezinárodní organizace pro standardizaci. Stanovuje průmyslové standardy

XML nahrazuje SGML a umožňuje vyjádření struktury; RDF umožňuje vyjádření sémantik.

Nový rozšiřitelný značkovací jazyk W3C (**XML**) poskytuje stejnou funkci jako SGML jednodušším a silnějším způsobem. Budoucí textové značkování z W3C se bude směřovat na XML spíše než na SGML. Tento způsob zpracování webu může být aplikován i na budoucí verze HTML, ale vše závisí na technikách práce na zpětné kompatibilitě a strategii přechodu.

1.2.2 Metadata

Web je stále více a více používán pro strojům srozumitelný obsah. Příchod XML je pro tento účel správně načasován a poskytuje obecnou syntaxi pro velké množství nových datových formátů.

Mnoho z těchto aplikací je více než jen strukturovanými daty. Data reprezentují strojům srozumitelné tvrzení o objektech na webu mimo web. Tyto „Metadata“ budou umožňovat umístit obrovské množství informací a už existujících aplikací na Web nejen pro lidské „surfování“, ale i pro porozumění strojům: vyhledávání, zdůvodnění (reasoning) a analyzování.

1.2.3 Vyjádření sémantik - RDF

RDF „Resource Description Framework“ je sadou specifikací, které umožňují metadatovým aplikacím, aby byly kombinovány a spolupracovali stejným způsobem při vyjádření sémantik, které sdílí. RDF je další vrstvou na vrcholu s XML.

V současné době existuje pouze velký průmysl produkující aplikace, které umisťují informace z odkazových informačních systémů (legacy information system) na web. Existuje také IT průmysl, který designuje aplikace pro surfování na webu. Jsou vytvářeny se strukturou, tak aby mohly být webové stránky automaticky generovány, získávány a překonvertovány do dobře definovaných strojům srozumitelných dat. RDF umožní, aby tato dlouhá cesta byla krátký okružem a umožní programům získat data přímo.

1.2.4 Rozšiřitelnost, kompatibilita

S přínosem zkušeností tvoří RDF velký krok vpřed proti specifikacím jako HTML nebo HTTP. Tyto specifikace byly rozšířeny během jejich evoluce. Jedním z rozvíjejících procesů bylo i experimentální přidávání tagů. Význam nebo důležitost těchto nových tagů pro software nebyl nikdy evidentní, během experimentů se neobjevilo nebezpečí vážné nekompatibility.

RDF má za cíl umožnit dokumentům, aby byly zapsány v směsi starých standardů a nových specifických experimentálních nebo proprietálních zdrojů dat, ale s dobře definovaným způsobem, jak vědět, co je důležitého, co může být ignorováno a tak, aby starý software uměl dedukovat, nebo stáhnout a rozumět novému. Výsledek bude umožňovat kombinaci aplikací, například tvorbu dokumentů, tak že se zkombinují pomocí dobře definovaného konceptu bankovní, inženýrské a právníkové zdroje. Síla Webu jako vyjadřujícího média se stane produktem (spíše než souhrnem) individuálních vývoje.

RDF bude dávat jasnější a rychlejší cestou vývoje pro nové nápady bez nutnosti nějakých ojedinělých řešení

1.2.5 Přejít k RDF

Plánem pro existující aplikace W3C jako např. PICS 1.0, napsaným před vznikem XML a RDF, je vytvořit přechod k verzi 2.0, definované v termínech RDF a tedy v XML. Toto umožní kombinovat popisky PICS s informacemi o soukromí (například) z projektu P3P – Platform for Privacy Preferences, který bude umožňovat užití RDF. P3P určuje jak mohou být informace využívány ke shromažďování údajů o uživateli. Mnoho nových aplikací (jako P3P) bude přímo postaveno na RDF. Konsorcium W3C podporuje existující aplikace při vytváření přechodu k RDF tak, aby se využilo této rozšiřitelnosti, použitelnosti a síly kombinace s ostatními aplikacemi.

2.Způsoby zpracování a ukládání dat

2.1 Jak zpracovat data

Získávání dat obecně mezi jakýmkoliv datovými zdroji a cíli, nebo v jiných vývojových prostředích většinou probíhá přes vytvoření ActiveX objektů. Tyto objekty potom musí být v daném prostředí podporovány. V současné době proto lze data z Internetu, či jiných datových zdrojů zpracovat pomocí speciálních nástrojů určených výhradně na web, nebo v obecných vývojových prostředích. Patří sem např. produkty Microsoft InterDev, Visual Basic, Visual C++, nebo Visual J++ a další. My se budeme věnovat specifické skupině zpracování asi největšího balíku – z celosvětové WWW síť, a to pomocí nástrojů, jejichž seznam a částečně i specifikace některých je popsána níže.

2.1.1 Historie přístupu a tvorby WWW

V počátcích od roku 1990 se spuštěním World-Wide Web postačili k existenci stránek pouhé 3 technologie

HTML(Hypertext Markup Language)	- zápis stránek
HTTP(Hyper Text Transfer protocol)	- přenos z www serveru do prohlížeče
URL(Uniform Ressource Locator)	- odkaz, jedinečná URL adresa každého objektu

Automatické generování stránek s informacemi proměnlivými v čase vyvolalo potřebu rozhraní, které definuje způsob spuštění programu a předávání dat mezi WWW-serverem a programem. Takovým rozhraním je CGI(Common Gateway Interface). Příkladem další fáze CGI jsou vyhledávací servery, kde uživatel zadá do formuláře, co chce na internetu vyhledat, odešle serveru, kde skript prohledá indexy. Výsledkem je stránka, která obsahuje všechny vyhledané odkazy. Odezva CGI skriptů byla však pomalá a tak vznikl jazyk Java, posléze JavaScript, kde výhodou byla nezávislost na platformě.

Úspěch Javy podnítil její přesunutí na stranu serveru SSJS(Server Side JavaScript). Obdobou Microsoft byly potom ASP, kde bylo možné použít k programování VBScript.

2.2 Rozdělení podle umístění skriptu

2.2.1 Klientské WWW technologie

Klientské technologie

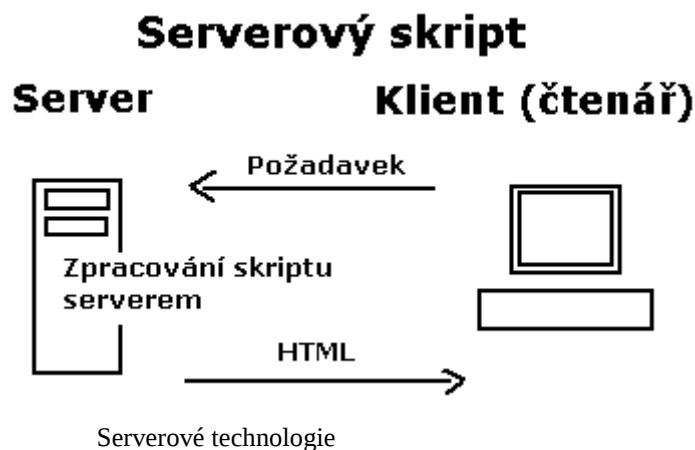
- JavaScript
- Java-aplety
- Dynamické HTML

Klientské technologie – spolu s HTML stránkou je prohlížeči odeslána část programového kódu, která se spustí (provede co má) na cílovém počítači (prohlížeči). Událost může být vázána na najetí, kliknutí myši na odkaz, otevření okna prohlížeče... Prohlížeč na cílovém počítači, ale musí znát programovací jazyk v němž je skript napsán, nebo nemusí mít povoleny některé funkce. Toto je hlavní nevýhoda.

2.2.2 Skript na serveru

Na serveru

- CGI-skripty
- PHP
- ASP(Aktive Server Pages)
- Server Side JavaScript



Na serveru – jestliže počítač uživatele požaduje webovou stránku ze serveru, server stránku sestaví a teprve pak odešle. servery mohou podle požadavku uživatele sestavit libovolnou stránku. Server může provést libovolný počet operací a výsledek pošle ve formátu HTML žadateli.

2.3 Přístup k datům - Datová rozhraní

Jsou to komponenty programovacích prostředí s určitými standardy, které pomáhají zpracovat, nebo spravovat data, ať již je to sešit MS Excel, Dokument Word, databáze Access na pevném disku PC v kanceláři, nebo zabezpečená databáze na serveru.

2.3.1 DAO

Data Access Object sjednocuje logiku pro přístup k datům z různých zdrojů. Zdroji mohou být relační databáze, objektové databáze, textové soubory. Zavádí rozhraní pro práci s daty. Patří k základním J2EE vzorům. Aplikační vrstva(program) potom využívá služeb DAO vrstvy pro přístup k datům jak externím, tak lokálním.

2.3.2 ADO

Komponenta Databáze Access ADO – ActiveXTM Data Object je souborem objektů implementovaných na bázi OLE DB – specifikuje několik standardních rozhraní pro přístup k datům. Umožňuje vytvářet klientské aplikace pro čtení a zápis dat na databázovém serveru přes ovladač ODBC. Jedná se rychlý, jednoduchý přístup s malými požadavky na využití paměti.

Obsahuje objekty :

- Connection – spojení do externí dtb
- Recordset – umožňuje operace nad externími daty SQL příkazy, projde množinu záznamů a vrátíte, které vyhovují objektu Command
- Command – procedura s podmínkou ve formě SQL dotazu
- Field – přístup k určitému poli uvnitř množiny záznamů(Recordset)
- Parametr – předání hodnot z objektu Command, nebo z procedur
- Error – vrací chyby ovladače ODBC

2.3.3 RDO

Remote Data Object. Ve Visual Basic Report Data Kontrol, spojuje RDO a ODBC. Tento ovládací prvek se používá ve spojení s SQL Serverem.

2.3.4 ODBC (Open Database Connectivity)

Další ovladač síťového rozhraní. Umožňuje pracovat libovolnému SQL serveru s libovolnou ODBC-podporující aplikací. Definice ODBC se postupem času změnila a už se nejedná o mezistupeň mezi aplikací a nativními ovladači databázového serveru. Tím zanikly výkonové ztráty v čase přístupu atd. Poskytuje, jednotný přístup k datům. Klient, je to ASP stránka, PHP skript nebo Win32 aplikace. Připojení k externí databázi. Vlastní ODBC ovladač přejímá od aplikace příkazy pro zpracování dat a vykonává tyto příkazy v datové základně. Výhodou tohoto řešení je možnost jednoduché záměny ODBC ovladače za jiný - tj. formátu zpracovávaných dat. Lze zpracovávat (číst i modifikovat) a prezentovat datové soubory v libovolném WIN produktu, který podporuje ODBC rozhraní (Delphi, Access, Word, Excel, MS-Query, Power Builder, Crystal Report,...).

2.4 Java script

JavaScript je jednoduchý programovací jazyk, který je možno zapisovat přímo do HTML stránky. Pochází z vývojářské dílny firmy Netscape, ale v současnosti jej podporuje i Internet Explorer firmy Microsoft.

JavaScript je snadno interpretovatelný programovací jazyk se základními objektově orientovanými schopnostmi. Umožňuje vložit do webových stránek tzv. „proveditelný obsah“ – to znamená, že stránka na webu již nemusí být jen statický dokument, ale může obsahovat dynamické programy, které komunikují s uživatelem, řídí prohlížeč a dynamicky vytvářejí obsah HTML stránky.

Tento skriptovací jazyk je poměrně rozšířený a stále více používaný právě pro oživení stránek WWW, pro vytváření jednoduchých aplikací, které jsou spouštěny prostřednictvím prohlížeče. V souvislosti s pojmem „Dynamické HTML“ mají programy v jazyce JavaScript oproti javovým appletům tu výhodu, že jejich zdrojový kód je přímo součástí stránky HTML a není jej potřeba nijak kompilovat – jedná se totiž o interpretovaný jazyk.

JavaScript je sice jazykem podobným Javě či C++, avšak podstatně jednodušším, zbaveným některých rysů. JavaScript je programovací jazyk bez typové kontroly, což znamená, že proměnné nemusí mít specifikovaný typ.

Jak již bylo řečeno, JavaScript je objektově orientovaný jazyk. Existují zde tedy metody a funkce, prostřednictvím kterých je možné ovlivnit údaje uložené v objektech. Kromě objektů typu Math (práce s matematickými funkcemi), Date (práce s časovými údaji) a String (práce s řetězci) je JavaScript omezen možnostmi prohlížeče. Může pracovat pouze s vlastními objekty takového prohlížeče, nebo objekty do něj vloženými (applety, ActiveX komponentami, PlugIns-moduly). JavaScript používá jazyka HTML a jeho elementů. Neumožňuje však práci s grafikou nebo se zvukem.

2.5 ASP

2.5.1 Popis ASP

Firma Microsoft vyvinula tuto aplikaci na bázi “serverové ” technologie k vytváření atraktivních dynamických webových stránek. Současně zajišťuje funkčnost na různých klientech od MS Explorer po Netscape Navigátor, včetně starších verzí.

Programovacím jazykem ASP je VBScript, pracující na bázi Visual Basicu, který běží na straně serveru a generuje výsledný kód v HTML, včetně např. kaskádních stylů. ASP spolupracuje i s JavaScriptem, nebo serverovými komponentami ActiveX. Je na vůli každého, jaký jazyk si na začátku stránky nadeklaruje.

Jako server nabízí Microsoft IIS(Internet Information server), nebo PWS(Personal Web Server) většinou pro vývoj aplikací off-line. Tyto primární servery pracují na platformě Microsoft – Windows NT atd. ASP je bezplatně nabízen jako součást serverových instalací od Microsoft.

Obsah stránek je tvořen podobně jako technologií CGI(Common Gateway Interface), kde se obsah stránky sestaví už na serveru. IIS podporuje nový typ aplikací s rozhraním ISAPI(Internet Server Application Programming Interface). ISAPI běží na rozdíl od CGI ve stejné paměťové oblasti jako webový server(jako dynamické knihovny DLL). Rozhraní ISAPI jsou tímto více než 10x rychlejší než CGI. Jedná se o stránky HTML s příponou *.asp, se skripty pro ISAPI.

ASP samozřejmě podporují rozhraní ODBC(Open Database Connectivity), spolu s SQL(Structure Query Language) pro práci s externími databázemi. Nejprve je potřeba nadefinovat příslušný datový zdroj ODBC a zároveň mít na serveru potřebné ovladače ODBC.

ASP podporují multitasking obsluhu více uživatelů najednou s efektivním využitím systémových zdrojů. Tzv. Multithreading“ obsluha souběžně připojených uživatelů zabezpečují Vlákna(threads) – výkonná část paměťového prostoru aplikace IIS. Zpráva tzv. Objektů znamená vytvoření jedné instance aplikačního objektu s jedním vláknem pro každého připojeného klienta.

ASP automaticky sdílí propojení na data, nimiž aplikace právě pracují přes ADO.

2.5.2 Program

Příkazy ASP se od běžného HTML oddělují znaky „<% ” a „%>“.Kód ASP většinou ve VBScriptu potom pracuje s objekty a komponentami.Jsou to komponenty ActiveX – technologie založené na Component Object Model (COM). Znamé moderní objektové programování.

Soubor Global.asa definuje začátek a konec jak celé aplikace, tak práce jednotlivých uživatelů. Podle toho se pak řídí rozložení proměnného pracovního toku(množství uživatelů, využívaná data na serveru..).

2.5.3 Objekty ASP

Application - aplikační proměnné(sdílení informací dostupných pro všechny uživatele) úkolem je zamezit konfliktu kdy dva uživatelé budou chtít upravovat jednu proměnnou současně

Session - stará se o vytváření uživatelských proměnných, kde jsou data pouze pro jednoho uživatele, která patří pouze jemu

Request - stará se o přístup k informacím a datům od klienta na server

Response - obsah vrácený prohlížeči stránkami ASP ze serveru.Obsahuje kolekci Cookies. Ta umožňuje nastavovat a zapisovat do kolekce Cookies na klientu. Pokud neexistuje, je vytvořena, eventuálně stará hodnota nahrazena novou

Server - vytváří přístup k metodám a funkcím, vytváří instance pro přístup k datům

2.5.4 Komponenty ASP

ADO

File Access - přístup k textovým souborům na webovém serveru

Browser Capabilities – identifikuje prohlížeč, který přistupuje na server a seznam podporovaných funkcí

Ad Rotátor – řízení rotace reklamních obrázků na webovém serveru další informace ohledně zobrazené reklamy

Content Linking – umožňuje spojovat oddělené webové stránky dohromady a jimi potom rolovat

2.6 PHP- „Professional Home Pages“

2.6.1 Popis PHP

PHP je spolu s databází Mysql a serverem nejrozšířenější technologií pro tvorbu dynamických WWW stránek. Stejně jako u ASP je kód aplikace na serveru, takže uživatel vidí pouze výsledný HTML, XLM kód, který je mu určen. Na rozdíl od ASP podporuje PHP většina web-hostingových serverů.

Kód PHP se odděluje znaky „<? ” a „?>“.

PHP podporuje protokoly IMAP, SNMP, NNTP, POP3, HTTP a další

2.6.2 Datové typy PHP

PHP podporuje následující datové typy, ve spolupráci s databázemi je seznam čtenější

Boolean(logická hodnota)

Integer(celé číslo)

Double(desetinné číslo)

String(řetězec)

Array(pole)

Object(objekt)

2.6.3 Podpora databází

Podpora databází Adabas D, dBase, express, FilePro (read-only), Hyperwave, IBM DB2, Informix, Intres, InterBase, FrontBase, SQL, Direkt, MS-SQL, MySQL, ODBC, Oracle (OCI7 and OCI8), Ovrimos, PostgreSQL, Solid, Sybase, Velocis, Unix dbm

2.6.4 Objektové programování

PHP podporuje objektové programování, kde každý objekt může být popsán jako seznam proměnných a souvisejících metod

2.6.5 Další funkce

- Podmínky if, elseif
- Příkazy cyklů while, do-while, for, foreach
- Pokračování, nebo ukončení cyklu continue, break
- Načtení souboru require, include
- Podpora formulářů
- Cookies (ověření uživatele)

2.6.6 Zpracování souborů s daty

Čtení souborů

- PHP pracuje se soubory a je-li to nastaveno, rovněž se vzdálenými soubory pomocí http a ftp protokolů
- PHP pracuje se složkami a s komprimovanými soubory
- PHP umí zpracovat soubory obsahující data
- PHP přečte data a uloží rovnou do databáze

LOAD DATA INFILE SQL příkaz na databázi MySQL podporuje import dat z textového souboru do tabulky. Příkaz je to jednak dosti mocný a jednak poměrně rychlý. Nejdůležitější volby jsou :

LINES TERMINATED BY umožňuje definovat znak, který bude oddělovat řádky. Pokud importujete z Windows, měli byste nastavit `'r\n'`.

FIELDS TERMINATED BY slouží k definici znaku, jímž se oddělují sloupce. Např. tabelátor je `'t'`,

PHP rozlišuje přístupová práva k jednotlivým souborům

Pomocí příkazu `Readfile` lze číst textové soubory. Uživatel pod nímž běží PHP, musí mít minimálně práva soubor číst. Jinak `Readfile` selže.

Pomocí třídy `dir` s metodami `read`, `rewind`, `close` lze rovněž pracovat s celým stromem adresářů.

PHP přímo podporuje zpracování souborů s hodnotami oddělenými středníky, ke zpracování a uložení do databáze. K rozsekání slouží funkce `fgetcsv`.

2.6.7 Další možnosti PHP pro práci se soubory

- Otevírat vzdálené soubory pomocí protokolů HTTP nebo FTP
- Otevírat porty (třeba COM)
- Číst ze standardního vstupu nebo zapisovat na standardní výstup
- Soubory kopírovat, přesouvat, mazat, měnit jim práva a tak dále
- Soubory šifrovat

2.6.8 Parsery – čtení dokumentu XML

Struktury XML využívají již hotové knihovny pro práci s XML. PHP používá Expat. Expat patří do velké skupiny software, kterou nazýváme **XML parsery**. "Parser" je program, který rozumí XML a umí k němu strukturovaně přistupovat. Parser umí z dokumentu číst a ví, co udělat, když v XML narazí na značku.

PHP má po instalaci podporu XML rovnou zapnutou. Lze ihned začít využívat funkcí pro práci s XML. Skript zpracovávající soubor XML :

1. Musí vytvořit instanci parseru.
2. Musí zaregistrovat obslužné funkce – definují se funkce v PHP, které stanoví, co s danou věcí (značky, data) dělat poté, co na ni v dokumentu narazíme. Parser musí vědět, že má tyto funkce zavolat.
3. Parseru se předá vlastní xml soubor a ten jej zpracuje a skončí.

2.6.9 Výstup – nahrání souborů na server

Zápis souboru na server pomocí protokolu ftp nebo http, pokud je to povoleno. Příkazy fopen, fwrite(fread), fclose. Musí být splněny následující podmínky:

Nastavení PHP na Vašem serveru musí otevírání souborů pomocí FTP podporovat

Server FTP musí podporovat běh v pasivním režimu

Můžete otevírat soubory buď pro čtení, nebo pro zápis, ale ne obojí

2.6.10 Výstup – HTML stránka

V případě, že chceme optimalizovat stránku pro tisk je nutné dodržet

Co nejméně druhů písma - třeba i jen jeden

Co nejméně grafiky - nejlépe žádná, ještě tak nějaké to jednoduché logo

Co nejméně barev - měla by stačit černá a bílá

2.6.11 Výstup – PDF soubor

Propracovaný projekt, zabývající se přímou tvorbou souborů PDF pomocí PHP je FPDF. U PDF můžete lehce nastavit písmo, znakovou sadu, orientaci stránky a její velikost, okraje. Sestava vypadá na všech systémech stejně. Knihovna FPDF nám dává dost možností. Je potřeba použít české fonty.

2.6.12 Výstup – XML soubor

Možnost výstupu dat například z MySQL pomocí PHP jsou textové soubory. Můžeme exportovat prakticky do libovolného otevřeného formátu, např. XML s tím, že do textu v cyklu se přidávají příslušné značky.

2.6.13 Výstup – Export do souboru z databáze

Použitím příkazu `SELECT INTO OUTFILE` umožňuje export dat z databáze ne ve formě sady záznamů, ale jako soubor.

neumí odstranit existující soubor. Museli bychom to provést ručně před započítím exportu.

na provedení příkazu jsou potřeba na serveru MySQL dostatečná práva

vzniklý soubor je přístupný pro zápis všem uživatelským účtům na serveru

Lepším řešením je vytvořit sadu záznamů, procházet ji a cílový soubor si sestavit, jaký chceme, kde chceme (na lokálu)

Nejdůležitější příkazy pro práci s databází :

Příkaz	Význam
<code>mysql_connect</code>	Připojí se k MySQL
<code>mysql_select_db</code>	Vybere pracovní databázi
<code>mysql_query</code>	Vykoná dotaz (výběrový, aktualizací atd.)
<code>mysql_fetch_array (_row, _object)</code>	Převezme řádek výsledku do asoc.pole (pole, objektu)
<code>mysql_num_rows</code>	Vrátí počet řádků ve výsledku

2.6.14 Výstup – Graf

V PHP se dají grafy jako obrázky generovat pomocí standardní knihovny GD2, která poskytuje pouze základní možnosti pro kreslení grafiky. Rozšíření od Švédské firmy Aitus – soubor knihoven JpGraph. Ke komerčnímu využití je bohužel distribuována pod licencí QPL 1.0..Je napsána v prostředí UNIX, plně funkční i pod Windows. Je však potřeba verze PHP 4.0.2 a vyšší. Funkčnost knihovny po dohrání lze vyzkoušet příkazem `phpinfo()`.

Pomocí JpGraph můžeme kreslit, resp. generovat následující grafy:

- sloupcové
- bodové
- linkové
- plošné
- 3D
- koláčové
- pavučinové

Knihovny JpGraph

- `jpgraph.php` – základní knihovna, musí být přilinkována pokaždé

Pro kreslení grafu je nutná jedna či více z následujících knihoven

- `jpgraph_log.php` – rozšíření pro kreslení logaritmického popisu os X a Y
- `jpgraph_line.php` – rozšíření pro možnost kreslit jakékoli linkové grafy
- `jpgraph_bar.php` – rozšíření pro kreslení sloupcových grafů
- `jpgraph_error.php` – rozšíření pro kreslení chybových grafů (myšleno jako hodnot, které udávají chybu např. v bodě)
- `jpgraph_scatter.php` – rozšíření pro kreslení bodových grafů
- `jpgraph_spider.php` – rozšíření pro kreslení pavučinových grafů
- `jpgraph_pie.php` – rozšíření pro kreslení koláčových grafů
- `jpgraph_pie3d.php` – rozšíření pro kreslení 3D koláčů

JpGraph nabízí následující možnosti :

- Množství typů grafů
- možnosti vykreslení (osy, popisky, pozadí, ...)
- velmi snadná práce (nastavení, předávání parametrů)
- automatický výběr formátu (JPG, PNG, GIF)
- velmi podrobná a obsáhlá dokumentace
- generování klikacích map
- logaritmické osy
- anti-aliasing u čar (bohužel nefunguje u koláčových grafů)
- poměrně snadná možnost úpravy přehledných knihoven
- a mnoho dalších, dle mého skromného názoru méně důležitých vlastností

2.7 XAML

Zatím posledním nástrojem v cestě za zpracováním metadat(data na pevném disku, data z webu, e-mailů..) od firmy Microsoft je jazyk XAML(eXtensible Application Markup Language). Definiční jazyk uživatelského rozhraní použitý poprvé ve Windows Vista.

XAML je jazyk, který definuje objektový model a je založený na XML. Je možné říci, že slouží hlavně k popisu uživatelského rozhraní (XAML lze použít i pro vytváření objektů které nemají s uživatelským rozhráním nic společného). Popis(vlastnosti, chování) uživatelského rozhraní Windows Vista jsou jednou z funkcí, kde lze XAML využít.

Kód a obsah jsou v XAML odděleny. Za každým jeho tagem se skrývá objekt a hlavním účelem XAML je tedy ne popisovat vzhled, ale zjednodušit vytváření objektů, které tvoří uživatelské rozhraní.

V XAML je možné vytvářet buď 'normálně' vypadající aplikace, které jsou spuštěny v okně a nebo aplikace, které mají navigační panel (tlačítko zpět a vpřed) pomocí kterého se dá pohybovat mezi jednotlivými XAML stránkami.

Vytvořené aplikace je také možné upravit tak, aby byly spustitelné v Internet Exploreru (jelikož se jedná o platformu Microsoft). Oba přístupy je možné kombinovat, takže otevřete aplikaci s navigačním panelem, kde se přepnete do Exploreru, ten zavřete a naopak. To znamená jednotný přístup k UI (User Interface), dokumentům, médiím

Využití stávajících :

UI znamená User Interface. Je to obecný pojem pro všechny platformy.

Druhy UI komponent prakticky vychází z formulářů v jazyce HTML.Jsou to :

- **PushButton** (tlačítko)
- **ComboBox** (roletové menu)
- **ListBox** (roletové menu - trvale otevřené)
- **CheckBox** (zaškrtačkové pole)
- **RadioButton** (výběrové pole)
- **Scrollbar** (posuvník)
- **ScrollPane** (něco jako frame - umožňuje zobrazovat MovieClip)

2.8 Google Office

V rámci boje o zákazníka se zpráva samostatných soukromých dat neomezuje pouze na lokální úložiště, nebo firemní servery. Rozšiřují se on-line služby providerů a portálů. Tyto nadstandardní služby přinesou kýžený výdělek ve formě reklamy. Od začátku roku 2007 společnost Gogole konkuruje se svým “free” kancelářským balíkem, jehož součástí je v rámci domény kromě Gmailu a možnosti publikování na webu (Google Pages) i “Word a Excel”. Přechod na tento free software zvažují firmy Disney, nebo studia Pixar. Ke kladům lze také přičíst, že již nyní jsou na webu k mání jednotlivé jazykové varianty.

2.8.1 Služby

Placený balík Google Apps Premier Edition obsahuje za 50 dolarů na rok za uživatele :

Gmail – 10GB

Google Talk

Google Calendar

Start Page

Docs & Spreadsheets (Word, Excel)

Page Creator

K tomu je nutné připočíst asistenci včetně tf. Podpory Google Spreadsheets

2.8.2 Výhody práce s daty na serveru

1. umožňuje snadnou práci více lidí nad jedním dokumentem,
2. umožňuje přístup k tabulkovému procesoru odkudkoli, je neuvěřitelně pohodlné pracovat v tabulkovém procesoru v internetovém prohlížeči, kde je možné využívat e-mail, kalendář a další nástroje, takže není nutné přecházet z e-mailu do jiného okna, čekat několik sekund na otevření aplikace a pak pracovat ve zcela jiném prostředí, které spotřebovává další paměťové prostředky,
3. tato aplikace je zdarma,
4. business výhoda je, že Google má v online světě velmi silné postavení a řada lidí jeho produkty ráda testuje, protože jeho značka je dost populární – v tom mu může málokdo konkurovat a Google Spreadsheet se tak může rychleji rozšířit.

2.8.3 Nevýhody Google Office

1. je funkčně velmi omezený,
2. uživatel musí být online, aby jej mohl používat,
3. nechutí uživatelů ke změnám - neradi se učí nové věci a mění své staré zvyky,
4. data jsou uložena vzdáleně, mimo disk uživatele, může docházet k výpadkům

2.8.4 Google Office - zhodnocení

Google Spreadsheet nemůže svou funkčností v současné době konkurovat Office od Microsoftu, ani třeba s OpenOffice.org. Tam, kde se pracuje s kontingenčními tabulkami, makry a kde navíc soubory bývají v desítkách MB, nemá tato aplikace zatím větší šance. Je však pravdou, že pokročilejší funkce používá 20% uživatelů. Balík se nemusí instalovat odpadá starosti s upgradováním. Naproti tomu nutnost On-line připojení k Internetu jde proti variabilitě tohoto sw. Skupina uživatelů se tedy omezí na lidi, co mají pevné připojení na obvyklých místech, nechtějí náročné funkce a mají rádi změnu, která není moc komplikovaná. Může to být plně vystačující aplikace pro ty kdo počítají daně, dělají referáty a zpracovávají protokoly do školy apod. Jestliže do budoucna Google dodá nástroj pro grafy a podle zkušeností z "provozu" rozšíří ku prospěchu stávající funkce, řada lidí sem postupně začne přecházet, a v delším časovém období začnou nové věci používat i lidé méně ochotní čímž se učí novému. Čeká se také na to jak se zachovají ostatní na trhu se zákazníky např. Yahoo a další.

Databázová podpora Google Office naznačuje jeho možnosti v nahrazení funkce MS Access. Google Spreadsheets může nahradit menší databázové aplikace, které byly napsány na podporu nějakého menšího procesu, projektu, u kterého je potřeba, aby k datům mohlo vzdáleně přistupovat více lidí a naráz je mohli editovat a změny rovnou i viděli. Na toto je již nyní GS skvěle použitelný. V aplikaci může každý účastník projektu mít svůj list a sem zaznamenávat všechny změny, které se týkají jeho práce, z kteréhokoliv místa na zemi. Bude označovat, které dokumenty k němu odešly, jaké byly přijaty, co je ještě nutné dodělat. Celý projekt se bude skládat z několika takových listů a bude zde aktuální přehled o celkové situaci.

Tato funkčnost je shodná s dnešními vyspělými klient server dynamickými webovskými stránkami, s tím rozdílem, že zde odpadá funkce správce serveru, webmastera, správce softwaru, správce serveru a další činnosti technické podpory.

Sám Microsoft musí v konkurenčním prostředí hledat takovéto směry a nečelit konkurenci pouze stávajícím krabicovým sw. Do budoucna pravděpodobně udělá některé online verze Office zdarma. Tomu již nyní nasvědčuje projekt Live.com, který se do budoucna jistě rozšíří a Microsoft tak půjde prakticky sám proti své původní filozofii na které zbohatl. Tím se bude snažit také část svých zisků o které přichází nahradit penězi z reklamy.

Prozatím není Google Spreadsheets služba srovnatelná s Office od Microsoftu, ani třeba s OpenOffice.org, je to velmi zajímavá aplikace a ukazuje směr, jakým se vydává dnešní softwarový svět. Google Spreadsheets tak začíná rozhýbávat stojaté vody softwarového průmyslu a tím alespoň částečně vytvoří silnější konkurenci, dále rozšíří počet věrných uživatelů Googlu a navíc silněji upozorní na budoucí vývoj IT.

3. Prezentace dat

3.1 Logické formátování dat - XML

3.1.1 DTD

DTD(Document Type Definition)

Pokud mají informační a komunikační systémy fungovat efektivně, je potřeba, aby se používaly jednotné formáty dat. Bez nich se spousta prostředků zbytečně vyplývá na implementaci různých pomocných utilit a modulů, které umožní spolupráci softwaru od různých výrobců.

Konzorcia proto schvalují formáty založené na XML, které vyhovují potřebám všech.

Pomocí DTD jsou také pevně definovány všechny stávající normy HTML. Tato datová struktura např. dokumentu XML definuje jeho značky(tagy), jak do sebe budou jednotlivá data vnořena, jaké budou možnosti jejich výskytu a na kterých místech. Definuje vztahy mezi jednotlivými daty atd.. Úkolem je vytvořit dobře propracovanou strukturu DTD, podobně jako u databáze.

Dokument XML je stromová struktura ve které jsou uspořádána data. Musí existovat hierarchie dat(kořen, nejvyšší třída a větvení. Co bude logicky výš (nadtřídou) a co níž (podtřídou). XML je syntaxe pro ukládání hierarchicky organizovaných dat jako jsou adresáře, katalogy, uživatelské příručky atd. Může bohužel obsahovat pouze textová data.

XML je jedním ze značkovacích jazyků. Značky v dokumentu můžeme rozdělit na **Formátovací** - formátování a struktura dokumentu

Logické - logické označení dat

HTML které nerozlišuje, mezi těmito typy, a díky tomu dokumenty HTML dosahjí slušných možností. Právě proto jsou DATA z HTML obtížněji vyzískatelná.

Nové trendy dali vzniknout novým specifikacím např. kaskádovým stylům (CSS).

3.1.2 Struktura XML dokumentu

Struktura programu registru vozidel, by logicky potom mohla vypadat například takto:

- Majitel
 - o Místo
 - Ulice
 - Město
 - PSC
 - o Jméno
 - Křestní
 - Příjmení
 - o Rodné číslo
- Výrobní číslo
- Cena
- Specifikace

Nadtřída 'Automobil' obsahuje informace o vlastníkovi vozidla, výrobní číslo, cenu a specifikaci.

Na rozdíl od HTML existuje způsob, jakým uspořádat položky v určitém pořadí.

XML definuje data jako stromovou strukturu. Záznamy z registru vozidel, popsané výše převedeme na stromovou strukturu. Je potřeba definovat strukturu, kde která data budou umístěna, a jaký je jejich význam – definujeme DTD.

DTD vytvořené pro příklad registru vozidel by mohlo vypadat např. takto:

PŘÍKLAD:

```
<?xml version="1.0" encoding="windows-1250"?>
<!DOCTYPE DOKUMENT [
  <!ELEMENT DOKUMENT (AUTO)>
  <!ELEMENT KARTA (MAJITEL,VYR_CISLO, CENA,SPEC)>
  <!ELEMENT DRZITEL (JMENO,ADRESA,RODNE_CISLO)>
    <!ELEMENT JMENO (KRESTNI,PRIJMENI)>
      <!ELEMENT KRESTNI (CDATA)>
      <!ELEMENT PRIJMENI (CDATA)>
        <!ELEMENT ADRESA (ULICE,MĚSTO,PSC)>
          <!ELEMENT ULICE (CDATA)>
          <!ELEMENT MĚSTO (CDATA)>
          <!ELEMENT PSC (CDATA)>
          <!ELEMENT RODNE_CISLO (CDATA)>
          <!ELEMENT SER_CISLO (#PCDATA)>
        <!ELEMENT CASTKA (#PCDATA)>
        <!ELEMENT SPEC_Z (#PCDATA)>
    ]>
</DOKUMENT>
<AUTO>
  <VYR_CISLO>5637676876</SER_CISLO>
  <MAJITEL>
    <JMENO>
      <KRESTNI>Petr</KRESTNI>
      <PRIJMENI>Musil</PRIJMENI>
    </JMENO>
    <ADRESA>
      <ULICE>Panská 81</ULICE>
      <MESTO>Praha</MESTO>
```

```
<PSC>11000</PSC>
</ADRESA>
<RODNE_CISLO>6710150300</RODNE_CISLO>
</MAJITEL>
<CENA>1000000</CASTKA>
<SPEC>Osobní automobil</SPEC>
</AUTO>
</DOCUMENT>
```

XML soubor neobsahuje žádné informace o formátování. Používá se formátovací soubor CSS, XSL, nebo jiný formátovací jazyk.

Variabilita (XML) spočívá v tom, že může být propojen na několik formátovacích souborů (např. pro tiskárnu, zároveň pro obrazovku prohlížeče, nebo pro novinky webu apod.)

3.1.3 Využití XML

Použití XML není omezeno jen na výměnu dat mezi obchodními partnery. V USA lze již pomocí XML posílat čtvrtletí finanční výkazy EDGAR pro úřad U. S. Securities. V ČR se používá ve státní zprávě produkt firmy Software602 XML Filler. Použití XML pro tyto účely se jeví jako zcela ideální, podpory tohoto formátu ve stávajících účetních a ekonomických systémech je v dohlednu.

Vyznačení významu jednotlivých částí stránky má pozitivní efekt na přesnost a rychlost vyhledávání informací.

Pomocí stylů XML snadno zcela automatizovaně z konvertujeme do mnoha dalších formátů. Stylové jazyky jako DSSSL a XSL jsou velice flexibilní. DSSSL naimportuje původní styl a předefinuje potřebné parametry. DSSSL styly jsou vlastně SGML dokumenty. Jednou proto můžeme z dokumentu vygenerovat PDF soubor vhodný pro DTP(programy slouží pro sazbu Desktop publishing) , podruhé zase sadu HTML stránek, kde co jedna stránka, to jedna kapitola dokumentu.

3.1.4 Složení XML dokumentu

Pod pojmem XML se skrývá :

- **XML File** – soubor obsahující posloupnost znaků, které definují XML soubor
- **XML Document** – což je model dat z tohoto souboru, vytvořený prohlížečem, který je umístěn v paměti.

Členění XML

XML dokument je stromem :

- s kořenovým elementem – **ROOT ELEMENT**
- dalšími uzly(poduzly) kořenového uzlu

Každý z uzlů musí být typu:

- znak - obsahuje jeden znak ze sady UNICODE/ISO-10646 a nemá žádnou jinou strukturu
- zpracovávající instrukce
 - má pole jméno
 - pole obsah(sekvence znaků)
- komentář
 - položka komentář obsahuje posloupnost znaků

Tyto tři uzly nemají potomka jsou listy(koncové body) ve stromě, které jsou vázány na uzel element

- element
 - kořenový element dokumentu(má 0 a více potomků)
 - má jméno(identifikátor-klíčové slovo)
 - sadu atributů(dvojici dat : klíčové slovo – hodnota)
 - potomci jsou řazeny v uzlu(lze najít xtý člen)
 - elementy řazeny nejsou(nelze určit pořadí)

Řazení elementů se využívá v DOM (Document Object Model) objektovém modelu dokumentu. Je to rozhraní nezávislé na platformě a použitém programovacím jazyku. Umožňuje aplikacím a skriptům dynamický přístup a změnu obsahu, struktury a stylu dokumentů. Mezi prohlížeči má dnes nejlepší podporu pro DOM má Mozilla.

3.2 XHTML

„Intelligentní“ webové stránky

XHTML znamená oproti klasickému HTML možnost definice vlastních značek, které přesně vyznačí význam jednotlivých částí stránky. Tím se zprůhlední struktura celého dokumentu a má to pozitivní efekt na kompatibilitu a rychlost vyhledávání informací.

V praxi lze využít dva způsoby :

- Vytvoření zcela unikátní vlastní sady značek.
- Stávající HTML doplnit o značky, kterými se označí části stránky důležité pro vyhledávání. Konsorcium W3C vytvořilo standard převodu, kde XHTML 1.0 odpovídá HTML 4.0.

(HTML 4.0) obsah se, rozdělil do několika nezávislých modulů. Budeme pak mít modul pro formátování textu, pro tvorbu odkazů, pro tabulky, pro výrobu formulářů nebo pro zařazování obrázků. Do budoucna každý prohlížeč bude společně s požadavkem na stránku posílat i svůj profil – informaci o tom, co dané zařízení zvládne. Pokud bude stránka dostupná ve více variantách, server vybere tu s odpovídajícím profilem. Tento poměrně obecný model umožní vývoj stránek pro mnoho zařízení s rozdílnými schopnostmi – PC, mobilní telefony, WebTV, organizéry nebo třeba herní konzole.

3.3 Grafické formátování stránky

3.3.1 CSS

CSS (Cascading Style Sheet) - Kaskádní styly (sešity) stylů. V současné době existují specifikace CSS1 a CSS2. Byly zavedeny s HTML 4.0, kde se jedná o interní, nebo externí soubor který rozšiřuje, případně separuje formátovací vlastnosti HTML . Mají charakter logických formátovacích nástrojů. Definovanou skupinou značek definuje vlastnosti.

3.3.2 XSL

XSL (eXtensible Stylesheet Language) – také jazyk tzv. Stylesheetů. Pro tvorbu transformačních stylů. Jazyk XSL však nabízí mnohem širší možnosti použití než CSS.

3.4 Aktuální data - RSS

3.4.1 Využití

RSS (Really Simple Syndication) znamená v překladu jednoduché publikování. Je to nejnovější způsob, jak dát lépe uživatelům vědět o novinkách na webových stránkách. Dokumenty jsou většinou psány v XML, XHTML.

V praxi to znamená např. soubor XML s informacemi o novinkách (v položce tagu channel), který je na serveru. RSS kanál je potom jako odkaz na XML soubor např. na úvodní stránce instituce.

Takovýmto způsobem se prezentují např. weby Google News, Idnes.cz, Computerworld, Geoinformace.cz. Tím si zvyšují popularitu a kvalitu z pohledu návštěvnosti. RSS kanály jsou na stránkách identifikovatelné.

Pro většinu webů je výhodné, nechat si RSS kanál generovat pomocí nějakého skriptovacího jazyka z dat serverové databáze. Zpráva stránek pak obsahuje administrátorskou sekci, odkud se novinky přidávají do položky item, nebo novinky obsahují speciální příznak a jsou automaticky přiřazeny do RSS kanálu.

3.4.2 Struktura RSS dokumentu

Struktura RSS dokumentu obsahuje :

1. Informaci o verzi XML, RSS

2. Channel

Prvek <image>

- <url> url adresa obrázku (gif, jpeg..) prezentující kanál
- <title> popis obrázku kanálu
- <link> url adresa, kde je kanál umístěn
- <width> výška, šířka obrázku
- <description> popis obrázku

Prvek <cloud>

- identifikuje webovou službu, která popisuje rssCloud

Prvek <ttl>

- určuje dobu, za kterou se RSS aktualizuje ze serveru

Prvek <textinput>

- většinou nepodporovaný element
- <title> popis odesílacího tlačítka v textovém vstupu
- <description> popis tlačítka
- <name> jméno textového objektu v textovém vstupu
- <width> výška, šířka obrázku
- <description> popis obrázku, který zpracovává vyhledávání-in

3. Item prezentuje jednotlivou položku daného kanálu

Prvek <source>	- adresa url(odkaz na XML soubor) a jméno RSS kanálu
Prvek <enclosure>	- pro vložení multimediálního souboru
	- <url> adresa kde je příloha
	- <lengh> velikost přílohy (v bytech)
	- <type> typ přílohy
Prvek <category>	- seznam/začlenění kategorií (nepovinný prvek)
Prvek <pubDate>	- datum, kdy má být položka zveřejněna
Prvek <guid>	- řetězec identifikující položku - zda se jedná o nový příspěvek
Prvek <comments>	- url adresa stránek – komentář položky
Prvek <author>	- e-mail na autora položky

3.4.3 Čtečky RSS dokumentu

Jsou to externí samostatné programy. Čtečky umožňují snadno přidávat kanály(propojení na další stránky s novinkami). Stačí kliknout na odkaz a pomocí průvodce si ho přidáte do seznamu. Zdroje lze řadit do složek, importovat a exportovat do OPML souborů,HTML. Otevřené články posílat mailem.

Zde jsou některé z nich :

Abilon
 Sharp Leader
 RSS Tracker
 RSS Point
 Feed Demon
 Blog Express

Prohlížeč Firefox umožňuje přidat kanály RSS jako aktivní záložky, Opera zobrazuje hlavičky a obsah ve vertikálním rozložení.

Naproti tomu Online čtečka - agregátor běží na web serveru a obsahuje stránku s titulky získanými z RSS(např. www.intravnews.com, www.kratce.cz).Zobrazuje titulky z RSS kanálů registrovaných serverů.

3.5 Progres nových multiplikačních vývojových prostředí

Důvody :

- Hardwarová inovace přestihuje software
- Displeje s vysokým DPI
- Přetížení různými formami informací
 - Dokumenty, hudba, video, fotografie apod.
 - Potřeba lépe a chytřeji vizualizovat informace

- Dvouvrstvý model(oddělení grafického návrhu aplikace a aplikačních logik)
- Model známý z ASP.NET
- Návrh vzhledu pomocí značkovacích jazyků (XAML)
- Používá stávajících zažitých logik(znalostí programátorů)

Nástroje :

- Rozšíření funkcí ovládacích prvků(tlačítka.lišty, chec boxy) obsahují
 - Commands – operace jako jsou Cut, Copy nebo Paste
 - Properties – vlastnosti modifikující vzhled nebo chování
 - Events – ovládací prvky vyvolávají události
 - Metody – část funkcionality je dostupná formou metod
 - Commands, Properties a Events (dostupné v XAML)
- Aplikační logika
- Data binding – vazba na datové zdroje
- Template – šablona definující vzhled prvku

3.6 Technologie Microsoft zpracování dat a grafiky

.NET (dot NET = tečka NET) - znamená všeobjímající název pro soubor technologií v softwarových produktech, které tvoří celou platformu, která se používá na vývoj pro Web, Windows i Pocket PC. Common Language Infrastructure je standardizovaná specifikace jádra .NET.

Základní komponentou je Microsoft .NET Framework, prostředí potřebné pro běh aplikací a nabízející jak spouštěcí rozhraní, tak potřebné knihovny. Pro vývoj .NET aplikací vydal Microsoft Visual Studio .NET.

WPF (Windows Presentation Foundation)

WPF (Windows Presentation Foundation) je nástupcem podsystémem .NET Frameworku, znamená jednotný přístup k UI, dokumentům a médiím. Je určen k tvorbě aplikací s bohatým grafickým rozhraním.

WPF okolo roku 2004 známé pod krycím názvem Avalon: znamená nový pohled na grafické uživatelské rozhraní a API založených na technologiích XML, .NET. Je postavené pro práci s vektorovou grafikou, bude využívat schopností moderních grafických adaptérů a technologie DirectX. Je to další krok vpřed od WinForms.

Evoluce vývojových nástrojů Microsoft

Pro vývoj aplikací pod Windows XP a Windows Server 2003 byl potřeba .NET Framework 3.0, Windows SDK + Visual Studio 2005 s rozšířením pro WPF.

Ve Windows Vista vystačíme pouze s Visual Studií. WPF lze samozřejmě nainstalovat také na Windows XP, nebo Windows Server 2003.

Využití XAML

Jestliže pohled do budoucnosti vyžaduje jiné zpracování dat, jde WPF právě touto cestou. Jedná se o deklarativní programování, které přináší možnost oddělit aplikační logiku od designu.

Jednou z novinek ve WPF je XAML, značkovací jazyk založený na XML, který slouží k tvorbě uživatelského rozhraní. Prakticky to znamená, že v jednom souboru je XAML kód a v druhém logika napsanou v některém z .NET jazyků).

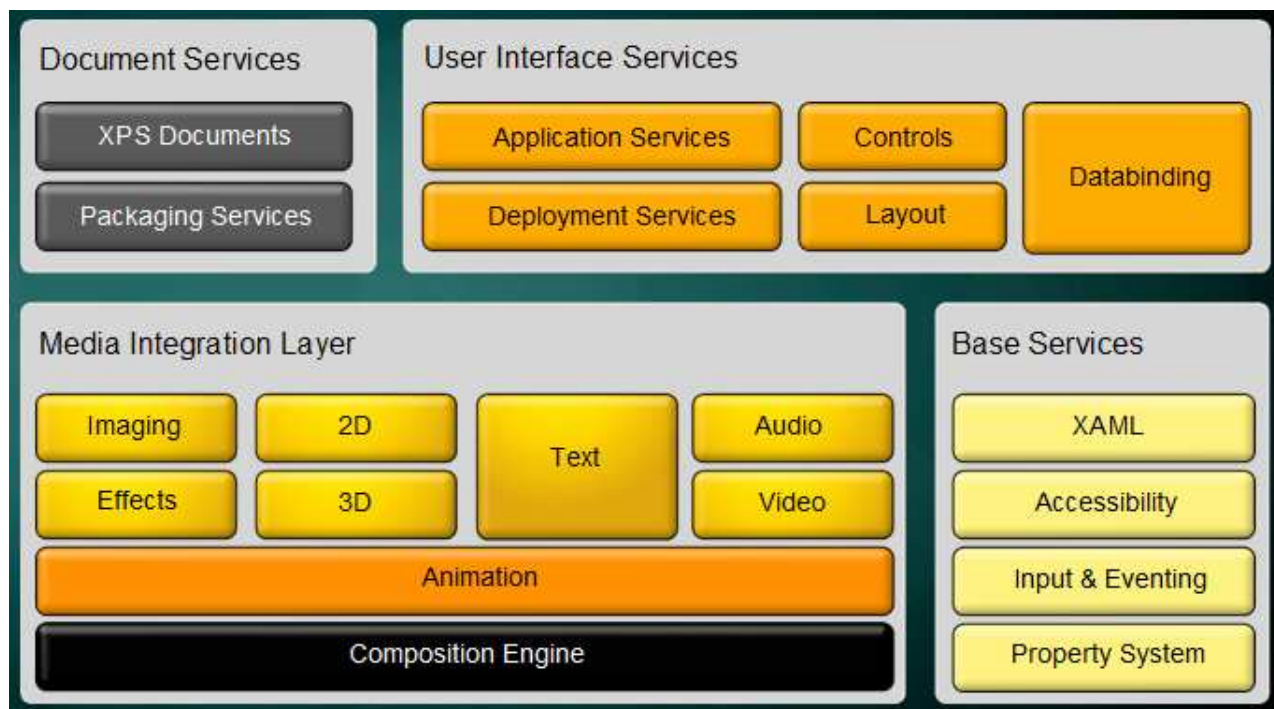
Grafika

WPF je konstruováno pro graficky bohaté aplikace s následujícími možnostmi:

- vektorové grafiky (graficky bezetrátová změna velikosti prvků založená na DPI,
- barevné gradienty, používání geometrických tvarů)
- animace (časové nebo založené na framech)
- multimédia (wma, mp3, avi, mpeg, wmv)
- efekty (stíny, záře, rozostření, průhlednost nebo zrcadlení)
- 3D grafika

Rendering – přesun zatížení na grafickou kartu

Mohlo by se zdát, že takovéto rozšíření grafických možností zvětší zatížení procesoru a tím se sníží celková rychlost běžících aplikací. Většinou tomu tak nebude. WPF běží na DirectX, veškerá grafika je tak renderována na grafické kartě. Grafická karta hardwarově musí podporovat vše od DirectX 7 po DirectX 9 a příslušné verze dalších softwarů., jinak může nastat situace, kdy je aplikace částečně renderována na GPU a částečně na CPU. V nejhorším případě bude aplikace celá renderována na procesoru, tehdy dojde k maximálnímu zpomalení aplikace a PC (v závislosti na síle procesoru).



Skladba komponent WPF

Data binding

Obecně znamená převod z abstraktních datových typů, zpráv a operací do konkrétní fyzické reprezentace zpráv. Jakmile se změní hodnota zdrojové vlastnosti, projeví se zároveň i v provázaných (nabíndovaných) vlastnostech a naopak. Pokud tedy v provázaných vlastnostech dojde ke změně, tato změna se projeví zároveň ve zdroji.

Je to proces propojení UI (uživatelského rozhraní) s aplikační logikou. Účelem je snadná synchronizace dat s vizuální reprezentací, která reflektuje na změny v UI i v datovém zdroji.

Pro definování konkrétních aspektů operací se používají elementy

Binding elementy

Cíl (Binding Target)

Libovolná vlastnost WPF elementu

Zdroje dat (Binding Source)

CLR objekt

WPF Element

ADO.NET (databáze...)

XML

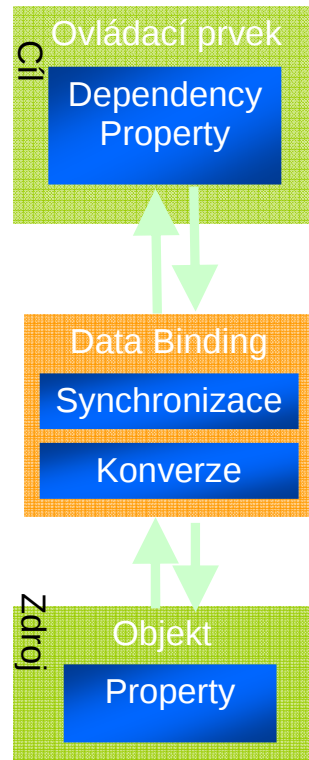
Modely

One Time, One Way, Two Way, One Way To Source

Dynamika

Master-Detail scénáře

Value Converter



Znázornění toku dat mezi zdrojem a UI

CLR (Common Language Runtime) poskytuje vestavěnou podporu pro interoperabilitu programovacích jazyků. Tato podpora nezaručuje, že kód napsaný vývojářem v jednom programovacím jazyce bude použitelný také pro vývojáře v jiném programovacím jazyce. Takové jazyky musí splňovat určitou základní množinu rysů - specifikaci společného jazyka - CLS.

CLS (Common Language Specification) je navržena tak, aby postihovala co nejvíce jazykových konstrukcí používaných vývojáři a přitom aby pokryla co nejvíce programovacích jazyků. Patří sem jazyky CLS-compliant (kompatibilní s CLS).

WSDL (Web Services Description Language) - METADATA

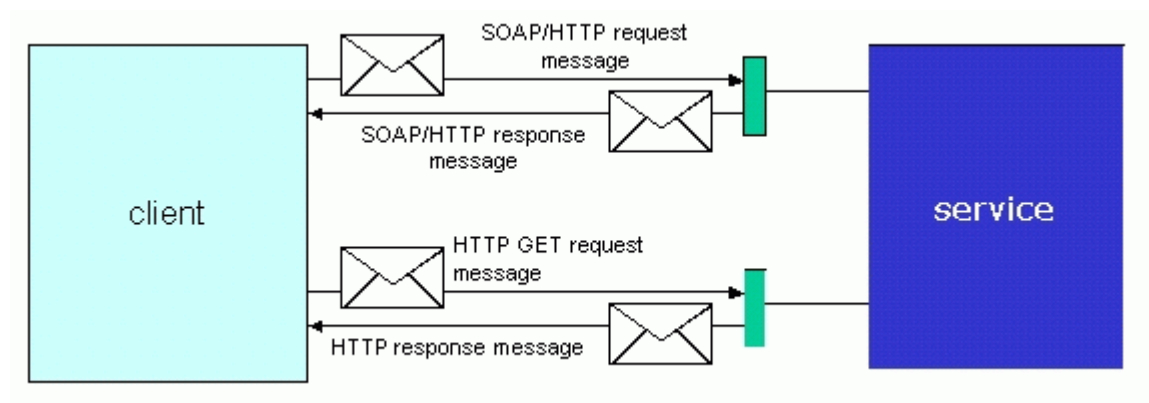
Je určený pro popis webových služeb a přístupu k nim. WSDL využívá XML pro popis síťových služeb (v zájmu standardizace komunikačních protokolů a formátů zpráv) jako kolekce komunikačních bodů schopných výměny zpráv. WSDL popisuje distribuované systémy a služby a tuto dokumentaci používá jako návod pro automatizaci detailů obsažených komunikačních aplikacích a prvcích.

WSDL dokument definuje služby jako kolekci koncových bodů v síti nebo protokolů. Abstraktní definice koncového bodu a zpráv je oddělena od jejich konkrétní podoby nebo datových vazeb. Podle filozofie XML oddělení kódu a dat. To umožňuje opětovné použití abstraktních popisů. Port je definován přiřazením síťové adresy s opětovně použitelnou vazbou.

Kolekce portů definuje službu. WSDL je definován následujícími elementy síťových služeb:

- Types - container pro definici typu dat používající nějaký typový systém (např. XSD).
- Message - abstraktní typová definice předávané zprávy.
- Operation - abstraktní popis akce podporované službou.
- Port Type - množina operací podporovaných jedním nebo více uzly.
- Binding - použitý protokol a určení formátu dat pro portu.
- Port - uzel definovaný jako kombinace binding a síťové adresy.

Service – přenosová cesta souvisejících uzlů.



WSDL terminologie popisující webovou službu – přenos dat

4.Ukázková aplikace

4.1 Zadání

4.1.1 Požadavky zadání

Vytvořit ukázkovou aplikaci ,která zpracovává naměřená a uveřejněná data na webu, obecně, která periodicky stahuje a ukládá zvolená data z různých WWW stránek vytvořených prostředí :

- HTML(nestandardizovaná forma dat v zdrojovém kódu)
- XML
- RSS

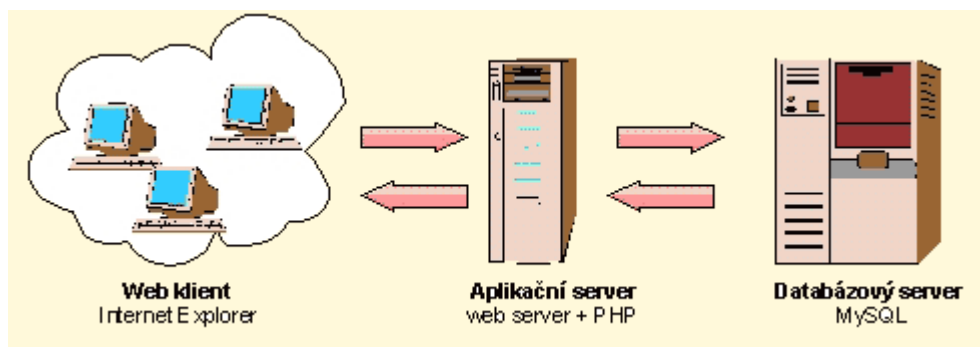
4.1.2 Rozbor

Požadavky na aplikaci

- snadná obsluha
- snadná instalace a nasazení na obslužném PC
- variabilita, možnost výběru, změny zdrojů
- variabilita výstupu - výběr, třídění dat
- kritéria výstupu vybírat ze seznamu
- kontrola připojení, nebo změna struktury zdroje
- vhodné uložení dat – databáze
- zabezpečit opakované spouštění aplikace

4.2 Řešení

Po rozboru požadavků jsem jako řešení pro vytvoření aplikace, zvolil technologii PHP, třívrstvou webovou aplikaci typu klient/server. Jelikož se jedná o stahování dat na jednu lokální stanici, hardwarově jsou všechny vrstvy na jednom místě - localhost.



1.vrstva – klient , uživatelské rozhraní, prohlížeč MS Internet Explorer, který umožní uživateli zadat

- vstupní požadavky(WWW servery, podmínky pro stahovaná data)
- podmínky pro výběr a třídění stažených dat
- výstupní požadavky(výběr, třídění, zobrazení stažených dat)

2.vrstva – spojuje uživatele s daty, která jsou na serveru. K datům přistupovat ještě z jiných stanic na síti. V našem případě se lze připojit zadáním IP adresy počítače , na kterém běží PHP, server a databáze.

3.vrstva – databáze MYSQL, přístup a distribuce dat

4.3 Programové vybavení a nastavení

Pro instalaci PHP na pracovní stanici s operačním systémem Windows XP Professional jsem použil freeware balíček EasyPHP 2.0b1.

Standardně se komponenty nainstalují do složky C:\Program Files\EasyPHP 2.0b1.

Adresář- kořen WWW dokumentů je potom C:\Program Files\EasyPHP 2.0b1\www

Po instalaci je možné zvolit jako výchozí jazyk češtinu

Součástí této beta verze jsou :

4.3.1 PHP 5.2.0

PHP je serverový skriptovací jazyk, který umožňuje dynamicky měnit, upravovat, spravovat obsah webových stránek a přizpůsobit obsah konkrétním uživatelům.

4.3.2 Nastavení PHP

Změna konfigurace se dělá v souboru C:\Program Files\EasyPHP 2.0b1\php5\php.ini, který je potřeba překopírovat do C:\Windows.

Pro funkčnost formulářů aplikace je zapotřebí změnit direktivu povolení globálních proměnných z URL adresy – z `register_globals = Off` na `register_globals = On`, dále pro spolupráci s db. MySql odkomentovat řádek `extension=php_mysql.dll`.

4.3.3 Web server Apache 2.2.3

Umožňuje přístup a sdílení dat na serveru pro jiné stanice na základě jejich požadavku.

4.3.4 Nastavení Apache

Pro naše potřeby není potřeba nic měnit, změna například kořen dokumentů se mění direktivou `DocumentRoot` v souboru `httpd.conf` - upravit textovým editorem (třeba PSPad), uložit a restartovat server.

Z lokálního PC budeme volat sami sebe z příkazové řádky Explorera - `http://127.0.0.1`, nebo `http://localhost`

4.3.5 Databáze MySql 5.0.27

Světově nejrozšířenější a nejvyužívanější databáze. Rapidně rozšiřuje možnosti PHP pro práci s daty

4.3.6 Nastavení MySql

Přes PHPMyAdmin (dostupný na `http://localhost/mysql`). Admin je `root@localhost`, dají se také vytvářet noví uživatelé a databáze. Ke správě databáze jsem použil HeidiSQL.

4.3.7 PhpMyAdmin 2.9.1.1

4.3.8 SQLiteManager 1.2.0

4.3.9 Další pomocný freeware - HeidiSQL

Tímto nástrojem jsem nahradil funkce phpMyAdminu 2.9.1.1 a SQLiteManageru 1.2.0. HeidiSQL je komplexní podpůrné grafické rozhraní pro práci s databází.

- umožňuje zřizovat, rušit připojení k různým databázím
- vytvářet, rušit uživatele
- vytvářet, měnit tabulky, přidávat primární klíče
- importovat data do tabulek
- exportovat data

4.3.10 JavaScript

Viz kapitola 2.4

4.4 Perioda stahování dat

Za účelem pravidelného spouštění skriptu PHP, jsem využil interní funkce Windows XP Naplánované úlohy. Pro spuštění skriptu PHP v Plánovači úloh, je zapotřebí vytvořit soubor .bat.

V prostředí Linuxu bychom využily nástroje cron, fcron, nebo Vixie cron(součást instalace Linux). Spuštění úlohy by znamenalo vložit skript do jednoho ze souborů /etc/cron.daily, /etc/cron.hourly, /etc/cron.weekly a /etc/cron.monthly. Je vhodné stahovat objemnější data v době malého zatížení sítě.

4.5 Tvorba aplikace

4.5.1 Databáze meteo

Tato část skriptu vytvoří novou databázi. K tomu jsou zapotřebí adminovská práva na MySQL. Název sice evokuje stahování informací o počasí, ale v databázi jsou data ze všech možných různých zdrojů.

```
$spojeni=mysql_connect("SQL_HOST","SQL_USERNAME","SQL_PASSWORD");//(server, uživatel,  
heslo db=heslo uživatel)  
$vysledek = mysql_query("CREATE DATABASE IF NOT EXISTS $db")  
or exit ("Spatny dotaz: " . mysql_error());
```

4.5.2 Tabulky v databázi

Primární klíč tabulky je sloupec s unikátní hodnotou pro každý záznam v tabulce. Těmito klíči mohou být vytvořeny navzájem relace-spojení mezi tabulkami.

Tabulky obsahují indexy pro efektivnější a rychlejší hledání záznamů, při aktualizaci zpomalují práci s databází. Indexace zabraňuje dublování záznamů. Nový a stávající záznam nemohou mít stejnou hodnotu.

4.5.3 Tabulka Meteo_url

Kód vytvoří tabulku Meteo_url, která obsahuje seznam www stránek, ze kterých chceme stahovat data.

Tabulku tvoří sloupce :

url_id	-	primární klíč
url	-	obsahuje www adresu včetně http://
url_typ	-	typ kódu www stránky (html, xml, rss, text)
zpozdeni	-	údaj v hodinách(aby stahovač věděl, jestli stahuje dnešní, nebo
včerejší		data

```
mysql_query("CREATE TABLE IF NOT EXISTS "Meteo_url" ("url_id" PRIMARY KEY,  
                                                    "url" VARCHAR (255),  
                                                    "url" VARCHAR (255),  
                                                    "typ_url" CHAR(10)),  
                                                    "zpozdeni" SHORTINT)
```

4.5.4 Tabulka Meteo_zdroje

Kód vytvoří tabulku Meteo_zdroje(je shodná s objektem Meteo_zdroj), která obsahuje seznam www stránek, ze kterých chceme stahovat data.

Tabulku tvoří sloupce :

id_zdroje	-	primární klíč
url_id	-	vazba na tabulku Meteo_url
typ_podminky-		která podmínka bude použita
		reg(regulární výraz), text_pred_za(text před a za daty), xpath – pro xml
podminka1	-	regulární výraz, text před požadovanými daty, xpath
podminka2	-	případný text za požadovanými daty
typ_dat	-	údaj ze vstupního formuláře, např. srážky, teplota, cena BMW
místo	-	údaj ze vstupního formuláře, např. Praha, ČR, Veleslavín, Mars
koeficient	-	jakým číslem násobíme stažená data(účel pro převodjednotek)
defaultvalue	-	pokud ve zdroji požadovaná data nejsou, záznam z konkrétního data má hodnotu NULL(neohlásí chybu), nebo má hodnotu zadanou do vstupního formuláře

```
mysql_query("CREATE TABLE IF NOT EXISTS "Meteo_zdroje" ("id_zdroje" PRIMARY KEY,
"url_id" INT,
"typ_podminky" CHAR (10),
"podminka1" VARCHAR (255),
"podminka2" VARCHAR (255),
"typ_dat" CHAR(50)),
"misto" CHAR(150)),
"koeficient" FLOAT
"hodnota" FLOAT)
```

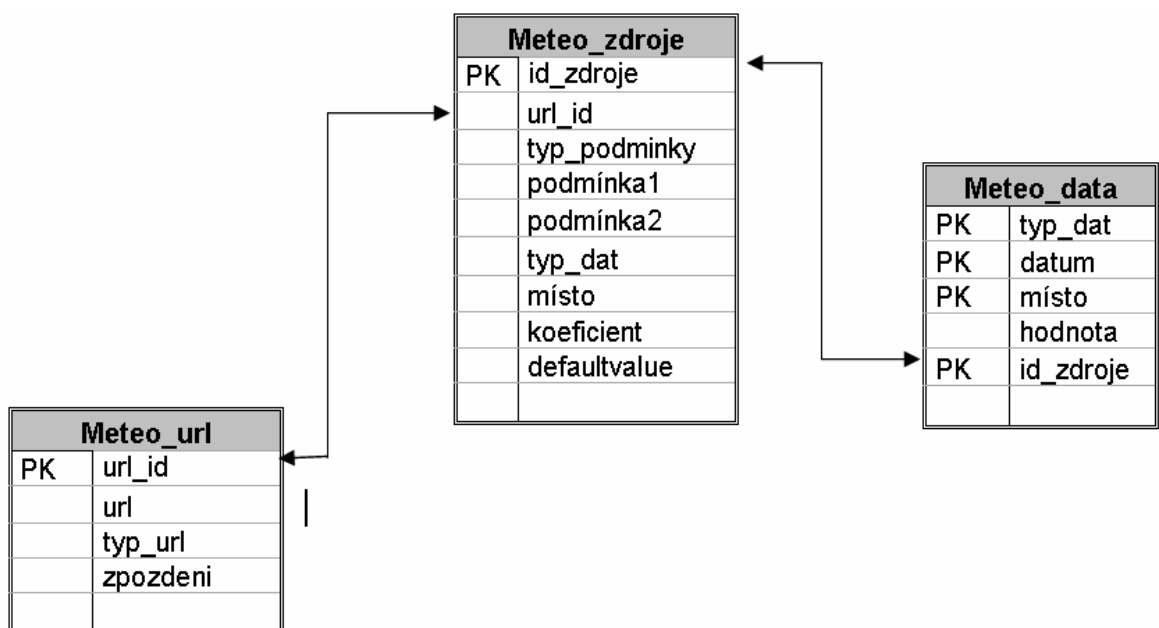
4.5.5 Tabulka Meteo_data

Kód vytvoří tabulku Meteo_data. Vazba na tabulku Meteo_zdroje reprezentují sloupce typ_dat, datum, místo, id_zdroje, tyto hodnoty jsou navíc primárními klíči (hodnoty jsou jedinečné) a proto je možné stahovat jen 1 x denně.

typ_dat	-	primární klíč údaj ze vstupního formuláře
datum	-	primární klíč datum
misto	-	primární klíč údaj ze vstupního formuláře
hodnota	-	stažená data
id_zdroje	-	primární klíč(integer)

```
mysql_query("CREATE TABLE IF NOT EXISTS "Meteo_data" ("id_zdroje" INT,
"typ_dat" CHAR (50),
"datum" DATE,
"misto" CHAR (150),
"hodnota" FLOAT)
```

4.5.6 Diagram struktury databáze



4.6 Popis souborů aplikace

4.6.1 Config.php

Je soubor, který obsahuje konstanty připojení k databázi. Výhodou je, že při změně názvu, nebo umístění databáze se změna provede pouze na jednom místě pro všechny skripty.

```
define("SQL_HOST","localhost");
define("SQL_DBNAME","meteo");
define("SQL_USERNAME","CB");
define("SQL_PASSWORD","localhost")
```

4.6.2 Soubor stazeni_dat.php

Soubor stažení_dat.php je hlavním strojem aplikace. Nejprve definuji připojení místní sítě k internetu (na stroji z kterého se skript spouští-přes proxy, nebo bez proxy).

Funkce OsetriChybu vypisuje chyby spojení s databází(mysql_error), případně že nebyla nalezena data ve stažené stránce. Ta je posléze volána v následném kódu skriptu. Text chyby se připíše na novou řádku souboru, jestliže soubor neexistuje(při první spuštění), je vytvořen.

Hlavní program se připojí k databázi. V případě nezdaru, vypíše chybu do souboru txt.

Následně z tabulky Meteo_url vybírá v cyklu www stránku po stránce (nastaví objekt Meteo_Url) a stránku zpracuje. Na každé stránce podle klíče Url_id sestaví dotazy, podle podmínek, které jsme zadali přes vstupní formulář vybere data ze stránky. Tyto hodnoty potom uloží do tabulky Meteo_data.

Hlavní program pracuje s již deklarovanými objekty Meteo_Url a Meteo_Zdroj a funkcemi.

Jednu zdrojovou stránku definuje Objekt Meteo_Url

Operace každou stránkou prezentují funkce-výběr konkrétní stránky(MeteoUrl_Nastav), její stažení(MeteoUrl_Stahni) pomocí interní fce. File_get_contents).

Funkce MeteoUrl_nastavDatum přidá parametr Zpoždění k aktuálnímu datu, dopočte a přiřadí výsledek stažené stránky.

Podmínky pro stažení jednoho údaje z konkrétní stránky definuje Objekt Meteo_Zdroj

Funkce MeteoZdroj_Nastav, nastaví objekt Meteo_Zdroj na vybrané stránce (podle jednoho přečteného řádku nastaví objekt). Funkce MeteoZdroj_Precti definuje tři způsoby výběru dat. Výběr typu podmínky následuje v kódu a závisí na zaškrtnutí volby select ve vstupním formuláři. Funkce MeteoZdroj_UlozData uloží jeden údaj ze stažené stránky, který stáhnu funkcí MeteoZdroj_Precti, do databáze.

```

<?php

define("PROXY",'proxy.mvcr.cz:8080'); // nebo define("PROXY",'172.16.2.12:8080') nebo
define("PROXY",false)
include ("./config.php");

function OsetriChybu($TextChyby,$Skonci=false)
{
    $F=fopen("./chyby.txt","ab"); // otevri soubor pro zapis na konec souboru
    fwrite($F,"Chyba: ".$TextChyby."\r\n"); // zapis chyby
    fclose($F); // zavri soubor

    echo($TextChyby."\r\n");
    if($Skonci) exit();
}

//=====
//===== objekt Meteo_Url =====
//=====
//
// Vlastnosti objektu
// - zhruba odpovídají jednomu řádku tabulky Meteo_url
//
// Funkce pro práci s daty v objektu:
// MeteoUrl_Nastav
// - přečte z sql tabulky Meteo_url jeden řádek a nastaví svoje vlastnosti
// MeteoUrl_Stahni
// - stažené z webu jednu stránku s daty
// MeteoUrl_NastavDatum
// - zjistí, ze kdy jsou na stránce hodnoty
//

class Meteo_Url
{
    public $IdUrl, $Url, $TypUrl, $Zpozdeni;
    public $Datum;
}

function MeteoUrl_Nastav($Objekt_MeteoUrl,$SqlData)
{
    /* tady je nastavení $Objekt_MeteoUrl->IdUrl, $Objekt_MeteoUrl->Url, $Objekt_MeteoUrl->TypUrl,
    $Objekt_MeteoUrl->Zpozdeni z databáze
    $SqlData byl získán pomocí mysql_fetch_array, obsahuje jeden řádek tabulky meteo_url
    fce Nastav přečte 1 řádek z tabulky a uloží jej do objektu */
    $Objekt_MeteoUrl->IdUrl = $SqlData["url_id"];
    $Objekt_MeteoUrl->Url = $SqlData["url"];
    $Objekt_MeteoUrl->TypUrl = $SqlData["typ_url"];
    $Objekt_MeteoUrl->Zpozdeni = $SqlData["zpozdeni"];
}

function MeteoUrl_Stahni($Objekt_MeteoUrl)
{
    //--- pokud není proxy
    if(!PROXY) return(file_get_contents($Objekt_MeteoUrl->Url));
    //--- pokud je proxy
    $opts = array(
        'http'=>array(
            'request_fulluri' => true,
            'proxy' => 'tcp://'.PROXY));
    $context = stream_context_create($opts);
    return(file_get_contents($Objekt_MeteoUrl->Url,false,$context));
}

function MeteoUrl_NastavDatum($Objekt_MeteoUrl,$ObsahStránky)
{
    // nastav datum, ze kterého jsou stažená data
}

```

```

// -----
// univerzální algoritmus, který by měl fungovat
// pro většinu stránek

$Objekt_MeteoUrl->Datum=time()-$Objekt_MeteoUrl->Zpozdeni*60*60;

// -----
// zde bude případná oprava pro speciální případy,
// univerzální algoritmus selže
}

//=====
//===== objekt Meteo_Zdroj =====
//=====
//
// Vlastnosti objektu
// - zhruba odpovídají jednomu řádku tabulky Meteo_zdroje
//
// Funkce pro práci s objektem:
// MeteoZdroj_Nastav
// - přečte z sql tabulky Meteo_zroje jeden řádek a nastaví svoje vlastnosti
// MeteoZdroj_PrectiData
// - ze stažené www stránky zjistí jeden údaj (vlastnost Hodnota),
// například maximální teplotu v Plzni ze dne 12. prosince 2009
// MeteoZdroj_UlozData
// - uloží zjištěnou hodnotu do tabulky Meteo_data
//
// Pomocné funkce:
// MeteoZdroj_PrectiData_TextPred_TextZa
// MeteoZdroj_PrectiData_Reg
// MeteoZdroj_PrectiData_XPath
//

class Meteo_Zdroj
{
    public $IdZdroje, $UrlId, $TypPodminky, $Podminka1, $Podminka2, $TypDat, $Misto,
    $Koeficient,$Defaultvalue;
    public $Hodnota;
}

function MeteoZdroj_Nastav($Objekt_MeteoZdroj,$SqlData)
{
    /*
        tady je nastavení $Objekt_MeteoZdroj->IdUrl, $Objekt_MeteoZdroj->Url,
        $Objekt_MeteoZdroj->TypUrl, $Objekt_MeteoZdroj->Zpozdeni z databáze
        $SqlData byl získán pomocí mysql_fetch_array, obsahuje jeden řádek tabulky meteo_url */

    $Objekt_MeteoZdroj->IdZdroje = $SqlData["id_zdroje"];
    $Objekt_MeteoZdroj->UrlId = $SqlData["url_id"];
    $Objekt_MeteoZdroj->TypPodminky = $SqlData["typ_podminky"];
    $Objekt_MeteoZdroj->Podminka1 = $SqlData["podminka1"];
    $Objekt_MeteoZdroj->Podminka2 = $SqlData["podminka2"];
    $Objekt_MeteoZdroj->TypDat = $SqlData["typ_dat"];
    $Objekt_MeteoZdroj->Misto = $SqlData["misto"];
    $Objekt_MeteoZdroj->Koeficient = $SqlData["koeficient"];
    $Objekt_MeteoZdroj->Defaultvalue = $SqlData["defaultvalue"];
    // Nejvyšší teplota: '<table class="hilo"><tr><th>Hi:</th> <td>'&#176;</td></tr>'
    // Nejnižší teplota: '</td></tr><tr><th>Lo:</th> <td>'&#176;</td></tr></table>'
    // Relativní vlhkost den: '<span class="day">Day: </span>[^\<]*humidity ([0-9]+)'

}

function MeteoZdroj_PrectiData_TextPred_TextZa($Objekt_MeteoZdroj,$ObsahStranky)
{
    // hledej pozici začátku

```

```

        $I=strpos($ObsahStranky,$Objekt_MeteoZdroj->Podminka1);
        if($I===false) return(false);
        $I=$I+strlen($Objekt_MeteoZdroj->Podminka1);
        // hledej pozici konce
        $J=strpos($ObsahStranky,$Objekt_MeteoZdroj->Podminka2,$I);
        if($J===false) return(false);
        //
        $T=substr($ObsahStranky,$I,$J-$I);
        return($T);
    }

function MeteoZdroj_PrectiData_Reg($Objekt_MeteoZdroj,$ObsahStranky)
{
    if(ereg($Objekt_MeteoZdroj->Podminka1,$ObsahStranky,$Vysledky) == false)
        return(false);
    return($Vysledky[1]);
}

function MeteoZdroj_PrectiData_XPath($Objekt_MeteoZdroj,$ObsahStranky)
{
    $xml = simplexml_load_string($ObsahStranky);
    if($xml === false) // pokud Obsahstranky neni korektní XML
    {
        return(false);
    }

    $HledanaHodnota = $xml->xpath($Objekt_MeteoZdroj->Podminka1); // metoda xpath vraci pole
    var_dump($HledanaHodnota);
    if(count($HledanaHodnota)<1) // pokud daný výraz xpath nebyl na stránce nalezen
(0 výsledků)
    {
        return(false);
    }

    return($HledanaHodnota[0]); // vratim prvni nalezeny vyraz
}

function MeteoZdroj_PrectiData($Objekt_MeteoZdroj,$ObsahStranky)
{
    if($Objekt_MeteoZdroj->TypPodminky=="pred_zat")
    {
        $X=MeteoZdroj_PrectiData_TextPred_TextZa($Objekt_MeteoZdroj,$ObsahStranky);
    }

    else if($Objekt_MeteoZdroj->TypPodminky=="reg")
    {
        $X=MeteoZdroj_PrectiData_Reg($Objekt_MeteoZdroj,$ObsahStranky);
        //deklarace výše generované stránky funkcemi PrectiData,ObsahStranky
    }

    else if($Objekt_MeteoZdroj->TypPodminky=="xpath")
    {
        $X=MeteoZdroj_PrectiData_XPath($Objekt_MeteoZdroj,$ObsahStranky);
    }

    if($X===false)
    {
        return(false);
    }

    $Objekt_MeteoZdroj->Hodnota = $Objekt_MeteoZdroj->Koeficient * (float)($X);
    return(true);
}

function MeteoZdroj_UlozData($Objekt_MeteoZdroj,$MeteoUrl_Datum)
{

```

```

$SqlPrikaz="REPLACE meteo_data SET typ_dat=\"\".$Objekt_MeteoZdroj->TypDat.\"\"
        .\", datum=\"\".date(\"Y-m-d\",$MeteoUrl_Datum).\"\"
        .\", misto=\"\".$Objekt_MeteoZdroj->Misto.\"\"
        .\", hodnota=\"\".$Objekt_MeteoZdroj->Hodnota
        .\", id_zdroje=\"\".$Objekt_MeteoZdroj->IdZdroje;
var_dump($SqlPrikaz);
mysql_query($SqlPrikaz);
    }

//=====
//===== hlavni program =====
//=====

//-----
//----- připojení k databázi -----
//-----

if(!mysql_connect(SQL_HOST, SQL_USERNAME, SQL_PASSWORD))
    OsetriChybu("Nelze se připojit k MySQL: " . mysql_error()
        ."\nMozne priciny:"
        ."\n - spatne nastaveny config.php (jmeno databaze, uzivatel, heslo)"
        ."\n - MySQL na serveru ".SQL_HOST." nebezi"
        ."\n - nezalozeny uzivatel ".SQL_USERNAME
        ."\n - spatne heslo uzivatele"
        ."\n - ...",true);

if(!mysql_select_db(SQL_DBNAME))
    OsetriChybu("Nelze se přepnout do databáze ".SQL_DB." : " . mysql_error(),true);

//-----
//----- stáhnou seznam URL -----
//-----

// položím dotaz do databáze
$Result=mysql_query("SELECT * FROM Meteo_url");

if(!$Result)
    OsetriChybu("Selhal SQL dotaz 'SELECT * FROM Meteo_url': " . mysql_error(),true);

// Meteo_Url_List je na začátku prázdné pole
$Meteo_Url_List=array();

// pro každý řádek tabulky Meteo_url
while($Data = mysql_fetch_array($Result,MYSQL_ASSOC))
{
    $Url=new Meteo_Url(); // vytvořím nový objekt typu Meteo_Url
    MeteoUrl_Nastav($Url,$Data); // zavolám metodu (funkci) objektu Url (typ objektu je
    Meteo_Url)
    $Meteo_Url_List[] = $Url; // vložím vyplněný objekt na konec pole Meteo_Url_List
}

// konec sql dotazu - uvolním paměť
mysql_free_result($Result);

//-----
//----- zpracuj jednotlivé zdroje -----
//-----

// pro každou Url (typu Meteo_Url)
foreach($Meteo_Url_List as $Url)
{
    // stažení obsahu jedné www stránky do textové proměnné ObsahUrl - volání metody Stahni objektu Url
    (typ Meteo_Url)
    $ObsahUrl=MeteoUrl_Stahni($Url);
    MeteoUrl_NastavDatum($Url,$ObsahUrl);
}

```

```

// pokud se stažení nepodaří - zbytek cyklu neprováděj, jdi na další Url
if($ObsahUrl===false) continue;

// najdi pro danou staženou stránku všechny zdroje z tabulky Meteo_zdroje (sql dotaz)
// id_url mám uloženou v objektu Url (vlastnost objektu)
// na jedné stránce mám obvykle víc zdrojů (maximální teplota v Praze, v Kladně, minimální
teplota v ...)
$Result=mysql_query('SELECT * FROM Meteo_zdroje WHERE url_id='.$Url->IdUrl);
if(!$Result)
    OsetriChybu("Selhal SQL dotaz 'SELECT * FROM Meteo_url': " . mysql_error(),true);

// pro každý řádek tabulky Meteo_zdroje (které se vztahuje k dané stránce)
while($SqlData = mysql_fetch_array($Result,MYSQL_ASSOC))
{
    // Zdroj je objekt typu Meteo_Zdroj, jeho vlastnosti odpovídají jednomu řádku tabulky
    Meteo_zdroje
        $Zdroj = new Meteo_Zdroj();

    // načtu řádek tabulky do objektu Zdroj
    MeteoZdroj_Nastav($Zdroj,$SqlData);

    // na stažené stránce najdu data odpovídající dané podmínce
    if(MeteoZdroj_PrectiData($Zdroj,$ObsahUrl))
    {
        // pokud je to povede - uloží stažená data do tabulky meteo_data
        MeteoZdroj_UlozData($Zdroj,$Url->Datum);
    }
    else
    {
        // pokud se to nepovede, zapíšu informaci do logu; pokud v logu bude více
        // hlášek k jednomu zdroji, tak je potřeba zkontrolovat, jestli se www stránka
nezměnila
        OsetriChybu("Vadný zdroj dat ".$Url->IdUrl.' '.$Zdroj->IdZdroje,false);
    }
}

// konec čtení dat z sql tabulky - uvolním paměť
mysql_free_result($Result);
}

?>

```

4.6.3 Soubor add1.php

Vstupní formulář pro vyplnění nových kritérií pro výběr dat. PHP kód vloží do formuláře už zadaný seznam www stránek, nebo lze přidat nový zdroj(stránku). V případě výběru již stahované stránky nejsou přístupná pole časové korekce (čas stahování/datum aktuálnosti dat) a typu stránky. To je ošetřeno Javascriptem.

Vpravo nahoře jsou odkazy na editaci/mazání stávajících zdrojů(www stránek), nebo dotazů do nich, případně odkaz zpět na přidání nových záznamů, nebo výpis všech stávajících.

Nový dotaz ke stažení hodnoty z webu
(V případě, že nevíte jak zadávat dotaz, přečtěte si [HELP](#) jak na to)

WWW stránka:

☒ Html ☐ XML ☐ RSS Zpoždění: hodin

Typ podmínky přidání dalších dat do databáze v závislosti na zdroji dat:
☒ Text před, za hodnotou ☐ Regulární výraz ☐ Hodnota ze souboru XML

Podmínka 1:
Podmínka 2:

Typ dat:
Místo:
Koeficient:
Default:

```
<?php
```

```
include ("./config.php");
```

```
//-----  
//----- připojení k databázi -----  
//-----
```

```
if(!mysql_connect(SQL_HOST, SQL_USERNAME, SQL_PASSWORD))  
    die("Nelze se připojit k MySQL: " . mysql_error())  
    ."\nMozne priciny:"  
    ."\n - spatne nastaveny config.php (jmeno databaze, uzivatel, heslo)"  
    ."\n - MySQL na serveru ".SQL_HOST." nebezi"  
    ."\n - nezalozeny uzivatel ".SQL_USERNAME  
    ."\n - spatne heslo uzivatele"  
    ."\n - ...");
```

```
if(!mysql_select_db(SQL_DBNAME))  
    die("Nelze se přepnout do databáze ".SQL_DB." : " . mysql_error());
```

```
mysql_query("SET NAMES 'cp1250';");
?>
<HTML>

<HEAD>

<TITLE>Vkládání</TITLE>

</HEAD>

<BODY BGCOLOR="WHITE">

  <div align="right"><a href="konkretni_dotaz.php">výpis dat</a> &nbsp; &nbsp; &nbsp; <a
href="del1.php">smazat/opravit zdroj/url</a> &nbsp; &nbsp; &nbsp; <a href="add1.php">přidat nový
zdroj/url</a></div>

  <FORM ACTION="insert.php" METHOD=POST NAME="formular">    <!--zacatek formulare, k obsluze
se použije soubor XXX.php-->

    <TABLE ALIGN="CENTER">

      <TR>
        <TD ALIGN="CENTER" COLSPAN=2><font size="5"><B>Nový dotaz ke stažení hodnoty z
webu</B></font></TD>
      </TR>
      <TR>
        <TD ALIGN="CENTER" COLSPAN=2>(V případě, že nevíte jak zadávat dotaz, přečtěte si <a
href="help.html">HELP</a> jak na to)</TD>
      </TR>
      <TR>
        <TD>&nbsp;</TD>

    </TR>

    <TR>
      <TD ALIGN="Left" valign="top">
        WWW stránka:
      </TD>
      <TD ALIGN="left" COLSPAN=2>
        <Select Name="url_id" onchange="Zesedni();">
          <Option value="novy">-- nová www (xml/rss) stránka --</Option>
          <?php
            $Result=mysql_query('SELECT * FROM Meteo_url ORDER BY url');
            while($Data = mysql_fetch_array($Result,MYSQL_ASSOC))
              echo("<option value=\"". $Data['url_id'].\"\">" .htmlspecialchars($Data['url'])."</option>\r\n");
            mysql_free_result($Result);
          ?>

        </Select><br>
        <INPUT TYPE=TEXT SIZE=100 MAXLENGTH=255 NAME="url" VALUE="http://..." ><br>

        <INPUT TYPE=RADIO NAME="typ_url" VALUE="html" CHECKED="CHECKED" >Html
        <INPUT TYPE=RADIO NAME="typ_url" VALUE="xml"> XML
        <INPUT TYPE=RADIO NAME="typ_url" VALUE="rss"> RSS
        &nbsp; &nbsp; &nbsp; &nbsp;
        Zpoždění: <INPUT TYPE=TEXT SIZE="3" MAXLENGTH=3 NAME="zpozdeni" VALUE="8" >
        hodin<br>
      </TR>
      <TR>
        <TD>&nbsp;</TD>

    </TR>

    <TR>
      <TD ALIGN="Left" COLSPAN=2>
        Typ podmínky přidání dalších dat do databáze v závislosti na zdroji dat.
      </TD>
    </TR>
    <TR>
      <TD ALIGN="CENTER"></TD>

  </BODY>
</HTML>
```

```

        <TD ALIGN="Left">
        <INPUT TYPE=RADIO NAME="typ_podminky" VALUE="pred_za" CHECKED="CHECKED" > Text
        před, za hodnotou
        <INPUT TYPE=RADIO NAME="typ_podminky" VALUE="reg"> Regulární výraz
        <INPUT TYPE=RADIO NAME="typ_podminky" VALUE="xpath"> Hodnota ze souboru XML<br>

    </TD>
</TR>

<TR>
<TD>&nbsp;</TD>
</TR>

<TR ALIGN="CENTER">
<TD ALIGN="LEFT">
    Podmínka 1:
</TD>
<TD ALIGN="Left">
    <INPUT TYPE=TEXT SIZE="80" MAXLENGTH=255 NAME="podminka1" VALUE="" >
</TD>
</TR>

<TR ALIGN="CENTER">
<TD ALIGN="LEFT">
    Podmínka 2:
</TD>
<TD ALIGN="LEFT">
    <INPUT TYPE=TEXT SIZE="80" MAXLENGTH=255 NAME="podminka2" VALUE="" >
</TD>
</TR>

<TR>
<TD>&nbsp;</TD>
</TR>

<TR ALIGN="CENTER">
<TD ALIGN="LEFT">
    Typ dat :
</TD>
<TD ALIGN="LEFT"><INPUT TYPE=TEXT SIZE="35" MAXLENGTH=50 NAME="typ_dat"
    VALUE="teplota" >
</TD>
</TR>

<TR ALIGN="CENTER">

<TD ALIGN="LEFT">
    Místo :</TD>
<TD ALIGN="LEFT"><INPUT TYPE=TEXT SIZE="45" MAXLENGTH=150 NAME="misto" VALUE="" >
</TD>
</TR>

<TR ALIGN="CENTER">
<TD ALIGN="LEFT">
    Koeficient :</TD>
<TD ALIGN="LEFT"><INPUT TYPE=TEXT SIZE="5" MAXLENGTH=50 NAME="koeficient"
    VALUE="1" >
</TD>
</TR>
<TR>

<TD ALIGN="LEFT">
    Default :</TD>
<TD ALIGN="LEFT"><INPUT TYPE=TEXT SIZE="5" MAXLENGTH=150 NAME="defaultvalue"
    VALUE="NULL" >
</TD>

```

```

</TR>

    <TR COLSPAN=2>
    <TD ALIGN="CENTER">

    </TD>
    </TR>

    <TR>
    <TD>&nbsp;</TD>
    </TR>
    <TR>
    <TD ALIGN="CENTER" COLSPAN=4>

    <INPUT TYPE=SUBMIT VALUE="Pridej podminku na stahovaná data" NAME="Confirm">

    <INPUT TYPE=BUTTON VALUE="Zpet" onClick=Zpet()>
    </TD>
    </TR>

</TABLE>
</HTML>

```

```

<?
}
?>

```

```

<SCRIPT Language="javascript">

function Zesedni()
{
    var Sedive

    if(document.formular.url_id.selectedIndex==0)
        Sedive=false
    else Sedive=true

    document.formular.url.disabled=Sedive
    document.formular.typ_url[0].disabled=Sedive
    document.formular.typ_url[1].disabled=Sedive
    document.formular.typ_url[2].disabled=Sedive
    document.formular.zpozdeni.disabled=Sedive

}

//funkce v javascriptu pro tlačítko zpet
function Zpet()

{

    history.back();

}

</SCRIPT>

```

4.6.4 Soubor insert.php

Zpracuje formulář add1.php. Vloží do databáze podmínky pro vytvoření nového dotazu ke stahování dat z www stránky, případně přidá i novou stránku do prohledávaných zdrojů. Počítá nové primární klíče pro tabulky meteo_url a meteo_zdroje, kontroluje zda jsou vyplněny vstupní údaje formuláře add1.

```
<?php          //Soubor vložení stahování z nové url, přidání podmínek stahování

    Header("Expires: ".GMDate("D, d M Y H:i:s")." GMT"); //omezení k použití pro tlačítko zpet v prohlizeci
                                                //stránka hned ztrácí svou platnost
include ("./config.php");

//-----
//----- připojení k databázi -----
//-----

if(!mysql_connect(SQL_HOST, SQL_USERNAME, SQL_PASSWORD))
    die("Nelze se připojit k MySQL: " . mysql_error()
        ."\nMozne priciny:"
        ."\n - spatne nastaveny config.php (jmeno databaze, uzivatel, heslo)"
        ."\n - MySQL na serveru ".SQL_HOST." nebezi"
        ."\n - nezalozeny uzivatel ".SQL_USERNAME
        ."\n - spatne heslo uzivatele"
        ."\n - ...");

if(!mysql_select_db(SQL_DBNAME))
    die("Nelze se přepnout do databáze ".SQL_DB." : " . mysql_error());

//-----
//----- kontrola dat formuláře ----
//-----

function PreciZFormulare($Nazev,$TextChyby)
{
    if(!array_key_exists($Nazev,$_REQUEST)) //hledaný prvek,zdroj
        die($TextChyby);
    return($_REQUEST[$Nazev]);
}

$url_id=PreciZFormulare("url_id","Nevybrali jste nic ze seznamu");
if($url_id == "novy")
{
    $url=PreciZFormulare("url","Nevyplnili jste url noveho zdroje");
    if($url=="http://...") die("Nevyplnili jste url noveho zdroje");
    $typ_url=PreciZFormulare("typ_url","Nevyplnili jste typ_url");
    $zpozdeni=PreciZFormulare("zpozdeni","Nevyplnili jste zpozdeni");
}

$typ_podminky=PreciZFormulare("typ_podminky","Nevyplnili jste typ_podminky");
$podminka1=PreciZFormulare("podminka1","Nevyplnili jste podminka1");
$typ_dat=PreciZFormulare("typ_dat","Nevyplnili jste typ dat (teplota, srážky, ...)");
$misto=PreciZFormulare("misto","Nevyplnili jste misto");
$koeficient=PreciZFormulare("koeficient","Nevyplnili jste koeficient");
$default_value=PreciZFormulare("defaultvalue","Nevyplnili jste default");

if ($typ_podminky=="pred_zs")
    $podminka2=PreciZFormulare("podminka2","Nevyplnili jste podminka2");
else
    $podminka2="";

//-----
```

```

//----- pokud je nove url - vloz jej do tabulky meteo_url ----
//-----

if($url_id == "novy")
{
    // nejdrive najdu nove id_url
    $SqlPrikaz="SELECT max(url_id) as max_id FROM meteo_url";
    $Result=mysql_query($SqlPrikaz);
    $Data = mysql_fetch_array($Result,MYSQL_ASSOC);
    $url_id = $Data["max_id"]+1;
    // a pak vlozim data do tabulky meteo_url
    $SqlPrikaz="INSERT into meteo_url (url_id,url,typ_url,zpozdeni) VALUES ($url_id
,'$url','$typ_url','$zpozdeni)";
    mysql_query($SqlPrikaz);
}

//-----
//----- vloz data do tabulky meteo_zdroje -----
//-----

// nejdrive najdu nove id_zdroje

$SqlPrikaz="SELECT max(id_zdroje) as max_id FROM meteo_zdroje";
$Result=mysql_query($SqlPrikaz);

echo(htmlspecialchars($SqlPrikaz)."<br>");

$Data = mysql_fetch_array($Result,MYSQL_ASSOC);

if($Data)
    $id_zdroje = $Data["max_id"]+1;
else
    $id_zdroje = 0;

// vlozim do tabulky

$SqlPrikaz="INSERT into meteo_zdroje
(id_zdroje,url_id,typ_podminky,podminka1,podminka2,typ_dat,misto,koeficient,defaultvalue) VALUES ".

"($id_zdroje,$url_id,'$typ_podminky','$podminka1','$podminka2','$typ_dat','$misto','$koeficient','$default_val
ue)";
$Result=mysql_query($SqlPrikaz);
var_dump($Result);
echo("<br>".mysql_error()."<br>");

?>
<HTML>

<HEAD>

<TITLE>Vloženo</TITLE>

</HEAD>

<BODY BGCOLOR="WHITE">
    Vloženo - <a href="add1.php">zpět</a><br />
    <?php echo(htmlspecialchars($SqlPrikaz)."<br>"); ?>

</BODY>
</HTML>

```

4.6.5 Soubor del1.php

Vypíše z databáze do HTML seznamu všechny stahované stránky, každou je možné smazat / editovat (delete.php / edit_url1.php). Lze také samostatně mazat / editovat (delete.php / edit_zdroj1.php).dotazy na data do těchto stránek.

[vypis dat](#) [smazat/opravit zdroj/uri](#) [přidat nový zdroj/uri](#)

Mazání a opravy zdrojů a url

- [http://aplikace.mvcr.cz/doprava/nehodyst/index.xml](#) [smaz celou url](#) [oprav url](#)
 - o typ podmínky: xpath, typ dat: **Počet dopravních nehod**, místo: **Praha u Centrotexu**, koeficient: 1, default: podmínka 1: //oblast[contains(@nazev podmínka 2
 - [smazat jeden zdroj](#) [upravit zdroj](#)
- [http://weather.man.com/local.aspx?&wealocations=wc%3a6410&unit=C](#) [smaz celou url](#) [oprav url](#)
 - o typ podmínky: pred_zs, typ dat: **Nejnižší teplota**, místo: **České Budějovice**, koeficient: 1, default: podmínka 1: <td></tr><tr><th>Lo</th><td></td></tr><tr><td></td></tr></table>
 - [smazat jeden zdroj](#) [upravit zdroj](#)
 - o typ podmínky: rsg, typ dat: **Relativní vlhkost den**, místo: **České Budějovice**, koeficient: 1, default: podmínka 1: Day: [^<]*humidity ([0-9]+]
 - [smazat jeden zdroj](#) [upravit zdroj](#)
 - o typ podmínky: pred_zs, typ dat: **Nejvyšší teplota**, místo: **České Budějovice**, koeficient: 1, default: podmínka 1: <table class="hlo"><tr><th>Hi</th><td></td></tr><tr><td></td></tr></table>
 - [smazat jeden zdroj](#) [upravit zdroj](#)
- [http://www.chmi.cz/meteo/opss/pocasi.php?ukazatel=uhmrsazek&pozadi=rapareg&gra=ano](#) [smaz celou url](#) [oprav url](#)
 - o typ podmínky: pred_zs, typ dat: **Srážky**, místo: **Praha**, koeficient: 1, default: 0
 - podmínka 1: uzauá
 - podmínka 2: kbahj
 - [smazat jeden zdroj](#) [upravit zdroj](#)
 - o typ podmínky: pred_zs, typ dat: **Srážky**, místo: **Ceske Budejovice**, koeficient: 1, default: 0
 - podmínka 1: jovice" STYLE="position: absolute; left:266; top:342"
 - podmínka 2: </DIV>
 - [smazat jeden zdroj](#) [upravit zdroj](#)

```
<?php
```

```
include ("./config.php");
```

```
//-----  
//----- připojení k databázi -----  
//-----
```

```
if(!mysql_connect(SQL_HOST, SQL_USERNAME, SQL_PASSWORD))  
die("Nelze se připojit k MySQL: " . mysql_error()  
."\\nMozne priciny:"  
."\\n - spatne nastaveny config.php (jmeno databaze, uzivatel, heslo)"  
."\\n - MySQL na serveru ".SQL_HOST." nebezi"  
."\\n - nezalozeny uzivatel ".SQL_USERNAME  
."\\n - spatne heslo uzivatele"  
."\\n - ...");
```

```
if(!mysql_select_db(SQL_DBNAME))  
die("Nelze se přepnout do databáze ".SQL_DB." : " . mysql_error());
```

```
//mysql_query("SET NAMES 'cp1250';");
```

```
//----- uložíme si seznam url do pole
```

```
$SeznamUrl=array();  
$Result=mysql_query('SELECT * FROM Meteo_url ORDER BY url');  
while($Data = mysql_fetch_array($Result,MYSQL_ASSOC))  
$SeznamUrl["=".$Data["url_id"]] = $Data["url"];  
mysql_free_result($Result);
```

```

?>
<HTML>

<HEAD>

<TITLE>Mazání a opravy</TITLE>

</HEAD>

<BODY BGCOLOR="WHITE">
  <div align="right"><a href="konkretni_dotaz.php">výpis dat</a> &nbsp; &nbsp; <a
href="del1.php">smazat/opravit zdroj/url</a> &nbsp; &nbsp; <a href="add1.php">přidat nový
zdroj/url</a></div>

  Mazání a opravy zdrojů a url<br>

  <ul>
  <?php
  // pro každé url
  foreach($SeznamUrl as $Id => $Hodnota)
  {
    echo("<li>".htmlspecialchars($Hodnota)." &nbsp; <a href='\"delete.php?typ=url&amp;id\".$Id.\"'>smaz
celou url</a>");
    echo(" &nbsp; &nbsp; <a href='\"edit_url1.php?id\".$Id.\"'>oprav url</a>");
    // pro kazdy zdroj url
    echo("<ul>");
    $Result=mysql_query('SELECT * FROM meteo_zdroje WHERE url_id'.$Id);
    while($Data = mysql_fetch_array($Result,MYSQL_ASSOC))
    {
      echo("<li>");
      echo("typ podmínky: ");
      echo(htmlspecialchars($Data["typ_podminky"]));
      echo(", typ dat: <b>");
      echo(htmlspecialchars($Data["typ_dat"]));
      echo("</b>, místo: <b>");
      echo(htmlspecialchars($Data["misto"]));
      echo("</b>, koeficient: ");
      echo(htmlspecialchars($Data["koeficient"]));
      echo(", default: ");
      echo(htmlspecialchars($Data["defaultvalue"]));
      echo("<br>podmínka 1: ");
      echo(htmlspecialchars($Data["podminka1"]));
      echo("<br>podmínka 2: ");
      echo(htmlspecialchars($Data["podminka2"]));
      echo("<br><a href='\"delete.php?typ=zdroj&amp;id\".$Data[\"id_zdroje\"].\"'>smazat jeden
zdroj</a>");
      echo(" &nbsp; &nbsp; <a href='\"edit_zdroj1.php?id\".$Data[\"id_zdroje\"].\"'>upravit zdroj</a>");
      echo("</li>");
    }
    mysql_free_result($Result);
    echo("</ul>");
    echo("</li>");
  }
  ?>

</ul>

</BODY>
</HTML>

```

4.6.6 Soubor delete.php

Skript na základě url_id celé stránky, nebo jen id_zdroje jednoho dotazu, odeslaného ze souboru del1.php vymaže odpovídající záznam(y) z databáze.

```
<?php                                     //SOUBOR PRO SMAZÁNÍ STRÁNKY,NEBO DOTAZU DO
STRÁNKY

    Header("Expires: ".GMDDate("D, d M Y H:i:s")." GMT");    //omezeni k pouziti pro tlacitko zpet v
prohlizeci

include ("./config.php");

function OsetriChybu($TextChyby,$Skonci=false)
{
    $F=fopen("./chyby.txt","ab"); // otevri soubor pro zapis na konec souboru
    fwrite($F,"Chyba: ".$TextChyby."\r\n");    // zapis chyby
    fclose($F);    // zavri soubor

    echo($TextChyby."\r\n");
    if($Skonci) exit();
}

//-----
//----- připojení k databázi -----
//-----

if(!mysql_connect(SQL_HOST, SQL_USERNAME, SQL_PASSWORD))
    OsetriChybu("Nelze se připojit k MySQL: " . mysql_error()
        ."\nMozne priciny:"
        ."\n - spatne nastaveny config.php (jmeno databaze, uzivatel, heslo)"
        ."\n - MySQL na serveru ".SQL_HOST." nebezi"
        ."\n - nezalozeny uzivatel ".SQL_USERNAME
        ."\n - spatne heslo uzivatele"
        ."\n - ...",true);

if(!mysql_select_db(SQL_DBNAME))
    OsetriChybu("Nelze se přepnout do databáze ".SQL_DB." : " . mysql_error(),true);

//-----
//----- kontrola dat formuláře ----
//-----

function PreciZFormulare($Nazev,$TextChyby)
{
    if(!array_key_exists($Nazev,$_REQUEST)) //hledaný prvek,zdroj
        die($TextChyby);
    return($_REQUEST[$Nazev]);
}

$typ=PreciZFormulare("typ","špatný formulář");
$id=PreciZFormulare("id","špatný formulář");

//-----
//----- pokud je typ "url" - promaž meteo_url i meteo_zdroje -
//-----

if($typ == "url")
{
    $SqlPrikaz="DELETE FROM meteo_url WHERE url_id=".$id;
    echo(htmlspecialchars($SqlPrikaz)."<br>");
    mysql_query($SqlPrikaz);
    $SqlPrikaz="DELETE FROM meteo_zdroje WHERE url_id=".$id;
    mysql_query($SqlPrikaz);
}
```

```

}

//-----
//----- pokud je typ "zdroj" - promaž jen meteo_zdroje -
//-----

if($typ == "zdroj")
{
    $SqlPrikaz="DELETE FROM meteo_zdroje WHERE id_zdroje=".$id;
    mysql_query($SqlPrikaz);
}

?>
<HTML>

<HEAD>

<TITLE>Smazáno</TITLE>

</HEAD>

<BODY BGCOLOR="WHITE">
    Smazáno - <a href="del1.php">zpět</a><br />
    <?php echo($typ."<br>"); echo($id."<br>");

    echo(htmlspecialchars($SqlPrikaz)."<br>"); ?>

</BODY>
</HTML>

```

4.6.7 Soubor edit_url1.php

Skript na základě url_id celé stránky, odeslaného ze souboru del1.php zobrazí vybranou adresu url pro editaci.

Oprava URL

URL:

Zpoždění: hodin Typ url: ☐ HTML ☒ XML ☐ RSS

[zpět](#)

```
<?php
```

```
include ("./config.php");
```

```
//-----  
//----- připojení k databázi -----  
//-----
```

```
if(!mysql_connect(SQL_HOST, SQL_USERNAME, SQL_PASSWORD))  
    die("Nelze se připojit k MySQL: " . mysql_error()  
        . "\nMozne priciny:"  
        . "\n - spatne nastaveny config.php (jmeno databaze, uzivatel, heslo)"  
        . "\n - MySQL na serveru ".SQL_HOST." nebezi"  
        . "\n - nezalozeny uzivatel ".SQL_USERNAME  
        . "\n - spatne heslo uzivatele"  
        . "\n - ...");
```

```
if(!mysql_select_db(SQL_DBNAME))  
    die("Nelze se přepnout do databáze ".SQL_DB." : " . mysql_error());
```

```
//mysql_query("SET NAMES 'cp1250';");
```

```
//----- přečteme id_url
```

```
if(!array_key_exists("id",$_REQUEST)) die("Neznámý řádek tabulky");  
$Url_id=$_REQUEST["id"];
```

```
//----- načteme data o daném id
```

```
$SeznamUrl=array();  
$Result=mysql_query('SELECT * FROM Meteo_url WHERE url_id='.$Url_id);  
$Data = mysql_fetch_array($Result,MYSQL_ASSOC);  
if(!is_array($Data)) die("Neznámý řádek tabulky");  
$Url=$Data['url'];  
$TypUrl=$Data['typ_url'];  
$Zpozdeni=$Data['zpozdeni'];  
mysql_free_result($Result);
```

```

?>
<HTML>

<HEAD>

<TITLE>Editace url</TITLE>

</HEAD>

<BODY BGCOLOR="WHITE">

Oprava URL<br>
<form action="edit_url2.php">
<input type="hidden" name="url_id" value="<?php echo($Url_id);?>">
URL: &nbsp; &nbsp; &nbsp; <input type="text" name="url" value="<?php
echo(htmlspecialchars($Url));?>" size="90"><br />
Zpoždění: <input type="text" name="zpozdeni" value="<?php echo($Zpozdeni);?>" size="5"> hodin
&nbsp; &nbsp;
Typ url:
<input type="radio" name="typ_url" value="html" <?php if($TypUrl=='html')
echo('checked="checked"');?>> HTML &nbsp; &nbsp;
<input type="radio" name="typ_url" value="xml" <?php if($TypUrl=='xml')
echo('checked="checked"');?>> XML &nbsp; &nbsp;
<input type="radio" name="typ_url" value="rss" <?php if($TypUrl=='rss') echo('checked="checked"');?>>
RSS<br />
&nbsp; &nbsp; &nbsp; &nbsp; &nbsp; &nbsp; &nbsp; &nbsp; &nbsp; &nbsp; &nbsp; &nbsp; &nbsp;
<input type="submit" value="Odeslat">
</form>
<a href="del1.php">zpět</a>

</BODY>
</HTML>

```

4.6.8 Soubor edit_url2.php

Potvrzením tlačítka Odeslat na základě url_id celé stránky, odeslaného ze souboru edit_url1.php uloží do databáze odpovídající změny. Ponechat stávající stav, ještě před odesláním změn do MySQL lze odkazem zpět. Funkce PreciZFormulare, kontroluje, zda byly vyplněna konkrétní položka(pole formuláře) a případně zabezpečí aplikaci proti podsunutí např. příkazu sql do polí formuláře(např delete databáze).

```
<?php

include ("./config.php");

//-----
//----- připojení k databázi -----
//-----

if(!mysql_connect(SQL_HOST, SQL_USERNAME, SQL_PASSWORD))
    die("Nelze se připojit k MySQL: " . mysql_error()
        . "\nMozne priciny:"
        . "\n - spatne nastaveny config.php (jmeno databaze, uzivatel, heslo)"
        . "\n - MySQL na serveru ".SQL_HOST." nebezi"
        . "\n - nezalozeny uzivatel ".SQL_USERNAME
        . "\n - spatne heslo uzivatele"
        . "\n - ...");

if(!mysql_select_db(SQL_DBNAME))
    die("Nelze se přepnout do databáze ".SQL_DB." : " . mysql_error());

//mysql_query("SET NAMES 'cp1250';");

//-----
//----- kontrola dat formuláře ----
//-----

function PreciZFormulare($Nazev,$TextChyby,$MaBytCislo)
{
    if(!array_key_exists($Nazev,$_REQUEST) || $_REQUEST[$Nazev]=="") //hledaný prvek,zdroj
        die($TextChyby);
    if($MaBytCislo && $_REQUEST[$Nazev] != "NULL" && !ereg("[0-9]+$",$_REQUEST[$Nazev]))
        die($TextChyby);
    return(str_replace("'",`"`,str_replace("`",`"`,$_REQUEST[$Nazev]))); // přepíšu " na ` a ` na `
}

$url_id=PreciZFormulare("url_id","Špatný data ve formuláři",true);
$url=PreciZFormulare("url","Nevyplnili jste url",false);
$typUrl=PreciZFormulare("typ_url","Nevyplnili jste typ URL",false);
$zpozdeni=PreciZFormulare("zpozdeni","Nevyplnili jste zpoždění",true);

//----- načteme data o daném id

$sqlPrikaz="REPLACE meteo_url SET url_id=".$url_id
            .", url='".$url.'"
            .", typ_url='".$typUrl.'"
            .", zpozdeni='".$zpozdeni';

mysql_query($sqlPrikaz);

?>
<HTML>

<HEAD>

<TITLE>Změněno</TITLE>

</HEAD>
```

```
<BODY BGCOLOR="WHITE">
  Změněno - <a href="del1.php">zpět</a><br />
  <?php echo(htmlspecialchars($SqlPrikaz)."<br>"); ?>
```

```
</BODY>
</HTML>
```

4.6.9 Soubor edit_zdroj1.php

Skript na základě id_zdroje jednoho dotazu, odeslaného ze souboru del1.php zobrazí vybraný dotazy pro editaci.

Oprava zdroje

Typ podmínky:

☒ text před/za
 ☐ regulární výraz
 ☐ xpath

Podmínka 1:

</td></tr><tr><th>Lo:</th><td>

Podmínka 2:

°</td></tr></table>

Typ dat:

Nejnižší teplota

Místo:

České Budějovice

Koeficient:

1

Default:

NULL

Odeslat

[zpět](#)

```
<?php

include ("./config.php");

//-----
//----- připojení k databázi -----
//-----

if(!mysql_connect(SQL_HOST, SQL_USERNAME, SQL_PASSWORD))
    die("Nelze se připojit k MySQL: " . mysql_error()
        . "\nMozne priciny:"
        . "\n - spatne nastaveny config.php (jmeno databaze, uzivatel, heslo)"
        . "\n - MySQL na serveru ".SQL_HOST." nebezi"
        . "\n - nezalozeny uzivatel ".SQL_USERNAME
        . "\n - spatne heslo uzivatele"
        . "\n - ...");

if(!mysql_select_db(SQL_DBNAME))
    die("Nelze se přepnout do databáze ".SQL_DB." : " . mysql_error());

//mysql_query("SET NAMES 'cp1250';");

//-----
//----- přečteme id_zdroje z odkazu -
//-----
if(!array_key_exists("id",$_REQUEST)) die("Neznámý řádek tabulky");
```

```
$Id_zdroje=$_REQUEST["id"];

//-----
//----- načteme z databáze data o daném id
//-----

$SeznamUrl=array();
$result=mysql_query('SELECT * FROM Meteo_zdroje WHERE id_zdroje='.$Id_zdroje);
$data = mysql_fetch_array($result,MYSQL_ASSOC);
if(!is_array($data)) die("Neznámý řádek tabulky");
$url_id=$data['url_id'];
$type_podminky=$data['typ_podminky'];
$podminka1=$data['podminka1'];
$podminka2=$data['podminka2'];
$type_dat=$data['typ_dat'];
$misto=$data['misto'];
$koefficient=$data['koefficient'];
$defaultvalue=$data['defaultvalue'];
if(is_null($defaultvalue))
    $defaultvalue = "NULL";
mysql_free_result($result);

//-----
//----- zobrazíme formulář -----
//-----
?>
<HTML>

<HEAD>

<TITLE>Editace zdroje</TITLE>

</HEAD>

<BODY BGCOLOR="WHITE">

Oprava zdroje<br>
<form action="edit_zdroy2.php">
<input type="hidden" name="id_zdroje" value="<?php echo($Id_zdroje);?>">
<input type="hidden" name="url_id" value="<?php echo($Url_id);?>">
Typ podmínky:
    <input type="radio" name="typ_podminky" value="pred_za" <?php if($Type_podminky=='pred_za')
echo('checked="checked");?>> text před/za &nbsp; &nbsp; &nbsp;;
    <input type="radio" name="typ_podminky" value="reg" <?php if($Type_podminky=='reg')
echo('checked="checked");?>> regulární výraz &nbsp; &nbsp; &nbsp;;
    <input type="radio" name="typ_podminky" value="xpath" <?php if($Type_podminky=='xpath')
echo('checked="checked");?>> xpath<br />
Podmínka 1: &nbsp; &nbsp; &nbsp;; <input type="text" name="podminka1" value="<?php
echo(htmlspecialchars($Podminka1));?>" size="90"><br />
Podmínka 2: &nbsp; &nbsp; &nbsp;; <input type="text" name="podminka2" value="<?php
echo(htmlspecialchars($Podminka2));?>" size="90"><br />
Typ dat: &nbsp; &nbsp; &nbsp;; <input type="text" name="typ_dat" value="<?php
echo(htmlspecialchars($Type_dat));?>" size="90"><br />
Místo: &nbsp; &nbsp; &nbsp;; <input type="text" name="misto" value="<?php
echo(htmlspecialchars($Misto));?>" size="90"><br />
Koeficient: &nbsp; &nbsp; &nbsp;; <input type="text" name="koefficient" value="<?php
echo($Koefficient);?>" size="10"> &nbsp; &nbsp;;
Default: &nbsp; &nbsp; &nbsp;; <input type="text" name="defaultvalue" value="<?php
echo($Defaultvalue);?>" size="10"><br />
&nbsp; &nbsp; &nbsp;; &nbsp; &nbsp; &nbsp;; &nbsp; &nbsp; &nbsp;; &nbsp; &nbsp; &nbsp;;
<input type="submit" value="Odeslat">
</form>
<a href="del1.php">zpět</a>

</BODY>
</HTML>
```

4.6.10 Soubor edit_zdroj2.php

Potvrzením tlačítka Odeslat na základě id_zdroje jednoho dotazu, odeslaného ze souboru edit_zdroj1.php uloží do databáze odpovídající změny. Ponechat stávající stav, ještě před odesláním změn do MySql lze odkazem zpět. Funkce PrectiZFormulare, kontroluje, zda byly vyplněna konkrétní položka(pole formuláře) a případně zabezpečí aplikaci proti podsunutí nepovolených příkazů do polí formuláře, které by mohly poškodit aplikaci.

```
<?php

include ("./config.php");

//-----
//----- připojení k databázi -----
//-----

if(!mysql_connect(SQL_HOST, SQL_USERNAME, SQL_PASSWORD))
    die("Nelze se připojit k MySQL: " . mysql_error()
        . "\n\n Mozne priciny:"
        . "\n - spatne nastaveny config.php (jmeno databaze, uzivatel, heslo)"
        . "\n - MySQL na serveru ".SQL_HOST." nebezi"
        . "\n - nezalozeny uzivatel ".SQL_USERNAME
        . "\n - spatne heslo uzivatele"
        . "\n - ...");

if(!mysql_select_db(SQL_DBNAME))
    die("Nelze se přepnout do databáze ".SQL_DB." : " . mysql_error());

//-----
//----- kontrola dat formuláře ----
//-----

function PrectiZFormulare($Nazev,$TextChyby,$MaBytCislo)
{
    if(!array_key_exists($Nazev,$_REQUEST) || $_REQUEST[$Nazev]== "") //hledaný prvek,zdroj
        die($TextChyby);
    if($MaBytCislo && $_REQUEST[$Nazev] != "NULL" && !ereg("[0-9]+$",$_REQUEST[$Nazev]))
        die($TextChyby);
    return(str_replace("\'", "\\\'", str_replace("\\\\", "\\\\", $_REQUEST[$Nazev]])); // přepíšu " na \' a \' na \
}

$id_zdroje=PrectiZFormulare("id_zdroje","Špatný data ve formuláři",true);
$url_id=PrectiZFormulare("url_id","Špatný data ve formuláři",true);
$typ_podminky=PrectiZFormulare("typ_podminky","Nevyplnili jste typ podmínky",false);
$podminka1=PrectiZFormulare('podminka1',"Nevyplnili jste podmínku 1",false);
$podminka2=PrectiZFormulare('podminka2',"Nevyplnili jste podmínku 2",false);
$typ_dat=PrectiZFormulare('typ_dat',"Nevyplnili jste typ dat",false);
$misto=PrectiZFormulare('misto',"Nevyplnili jste místo",false);
$koeficient=PrectiZFormulare('koeficient',"Nevyplnili jste koeficient",true);
$defaultvalue=PrectiZFormulare('defaultvalue',"Nevyplnili jste defaultní hodnotu",true);

//-----
//----- uložíme data do databáze ----
//-----

$sqlPrikaz="REPLACE meteo_zdroje SET id_zdroje=".$id_zdroje
        .", url_id=".$url_id
        .", typ_podminky='".$typ_podminky.'"
        .", podminka1='".$podminka1.'"
        .", podminka2='".$podminka2.'"
        .", typ_dat='".$typ_dat.'"
        .", misto='".$misto.'"
        .", koeficient=".$koeficient
```

```

        .", defaultvalue=".$Defaultvalue;

mysql_query($SqlPrikaz);

//-----
//----- vypíšeme hotovo -----
//-----

?>
<HTML>

<HEAD>

<TITLE>Změněno</TITLE>

</HEAD>

<BODY BGCOLOR="WHITE">
    Změněno - <a href="del1.php">zpět</a><br />
    <?php echo(htmlspecialchars($SqlPrikaz)."<br>"); ?>

</BODY>
</HTML>

```

4.6.11 Soubor konkretni_dotaz.php

Je formulář společně s kódem PHP, spojí ve kterém si uživatel sestaví podmínku pro výběr dat a vybere formátování pro výstup. Některá vstupní data do formuláře jsou z databáze. Je to předvyplněný časový rozsah dat pro zobrazení a nabídky pro výběr dat. Výsledek posílá ke zpracování skriptu zobraz_data.php.

[výpis dat](#) [smazat/opravit zdroj/uri](#) [přidat nový zdroj/uri](#)

Výběr dat

Typ dat:

Nejnižší teplota

Místo:

vše

Od:

2008-10-24

Výsledek:

Webová stránka

Hodnoty:

Průměr

Třídění:

Datum

Potvrdit

Zpět

Vybraná veličina, nebo popis stahovaných dat

Lokalizace

Do: 2008-11-14

Výstup do souboru:html, xml, csv

K provedení dotazu jsou povinné vyplnit všechny dotazy !!!

```
<?php
```

```
include ("./config.php");
```

```
function OsetriChybu($TextChyby,$Skonci=false)
{
    $F=fopen("./chyby.txt","ab"); // otevri soubor pro zapis na konec souboru
    fwrite($F,"Chyba: ".$TextChyby."\r\n"); // zapis chyby
    fclose($F); // zavri soubor

    echo($TextChyby."\r\n");
    if($Skonci) exit();
}
```

```
//-----
//----- připojení k databázi -----
//-----
```

```
if(!mysql_connect(SQL_HOST, SQL_USERNAME, SQL_PASSWORD))
    OsetriChybu("Nelze se připojit k MySQL: " . mysql_error()
        ."\r\nMozne priciny:"
        ."\r\n - spatne nastaveny config.php (jmeno databaze, uzivatel, heslo)"
        ."\r\n - MySQL na serveru ".SQL_HOST." nebezi"
        ."\r\n - nezalozeny uzivatel ".SQL_USERNAME
        ."\r\n - spatne heslo uzivatele"
        ."\r\n - ...",true);

if(!mysql_select_db(SQL_DBNAME))
    OsetriChybu("Nelze se přepnout do databáze ".SQL_DB." : " . mysql_error(),true);

//mysql_query("SET NAMES 'cp1250';");
```

```

$Result=mysql_query('SELECT min(Datum) as MinDatum FROM Meteo_data');
$Data = mysql_fetch_array($Result,MYSQL_ASSOC);
$rozsah_od=$Data["MinDatum"];
mysql_free_result($Result);

$Result=mysql_query('SELECT max(Datum) as MaxDatum FROM Meteo_data');
$Data = mysql_fetch_array($Result,MYSQL_ASSOC);
$rozsah_do=$Data["MaxDatum"];
mysql_free_result($Result);

?>

<HTML>

<HEAD>

<TITLE>Konkrétní dotaz</TITLE></font>
<meta http-equiv="Content-Type" content="text/html; charset=windows-1250" />

</HEAD>

<BODY BGCOLOR="WHITE">

    <div align="right"><a href="konkretni_dotaz.php">výpis dat</a> &nbsp; &nbsp; <a
href="del1.php">smazat/opravit zdroj/url</a> &nbsp; &nbsp; <a href="add1.php">přidat nový
zdroj/url</a></div>

    <font face="Arial, Helvetica" size="5">
    <p style="text-align: center"><B>Výběr dat</B></p>
    </font>

    <FORM ACTION="zobraz_data.php" METHOD=POST NAME="formular">    <!--zacatek formulare, k
obsluze se pouzije soubor XXX.php-->

        <TABLE>
        <TR>
        <TD>
        <font color="black">        Typ dat:</font>

            </TD>
            <TD>

                <SELECT NAME="TYP_DAT">
                <?php
                $Result=mysql_query('SELECT typ_dat FROM Meteo_data GROUP BY typ_dat ORDER BY
typ_dat');
                while($Data = mysql_fetch_array($Result,MYSQL_ASSOC))
                echo("<option value='\"".$Data["typ_dat"]."\">\"".$Data["typ_dat"]."\"</option>\r\n");
                mysql_free_result($Result);
                ?>

                </SELECT>
            </TD>
            <TD> </TD>
            <TD>případný komentář
            </TD>

        </TR>
        <TR>
        <TD>
            Místo:
        </TD>
        <TD>
            <SELECT NAME="MISTO">

```

```

        <option value="vsechno">vše</option>
        <?php
        $Result=mysql_query('SELECT misto FROM Meteo_data GROUP BY misto ORDER BY misto');
        while($Data = mysql_fetch_array($Result,MYSQL_ASSOC))
            echo("<option value='".$Data["misto"]."'">".$Data["misto"]."");
        mysql_free_result($Result);
        ?>
    </SELECT>
</TD>
<TD> </TD>
<TD>případný komentář
</TD>

</TR>
    <TR>
<TD>
    Od :
</TD>
<TD>
    <INPUT TYPE=TEXT NAME="od" VALUE="<?php echo $rozsah_od?>" >
</TD>
<TD>
    Do:
</TD>
<TD>
    <INPUT TYPE=TEXT NAME="do" VALUE="<?php echo $rozsah_do?>" >

</TD>

</TR>
<TR>
    <TD>

    Výsledek :

</TD>
    <TD>

    <SELECT NAME="VYSLEDEK">
    <OPTION VALUE="WEB" > Webová stránka
    <OPTION VALUE="XML" > XML
    <OPTION VALUE="csv" > soubor csv

    </SELECT>
</TD>
    <TD> </TD>
    <TD>případný komentář
    </TD>

    </TR>

<TR>
    </TR>

<TR>

    <TD WIDTH=200>

    Hodnoty :

    <TD WIDTH=500>

    <SELECT NAME="HODNOTY">
    <OPTION VALUE="AVG" > Průměr
    <OPTION VALUE="SUM" > Součet
    <OPTION VALUE="DAY" > Jednotlivé dny

```

```

</SELECT>
<TD>

    </TR>

<TR>

    <TD WIDTH=200>

        Třídění :
        <TD WIDTH=500>

            <SELECT NAME="TRIDENI">
            <OPTION VALUE="datum" > Datum
            <OPTION VALUE="misto" > Místo
            <OPTION VALUE="hodnota" > Hodnota

            </SELECT>
        </TR>

        <TR>
    </TR>
        <TR>
    </TR>
</TABLE>

<TABLE>

<TR>

<TD WIDTH=600 ALIGN="CENTER">

    <INPUT TYPE=SUBMIT VALUE="Potvrdit" NAME="Confirm">

    <INPUT TYPE=BUTTON VALUE="Zpet" onClick=Zpet()>
</TR>
</TABLE>

<INPUT TYPE=HIDDEN NAME="admin" VALUE="0");

</FORM>

<P>K provedení dotazu jsou povinné vyplnit všechny dotazy !!!
</BODY>

</HTML>          <!--Ukončení stránky pro běžné uživatele-->

<SCRIPT Language="javascript"> //funkce v javascriptu pro tlačítko zpet

<!--
function Zpet()

{

    history.back();

}

//-->

</SCRIPT>

```

4.6.12 Soubor zobraz_csv.php

Skript vytvoří načtením dat z asociativního pole výsledků výstupní textový soubor csv. Přípona souboru je txt, oddělovacím znakem je středník.

```
Srazky;2008-10-30;Ceske Budejovice;3
Srazky;2008-11-13;Ceske Budejovice;6
Srazky;2008-11-14;Ceske Budejovice;6
```

```
header("Content-Type: text/plain; charset=windows-1250");
//header("Content-Type: application/force-download");
//header("Content-Disposition: attachment; filename=\"data.txt\"");
```

```
while($Data = mysql_fetch_array($Result,MYSQL_ASSOC))
{
    echo(implode(";", $Data));
    echo("\r\n");
}
```

4.6.13 Soubor zobraz_web.php

Obsahuje html šablonu, se skriptem. Načtením dat z asociativního pole výsledků vytvoří výstupní soubor ve formátu html.

Vybraná data			
Typ dat	Místo	Datum	Hodnota
Srazky	Ceske Budejovice	2008-10-30	3
Srazky	Ceske Budejovice	2008-11-13	6
Srazky	Ceske Budejovice	2008-11-14	6

```
<HEAD>

<TITLE>Konkrétní dotaz</TITLE>
<meta http-equiv="Content-Type" content="text/html; charset=windows-1250" />

</HEAD>

<BODY BGCOLOR="WHITE">

<table border="1">
<?php

while($Data = mysql_fetch_array($Result,MYSQL_ASSOC))
{
    echo("<tr>");
    echo("<td>".$Data["typ_dat"]."</td>");
    echo("<td>".$Data["misto"]."</td>");
    if($TypHodnoty=="DAY")
        echo("<td>".$Data["datum"]."</td>");
    echo("<td>".$Data["moje_hodnota"]."</td>");
    echo("</tr>\r\n");
}

?>
</table>

</BODY>
```

4.6.14 Soubor zobraz_xml.php

Obsahuje šablonu xml, se skriptem PHP. Načtením dat z asociativního pole výsledků vytvoří výstupní soubor ve formátu xml.

```
<?xml version="1.0" encoding="windows-1250" ?>
- <data typ="Srazky">
  <sql_dotaz>SELECT typ_dat, datum, misto, hodnota as moje_hodnota FROM Ymeteo_data WHERE typ_dat='Srazky' AND datum>='2008-10-24' AND
    datum<='2008-11-14' ORDER BY datum</sql_dotaz>
  <hodnota misto="Česke Budejovice" datum="2008-10-30">3</hodnota>
  <hodnota misto="Česke Budejovice" datum="2008-11-13">6</hodnota>
  <hodnota misto="Česke Budejovice" datum="2008-11-14">6</hodnota>
</data>
```

```
<?xml version="1.0" encoding="windows-1250" ?>
<data typ="<?php echo $TypDat?>">
<sql_dotaz>
  <?php echo(htmlspecialchars($SqlPrikaz));?>
</sql_dotaz>
<?php

while($Data = mysql_fetch_array($Result,MYSQL_ASSOC))
{
  echo("<hodnota misto='".$Data['misto']."'");
  if($TypHodnoty=="DAY")
    echo(" datum='".$Data['datum']."'");
  echo(">");
  echo($Data['moje_hodnota']);
  echo("</hodnota>\r\n");
}

?>
</data>
```

4.6.15 Soubor zobraz_data.php

Skript zobraz_data sestaví na základě kritérií zadaných do formuláře konkrétní dotaz. Ten vybere z tabulky meteo_data, která vyhovují dotazu. Vytvoří výstupní soubor ve formátu podle požadavku uživatele, který zadal do formuláře.

```
include ("./config.php");

function OsetriChybu($TextChyby,$Skonci=false)
{
    $F=fopen("./chyby.txt","ab"); // otevri soubor pro zapis na konec souboru
    fwrite($F,"Chyba: ".$TextChyby."\r\n"); // zapis chyby
    fclose($F); // zavri soubor

    echo($TextChyby."\r\n");
    if($Skonci) exit();
}

//-----
//----- připojení k databázi -----
//-----

if(!mysql_connect(SQL_HOST, SQL_USERNAME, SQL_PASSWORD))
    OsetriChybu("Nelze se připojit k MySQL: " . mysql_error()
        ."\r\nMozne priciny:"
        ."\r\n - spatne nastaveny config.php (jmeno databaze, uzivatel, heslo)"
        ."\r\n - MySQL na serveru ".SQL_HOST." nebezi"
        ."\r\n - nezalozeny uzivatel ".SQL_USERNAME
        ."\r\n - spatne heslo uzivatele"
        ."\r\n - ...",true);

if(!mysql_select_db(SQL_DBNAME))
    OsetriChybu("Nelze se přepnout do databáze ".SQL_DB." : " . mysql_error(),true);

mysql_query("SET NAMES 'cp1250';");

//-----
//----- zpracování dat formuláře ----
//-----

function PreciZFormulare($Nazev,$TextChyby)
{
    if(!array_key_exists($Nazev,$_REQUEST))
        die($TextChyby);
    return($_REQUEST[$Nazev]);
}

$TypDat=PreciZFormulare("TYP_DAT","Nevyplnili jste typ dat");
$Misto=PreciZFormulare("MISTO","Nevyplnili jste misto");
$DatumOd=PreciZFormulare("od","Nevyplnili jste datum od");
$DatumDo=PreciZFormulare("do","Nevyplnili jste datum do");
$TypVysledku=PreciZFormulare("VYSLEDEK","Nevyplnili jste typ vysledku");
$TypHodnoty=PreciZFormulare("HODNOTY","Nevyplnili jste typ hodnot");
$Trideni=PreciZFormulare("TRIDENI","Nevyplnili jste zpusob trideni");

//-----
//----- sestavíme sql dotaz -----
//-----

if($TypHodnoty=="AVG")
    $SqlPrikaz="SELECT typ_dat, datum, misto, avg(hodnota) as moje_hodnota";
else if($TypHodnoty=="SUM")
    $SqlPrikaz="SELECT typ_dat, datum, misto, sum(hodnota) as moje_hodnota";
else    $SqlPrikaz="SELECT typ_dat, datum, misto, hodnota as moje_hodnota";
```

```

$SqlPrikaz=$SqlPrikaz." FROM Meteo_data WHERE datum>=".".$DatumOd.""
." AND datum<=".".$DatumDo.""";
if($Misto!="vsechno")
    $SqlPrikaz=$SqlPrikaz." AND misto=".".$Misto.""";
if($TypDat!="vsechno")
    $SqlPrikaz=$SqlPrikaz." AND typ dat=".".$TypDat.""";
if($TypHodnoty!="DAY")
    $SqlPrikaz=$SqlPrikaz." GROUP BY typ_dat, misto";
$SqlPrikaz=$SqlPrikaz." ORDER BY ".$Trideni;
//-----
//----- položíme sql dotaz, zobrazíme výsledek -----
//-----

$Result=mysql_query($SqlPrikaz);

if($TypVysledku=="WEB")
    include("zobraz_web.php");
else if($TypVysledku=="XML")
    include("zobraz_xml.php");
else include("zobraz_csv.php");

mysql_free_result($Result);

```

4.6.16 Soubor help.html

Soubor se spouští odkazem ze vstupního formuláře zadávání podmínek pro stahování dat. Obsahuje návod jak vybrat typ podmínky, a jak ji zadávat. Obsahuje syntaxi a postup tvoření podmínek pro příkaz xpath a regulární výrazy.

Aplikace stahování www, nápověda

[Obecně o programu](#)
[Vstupní formulář - stahujeme jakoukoliv stránku](#)
[Pomůcka pro zadání podmínky - stežerí dat pomocí regulárního výrazu](#)
[Pomůcka pro zadání podmínky - stežerí dat XML](#)
[Formulář Výběr a zobrazení stažených dat dat](#)

Obecně
Tato aplikace vám umožní stahovat data z www stránek, podle kritérií, které zadáte do vstupního formuláře. Postup jak na to je popsán v následujících bodech.

Vstupní formulář
Ve vstupním formuláři vyplňujete položky:

WWW stránka - obsahuje www adresu, z které chcete stahovat včetně http://

Typ podmínky - výběrové pole ze tří možností.
Další postup určíte výběrem ze seznamu Typ_podminky ve formuláři v závislosti na formátu zdroje dat.

Výběr typu podmínky

- 1.reg(regulární výraz)** - regulární výraz se doplňuje pouze do pole podmínka1. Vyzít se dá pro všechny typy html, xml, rss, text
- 2.text_pred_za** - text před požadovaným údajem ze zdrojového kódu stránky se vypíše(zkopíruje) do pole podmínka1 text za požadovaným údajem ze zdrojového kódu stránky se vypíše(zkopíruje) do pole podmínka2. Využit se dá pro všechny typy html, xml, rss, text:
- 3.xpath** - výraz, který předá parametr interní funkci: PHP xpath, definované pro filtrování záznamů ve strukturně elementů dokumentu xml. Podmínka se doplní pouze do pole Podmínka1.

Položky formuláře:

Podmínka1, Podmínka2 - řetězec max. 255 znaků

Místo - popis lokace dat, např. Praha, Mars, Stodůlky, Idnes...

Typ dat - popis dat, např. Maximální teplota, Srážky, Kurz dolaru...

Koeficient - jakým číslem vynásobíme stažená data(použitelné pro převod jednotek)

Zpoždění - údaj v hodinách, který je potřeba přičíst, odečíst od data kdy data stáhneme, aby stažená hodnota odpovídala "datu měření"

Default - TRUE/FALSE - zadáním hodnoty NULL program neskončí výpisem chyby - stále - stránka bez bez dat, které vyhovují podmínkám - bez srážek(ten den nepřišlo), případně lze do pole doplnit číslo (z uvedenému datu, jestliže není údaj uvést, přiřadí 0 - nepřišlo, nebo jindy číselný údaj)

[Pomůcka pro zadání podmínky - stežerí dat pomocí regulárního výrazu](#)

Výraz podmínky zadávejte pouze do pole Podmínka1 formuláře.
Regulární výrazy:
Regulární výrazy slouží ke zpracování textových řetězců. Je to maska podle které s. určíme, zda se v daném textu li vyhovující hledaný řetězec vyskytuje.
K definování regulárních výrazů se používají zástupné znaky- Metaznaky. Pokud chceme metaznak zapsat tak, aby nebyl chápán jako metaznak, ale jako obyčejný znak (třeba plus či hvězdička), stačí před metaznakem znak backslash v regulárním výrazu zpětně lomítko \

. (tečka) - zastupuje právě jeden libovolný znak.
^ - označuje začátek řetězce

Pokud chceme v určitém místě povolit právě jednu ze skupiny znaků, uzavřeme takovou skupinu znaků do hranatých závorek ([a]). Pokud chceme naopak dlehnout, že v daném místě smí být libovolný znak s výjimkou několika určitých znaků, uzavřeme „zakázané“ znaky opět do hranatých závorek a za [ještě doplníme ^. Zápis [^xyz], tak bude znamenat, že se v daném místě má vyskytovat jeden libovolný znak kromě znaků x a y.
Kvantifikátor určuje kolikrát se má opakovat znak předcházející kvantifikátoru

Kvantifikátor	Počet opakování
?	minimálně 0krát, maximálně 1krát
*	minimálně 0krát (maximálně neomezeno)
+	minimálně 1krát (maximálně neomezeno)
*(n)	právě nkrát
*(m,n)	minimálně mkrát, maximálně nkrát
*(m,)	minimálně mkrát (maximálně neomezeno)

Předdefinované skupiny znaků

\d	číslíce 0-9
\D	jakýkoliv znak kromě číslic 0-9
\w	znaky „slova“ (ekvivalentní zápisu [a-zA-Z0-9_])
\W	jakýkoliv znak kromě znaků „slova“ (ekvivalentní zápisu [^a-zA-Z0-9_])
\s	„bílé“ znaky (mezera, tabulátor, znaky pro zalomení řádků)
\S	jakýkoliv znak kromě „bílého“ znaků

zápis \d zastupuje všechny číslice 0-9)...

Příklady	
Regulární výraz	Odpovídá...
a+	sekvence písmen a (1 a více znaků)
a*	sekvence písmen a (0 a více znaků)
o?kov	očekov či kov
te(efon)?	tel či telefon
te?f(on)ax	telefon či telefax
[0-9][1-9][0-9]	čísla 0 až 99
\d(2)	sekvence dvou číslic desítkové soustavy (00, 01, ..., 98, 99)
[0-9a-fA-F]([1-9a-fA-F][0-9a-fA-F])*	hexadecimální čísla
(19 20)\d(2)	letopočty 1900-2099
\d(2,5)	sekvence dvou až šesti číslic
[^,]*	neprázdná sekvence znaků mezi nimiž nesmí být mezera (), čárka (,) či tečka (.)
^P*	řetězec, který začíná písmenem P za nímž následuje libovolný (i nulový) počet libovolných znaků
\d+0\$	řetězec, který končí znakem 0 (nula), kterému předchází minimálně jedna číslice
a+b	ab, aab, aab atd.
a++b	a+b
x(5,10)	5-10 znaků x zapsaných za sebou

Pomůcka pro zadání podmínky - stažení dat XML

Zadání podmínky pro stažení dat ze stránek ve formátu XML (v hranatých závorkách je podmínka)

Výraz podmínky zadávejte pouze do pole Podmínka formuláře.

Zde je příklad :

```
//oblast[contains(@nazev,'republika')]/mis:ol[@kde="Praha"]/@pocet_nehod
```

- // - kdekoliv v dokumentu hledat
- oblast - element s názvem oblast
- [...] - který splňuje podmínku
- contains (@nazev,"republika") - že se jedná o atribut s „nazev“ obsahující podřetězec „republika“
- / - a přímo v tomto elementu
- mis:ol - hledat element mis:ol
- [...] - který splňuje podmínku
- [@kde="Praha"] - že jeho atribut „kde“ má hodnotu „Praha“
- / a v něm
- @pocet_nehod - vybrat atribut pocet_nehod - jeho hodnota je výsledek

další informací viz web - php funkce xpath

Formulář Výběr a zobrazení stažených dat

Jestliže už stahujete nějaká data pravidelně, můžete si je zobrazit. Po stisku tlačítka Zobrazit se vám má na obrazovce objevit výstupní soubor, ve formátu, který jste zvolili v kolonce Výsledek a to jako :

Webová stránka

XML soubor

Textový csv soubor (oddělovacím znakem je středník)

POZOR!! jestliže zvolíte pro výběr špatnou kombinaci očí Typ dat, Místo, Od, Do prohlížeč vrátí prázdný seznam() vybraných záznamů).

Seznam kritérií k výběru :

Typ dat, Místo - jsou to položky vámi zadané již ve vstupním formuláři

Od, Do - zde předvyplněn datum stažení nejstarších a nejnovějších dat absolutně

Hodnoty - lze zobrazit všechny hodnoty, průměr, nebo součet

Třídění - řídíte vzestupně podle Data, Místa, Hodnoty

5. Zhodnocení

Tato práce popisuje strukturu webu a nástroje, pro jeho tvorbu. V úvodní části jsou hrubě popsány některé možné aplikace pro vytvoření webové stránky.

Praktická část ukazuje možnosti zpracování užitečných dat, nezávisle na platformě zdrojové www stránky. K práci jsem využil nástroje PHP, který již má k tomuto účelu vytvořenu celou řadu hotových interních funkcí a je v tomto směru nejlepším řešením. Aplikace je použitelná ke stahování širokého spektra různých oborů, např. zpracování některých uveřejněných výsledků měření.

Aplikace je variabilní a víceúčelová z hlediska vstupů. Výstupy www technologiemi jsou omezené a dále se počítá s využitím stažených dat v jiných nástrojích určených pro statistiky, grafy (např. MS Excel). Případně lze výstupy z těchto skriptů použít jako vstupy pro další webové stránky, například pro porovnávání shodných dat z několika nezávislých zdrojů. Aplikaci je možné rozšířit o tvorbu grafů viz popis některých aplikací v bodě 3, nebo ji dále rozšířit.

Data lze z databáze aplikace získat kromě standardního výstupu založeného na konkrétních dotazech i přes grafické rozhraní a dále je libovolně zpracovat dle uvážení. Příkladem může být výše zmiňovaný freeware HeidiSQL.

Vytvořením aplikace jsem získal další vědomosti z oblasti HTML, XML, PHP, RSS, tvorby dotazů SQL, regulárních výrazů a dalších, které jsou popsány v práci.

Seznam použité literatury

- [1] Kosek J. : PHP - tvorba interaktivních internetových aplikací, Grada 1999.
- [2] Kosek J. : XML pro každého, Grada 1999.
- [3] Jiří Bráza : Učebnice základů jazyka PHP 4, Grada 2003.
- [4] Kosek J. : HTML, tvorba dokonalých WWW stránek, Grada 1999.
- [5] Satrapa : Tvorba WWW stránek. Computer Press.
- [6] Kosek J. : XML pro každého , Grada 2000
- [7] Počítačové časopisy CHIP, Computer, atd.

Zdroje z Internetu

- [1] <http://www.builder.cz>
- [2] <http://www.interval.cz>
- [3] <http://www.kosek.cz>
- [4] <http://www.php.net>
- [5] <http://www.root.cz>
- [6] <http://www.triky.cz>
- [7] www.linuxsoft.cz
- [8] www.zive.cz
- [9] <http://tvorba-webu.zdarek.com>

Příloha

Součástí diplomové práce je popisovaná praktická ukázka. Na přiloženém CD je program sloužící pro zpracování dat z internetu .