

Model-View-Controller architektura PHP frameworks

Zpracovali

Bc. Petr Fořt a Tomáš Přibyl

ČVUT FJFI KSE ASI, ZS 2012/2013, 18OOP

Obsah

Úvod do MVC

- Základní informace o MVC a návrhových vzorech
- Informace o vzniku
- Praktické příklady

PHP frameworky

- Úvodní informace, historie vzniku
- Přehled, příklady a ukázky vybraných
- Porovnání výhod a nevýhod

Závěrečné shrnutí

Informace o návrhových vzorech

MVC (Model, View, Controller) = návrhový vzor, někdy také struktura nebo architektura aplikace

Návrhové vzory (Design Patterns)

- Informační systémy si mohou být do jisté míry podobné, stejně tak řešení problémů v nich
- Návrhové vzory v programování představují obecné řešení určitých problémů
- Rozvoj především s rozšířením OOP, je snažší použít již stávající řešení, platí princip znovupoužitelnosti návrhového vzoru

Informace o návrhových vzorech

Existují desítky až stovky návrhových vzorů

Základní dělení návrhových vzorů:

Creational Patterns

- Řeší problémy související s vytvářením objektů
- Např. postup dynamického výběru třídy nového objektu za běhu aplikace a zajišťují jejich správný počet

Structural Patterns

- Možnosti uspořádání jednotlivých tříd/komponent v systému

Behavioral Patterns

- Řeší chování systému, využívají především dědičnosti

Informace o návrhových vzorech

Creational Patterns

Prototype

Abstract Factory

Singleton (Jedináček)

Factory Method

Structural Patterns

Composite (Strom, Složenina)

Decorator (Dekorátor)

Creational Patterns

Iterator (Iterátor)

Strategy (Strategie)

Observer (Pozorovatel)

State (Stav)

MVC

U MVC je aplikace rozdělena do 3 logických částí

1. Model

Zprostředkovává přístup k datům z datového úložiště (databáze)

Stará se o business logiku aplikace

2. View (Pohled)

Zobrazuje uživatelské rozhraní (např. vykreslení HTML stránky)

3. Controller (Řadič, Kontroler)

Řídí aplikační logiku a tok událostí v aplikaci

Interakce od uživatele může mít za následek provedení změny v databázi (model), případně změnit vzhled (view)

Informace o vzniku MVC

- Model formuloval Trygve Reenskaug v roce 1979 při návštěvě Xerox PARC (Palo Alto Research Center) jako část Smalltalk systému
- MVC byla koncipována jako **obecné řešení problému uživatelů** ovládající velké a složité soubory dat

Trygve Mikkjel Heyerdahl Reenskaug

* 1930, Norsko, profesor na univerzitě v Oslu



MVC

Proč znát a **používat MVC**?

- Přehlednost rozsáhlejšího kódu
- Snažší kooperace a specializace členů týmu
- Požadavky trhu práce

Kde se lze **setkat s MVC**? Příklady:

- JavaServer Faces, Jakarta Struts, NeXTSTEP
- Microsoft Foundation Classes (MFC), ASP.NET
- drtivá většina **PHP frameworků**, a další..

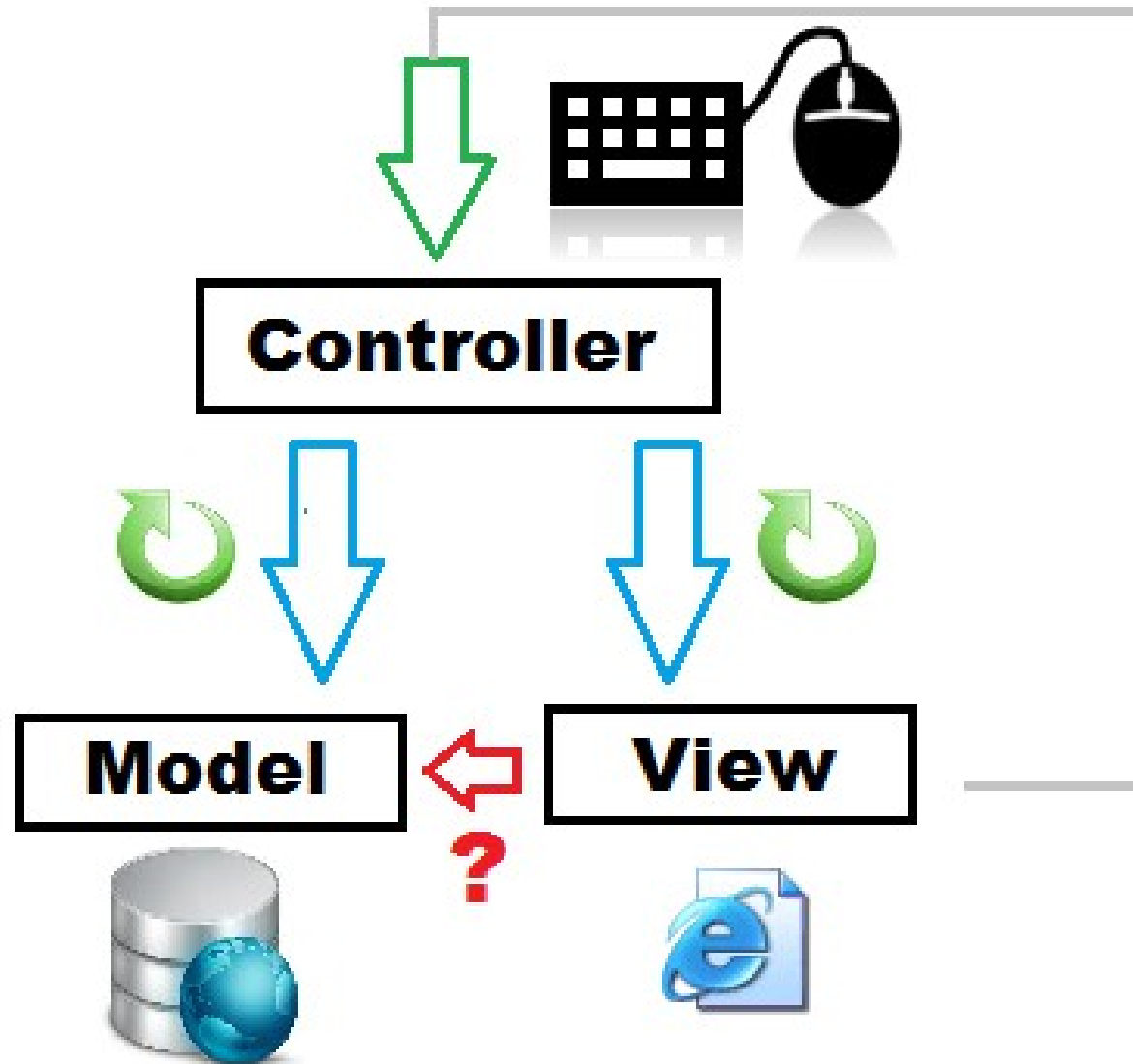
MVC

Model 1 = „poloviční MVC“, odděluje jen datový model od uživatelského rozhraní s řídicí logikou.

Kde se lze **setkat s realizací Model-1?**

- Swing (knihovna pro tvorbu grafického uživatelského rozhraní v jazyce Java)

MVC



Příklady k MVC

Příklad 1

Prezentace funkce jednotlivých nezávislých částí
M, V a C na triviálním příkladě v Excelu

Správný návrh může být leckdy obtížný. Co vše
má patřit do té které části?

Poučka: „*Prohodíte-li Controller a View a všechno bude
dobře fungovat, pak máte MVC návrh správně*“

MVC

Co vše patří do modelu? Typické příklady:

- Práce s daty aplikace (získání dat, práce s daty a předání informace o změně dat do Controlleru)
- Informace o autentizaci uživatele a jeho oprávněních, ale neřeší se už to, jakým způsobem je získal
- Hodnoty proměnných, session a cookies v aktuální relaci, které budou mít návaznost na databázi (typicky obsah košíku, apod.)
- Validace vstupních dat
- Patří sem to, co by mohly ošetřovat procedury v databázi

MVC

Co vše patří do modelu? Typické příklady:

Bussiness logika – příklady:

- Nelze dokončit objednávku před splacením.
- Nelze přijímat objednávky o svátcích.
- A další.. záleží na požadavcích konkrétního systému.

Do modelu zásadně nepatří věci, které jsou závislé na Controlleru nebo View.

MVC

V ideálním případě by měl model sloužit k odstínění od konkrétní databázové implementace, od konkrétních databázových tabulek a atributů

Příklad 2

```
$user = new User;  
$user->setUsername('Petr');  
$user->setPassword('heslicko');  
$user->store();
```

Rozdíl oproti:

```
$user = User::create('Petr', 'heslicko');
```

Příklady k MVC

Existuje velký počet návodů, jak si postavit vlastní MVC framework v PHP

Doporučuji:

- TinyMVC
- KissMVC
- tutoriál ze stránek:

<http://php-html.net/tutorials/model-view-controller-in-php>

Příklad 3

Ukázka a popis funkčního kódu z php-html.net

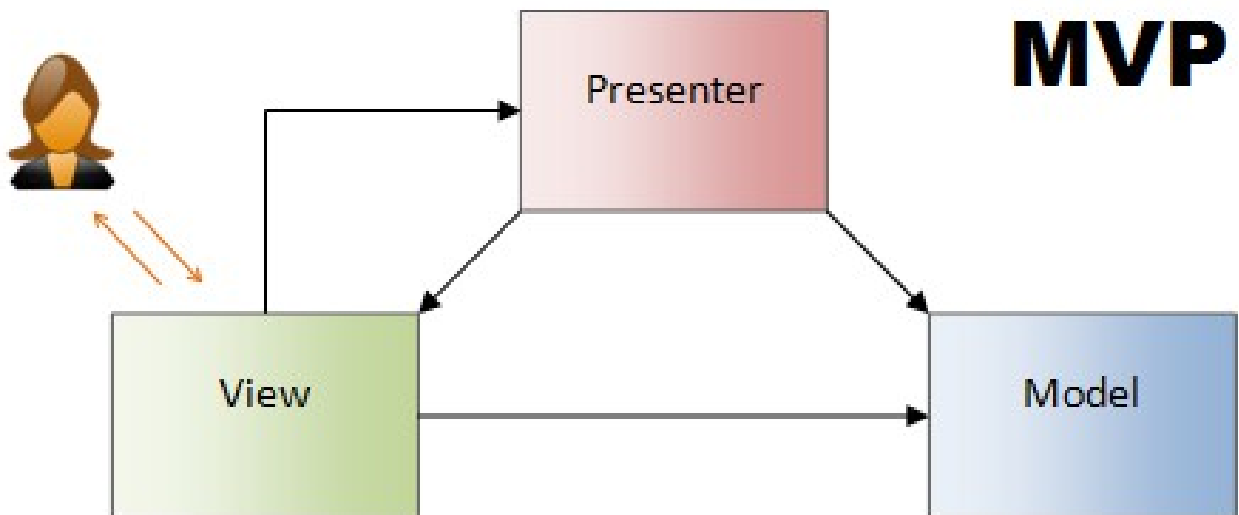
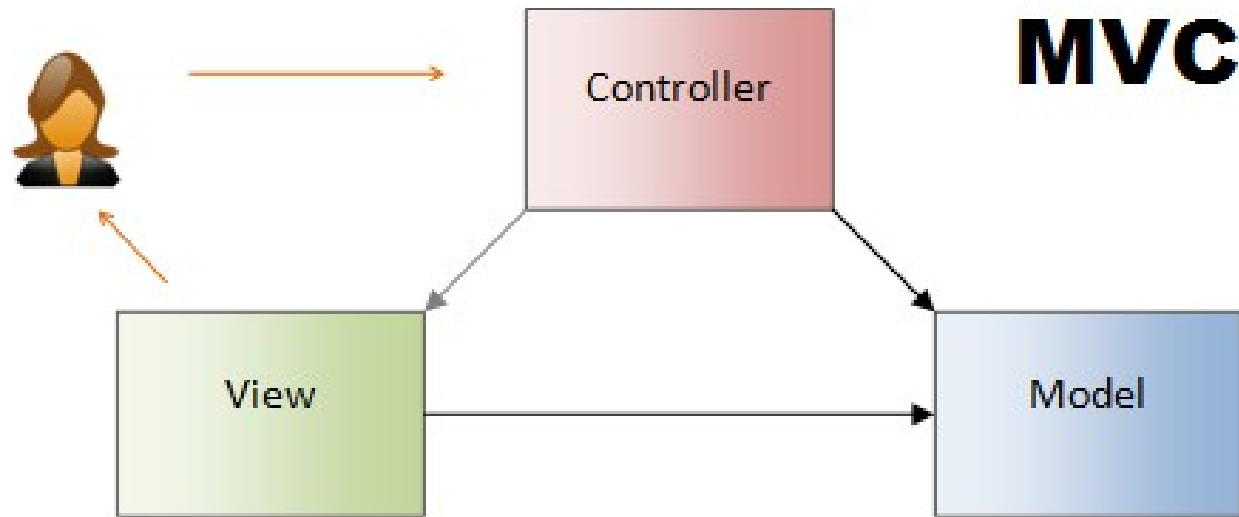
MVC vs. MVP

- **Architektura MVP** (Model, View, Presenter) je odvozená od modelu MVC
- **Příklady**, kde se používá MVP:
 - Nette Framework
 - Některé RIA technologie (Rich Internet Application):
např. Flex, Silverlight, JavaFX
 - ASP.NET Web Forms

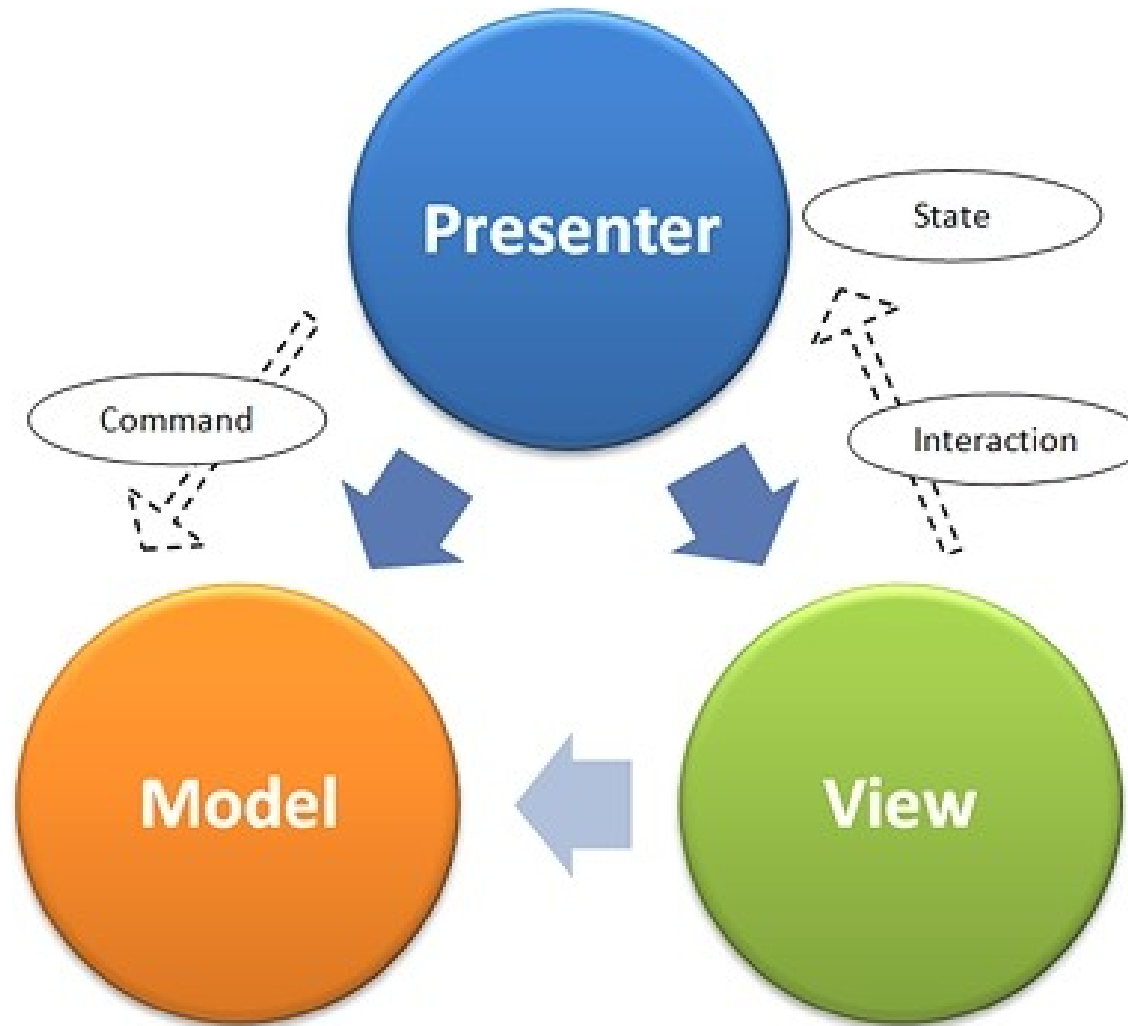
MVC vs. MVP

- Shodná zůstala pouze funkce Modelu
- Presenter obvykle pracuje přímo s View, kromě toho obsahuje aplikační a prezentační logiku (odtud také název Presenter)
- View u MVP navíc oproti MVC zpracovává uživatelský vstup (typicky volá nějakou metodu Presenteru – např. tlačítko Save → metoda OnSave)

MVC vs. MVP



MVC vs. MVP



PHP frameworky

Framework obsahuje **knihovny**, které **usnadňují** vývoj aplikace v nějakém programovacím jazyce

Kromě PHP frameworků existují i jiné: JQuery, .NET, ..

Výhodou bývá především:

- **Rychlost** vývoje, lepší kooperace v týmu
- **Vyšší bezpečnost**
- Snadná portace na **jiný databázový** systém
- u PHP implementace „hezkých url“ (**SEO Friendly URL**)

Bez: <http://a-tym.ic.cz/?stranka=skola>

S: <http://ajtaci.net/programovani/programujeme-pro-android-dil-5/>

PHP frameworky

- Nejčastěji se lze setkat s těmito frameworky:

Zend Framework

Nette Framework

Ruby on Rails (RoR)

Symfony

CakePHP

CodeIgniter

Prado

PHP frameworky

Časté pojmy

- **Autoload**: automatické načítání tříd (namísto jednotlivého includování)
- **Routing**: přesměrovávací mechanismus, pomocí kterého se po zadání URL zobrazí příslušné view
- **Minified verze**: framework v jediném souboru
- **Template (šablona)**: obdoba HTML, kde jsou navíc, a nemá příponu .htm(l)
- **ORM (Object Relation Model)**: způsob, jakým se zajišťuje propojení objektů s databází (tabulkami)

Vznik PHP frameworků

- Dříve bylo jen **čisté PHP**, kdy vývoj php stránek začínal od čistého listu
- Ve snaze ulehčení a zrychlení vývoje vznikaly první knihovny s **obecnými funkcemi** a třídami
- Potom přišly **šablony**, které už podstatně ulehčily vývoj webu v PHP (i pro začátečníky)
- Dnes již PHP frameworky založené na OOP

Vznik PHP frameworků

- První PHP frameworky vznikaly v době, kdy se používalo PHP 4, díky nedokonalé OOP neměly velký úspěch
- Největší rozmach PHP frameworků začal s vydáním verze PHP 5.0, kdy bylo PHP obohaceno o plně funkční OOP model
(Pozn.: Ve verzi 5.3 bylo PHP obohaceno o další funkce, např. o jmenné prostory a ošetřování výjimek pomocí `try-catch`)
- Dnes existují až stovky různých frameworků

PHP frameworky – Nette

- Kvalitní **český framework**, šířený pod open source licencí (New BSD nebo GNU GPL 2.0 / 3.0)
- Autor: David Grudl
- Ke stažení: <http://nette.org/> (3.6 MB ZIP, 2.0.5 stable)
- Kvalitní **dokumentace** v češtině: <http://docs.nette.org>
(dříve spousta chyb, dnes již je zpracována kvalitně).
- Vhodný **i pro začátečníky**
- V ČR široká **komunita**
- Vhodný spíše pro méně náročné projekty

PHP frameworky – Nette

- Používá návrh **MVP**
- Automatická **filtrace bezpečnostních hrozeb**:
 - Session hijacking, session stealing, session fixation
 - URL attack, control codes, invalid UTF-8
 - Cross-Site Request Forgery (CSRF)
 - Cross-Site Scripting (XSS)
- U starších projektů mohou nastat problémy verze Nette **s prefixy a bez prefixů** jmenných prostorů (při nekompatibilitě nutné přepsat: např. `Form` vs. `NForm`)
- Pohodlné ladění díky nástroji „Laděnka“ (**Debugger**)

PHP frameworky – Nette

- Podporuje **ORM**, práce s databází přes rozhraní DiBi
- Podporuje vlastní **šablonovací systém**
- Konfigurace pomocí úpravy **INI a PHP souborů**

2 režimy práce:

- **Debug mód**: nabízí veškeré informace pro programátora s pohodlným výpisem vzniklých chyb
- **Normální režim**: veškeré informace jsou skryté, logování chyb mimo dosah běžného uživatele (například do souboru)

PHP frameworky – Nette

Ukázka modelu:

```
class Members extends DibiTable
{
    public function get_member_name($id=0)
    {
        $row = $this->fetch(array('id' => $id));
        return $row->name;
    }
}
```

PHP frameworky – Nette

Ukázka pohledu:

```
<h1>{$title}</h1>
<table border="1" cellspacing="5">
  <tr>
    <th>Id</th>
    <th>Jmeno</th>
    <th>Email</th>
    <th>Telefon</th>
    <th>Akce</th>
  </tr>
  {foreach $users as $user}
  <tr>
    <td>{$user->ID}</td>
    <td>{$user->name} {$user->surname}</td>
    <td>{$user->email}</td>
    <td>{$user->phone}</td>
    <td><a href="{ $component->link('edit', $user->ID) }">Edit</a></td>
  </tr>
  {/foreach}
</table>
```

PHP frameworky – Nette

Ukázka kontroleru:

```
public function handleSave($id = 0)
{
    $request = $this->request;
    if (!$request->isPost()) return;

    if ($id !== 0) {
        $data = array(
            'name'      => trim($request->post['name']),
            'surname'   => trim($request->post['surname']),
            'phone'     => trim($request->post['phone']),
            'email'     => trim($request->post['email']));
        $user = new Users();
        $user->update($id, $data);
    }

    $this->redirect('default/' . $request->post['id_member']);
}
```

PHP frameworky – Nette

Příklad 4

Zprovoznění Nette na lokálním serveru (sandbox)

Nastavení módu ladění a produkčního módu

Příklad 5

Demonstrační ukázky, které jsou součástí Nette:

`\NetteFramework-2.0.5-PHP5.3\examples\...`

- CD-collection (sbírka CD)
- Micro-Blog (malý blog)
- Fifteen (jednoduchá hra)

PHP frameworky – Nette

Příklad 6

Automat na kávu

PHP frameworky – Nette

Proč používat Nette?

- Častý požadavek na českém trhu práce
- Rychlý vývoj aplikací:

Spousta připravených doplňků: <http://addons.nette.org/cs/>

Snadná koordinace práce v týmu (HTML kodér, programátor)

- Kvalitní dokumentace a mnoho uživatelských fór v češtině

Příklady, kdo Nette již používá:

- Root.cz, Lupa.cz, Slevomat, ESET,
Mladá Fronta, OVB Alffianz, klaus.cz, ...

PHP frameworky – Zend

- Robustní a propracovaný PHP MVC framework od tvůrců PHP, ke stažení: <http://framework.zend.com/> (2.5 MB ZIP, verze 2.0.2)
- Poměrně složitý, obsahuje dlouhé názvy
- Pokrývající veškeré potřeby při tvorbě web. aplikací
- Profesionální editor Zend Studio (Trial verze zdarma)
- Celosvětová komunita
- Od verze 2 vylepšena velikost i rychlost (oproti 30 MB)
- Určený především pro rozsáhlé komeční projekty

PHP frameworky – Zend

- Podporuje nadstandardní **validační metody** (IP adresu, datum, reg. výrazy..) a spoustu **kryptografických** algoritmů
- Používá **rozhraní PDO** a podporuje celou řadu databází (MySQL, MSSQL, SQLite, Oracle, PostgreSQL, IBM DB2, ..)
- **Šablonovací systém**: používá .phtml nebo SMARTY
- Konfigurace přes INI nebo XML (`Zend_Config_Ini`, `Zend_Config_Xml` načítají konfigurační data)
- Lze využít XML i ke tvorbě menu
- Velká řada **modulů**: např. ověřování uživ. práv (`Zend_Acl`), posílání emailů (`Zend_Mail`), tvorba PDF (`Zend_Pdf`)

PHP frameworky – Zend



PHP frameworky – Zend

Ukázka modelu:

```
<?php

class Users extends Zend_Db_Table
{
    protected $_name = 'users';
}

?>
```

PHP frameworky – Zend

Ukázka pohledu:

```
<table cellpadding = "10" border="1" cellspacing = "5">
<?php foreach($this->useri as $user) : ?>
<tr>
<td><?php echo $this->escape($user->ID);?></td>
<td><?php echo $this->escape($user->name) . " " .
    $this->escape($user->surname);?></td>
<td><?php echo $this->escape($user->email);?></td>
<td><?php echo $this->escape($user->phone);?></td>
<td>
    <a href="<?php echo $this->baseUrl; ?>/users/edit/id/<?php
        echo $user->ID.'/id_clena/'.$user->id_member;?>">Edit</a>
</td>
</tr>
<?php endforeach; ?>
</table><br />
<a href="<?php echo $this->baseUrl; ?>/index">Zpět na seznam členů</a>
```

PHP frameworky – Zend

Ukázka kontroleru:

```
require_once 'Zend/Controller/Action.php';
require_once 'Zend/Session/Namespace.php';
require_once 'Zend/Form.php';
require_once 'Zend/Form/Element/Text.php';

class UsersController extends Zend_Controller_Action
{
    function indexAction()
    {
        $useri = new Users();
        $id = (int)$this->_request->getParam('id', 0);
        $member = new Members();
        $this->view->member = $member->fetchRow('ID='.$id);
        $this->view->title = "Uživatelé člena ".$this->view->member->name;
        $namespace = new Zend_Session_Namespace();
        if (isset($namespace->vypis)) {
            $this->view->vypis = $namespace->vypis;
            unset($namespace->vypis);
        }
        if ($id > 0) $this->view->useri = $useri->fetchAll('id_member='.$id);
    }
}

?>
```


PHP frameworky - Zend

Příklad 7

Demonstrační ukázky, které jsou součástí:

`\ZendFramework-2.0.2\demos\Zend\ProgressBar\`

- `JSPush.php`
- `Upload.php`

PHP frameworky – CakePHP

- Poměrně oblíbený; rozsáhlý, ale pomalejší

Ke stažení: <http://cakephp.org/>

(1.8 MB ZIP, 2.2.2 stable)

- Celkem jednoduchý na naučení
- Dlouhý vývoj nové verze frameworku
- Dobrá komunita

PHP frameworky – CakePHP

Ukázka modelu:

```
class User extends AppModel
{
    var $name = 'User';

    function save($data = null, $id) {
        $this->id = $id;
        $returnval = parent::save($data, false);
        return $returnval;
    }
}
```

PHP frameworky – CakePHP

Ukázka pohledu:

```
<?php foreach( $members as $member) { ?>
<tr>
    <td><?php echo $member['Member']['ID']    ?></td>
    <td><?php echo $member['Member']['name'] ?></td>
    <td><?php echo $member['Member']['type'] ?></td>
    <td><?php echo $member['Member']['town'] ?></td>
    <td><?php echo
$html->link('edit', '/members/edit' . $member['Member']['ID']) . " " .
$html->link('uživatelé', '/users/index' . $member['Member']['ID']); ?>
    </td>
</tr>
<?php
}
?>
```

PHP frameworky – CakePHP

Ukázka kontroleru:

```
function edit($id_user, $id_member) {
    if(empty($this->data)) {
        $this->pageTitle = 'Editace uživatele';
        $this->set('users', $this->User->find("id=$id_user"));
        $this->set('id_mem', $id_member);
        $this->render();
    } else {
        $this->cleanUpFields();
        if($this->User->save($this->data, $id_user)) {
            $this->Session->setFlash('Uživatel byl úspěšně upraven.');
```

```
            $this->redirect('/users/view/'.$id_member);
        } else {
            $this->Session->setFlash('Data se nepodařilo upravit.');
```

```
            $this->redirect('/users/view/'.$id_member);
        }
    }
}
```

PHP frameworky – Ruby on Rails

- Velmi populární framework
- Ke stažení: <http://rubyonrails.org/>
(11 MB spustitelná instalace, verze 3.2.8)
- Někdy se též používá zkrácenina RoR
- Spousta frameworků z něj čerpají inspiraci
- Tvůrce: David Heinemeier Hansson

PHP frameworky – Ruby on Rails

Ukázka modelu:

```
class Users < ActiveRecord::Base 2 end
```

PHP frameworky – Ruby on Rails

Ukázka pohledu:

```
<h1><%= @title %></h1>
<p style="color: green"><%= flash[:notice] %></p>
<table border="1" cellspacing="5">
  <tr>
    <th>Id</th>
    <th>Jmeno</th>
    <th>Email</th>
    <th>Telefon</th>
    <th>Akce</th>
  </tr>
  <% for users in @users %>
    <tr>
      <td><%= users.id %></td>
      <td><%=h users.name+' '+users.surname %></td>
      <td><%=h users.email %></td>
      <td><%=h users.phone %></td>
      <td><%= link_to 'Edit', '/users/edit/'+users.id.to_s %></td>
    </tr>
  <% end %>
</table>
```


PHP frameworky – Ruby on Rails

Ukázka kontroleru:

```
class UsersController < ApplicationController
  def index
    @users = Users.find(:all, :conditions => ["id_member = ?", params[:id]])
    @member_name = Members.find(params[:id]).name
    @title = 'Uživatelé člena ' + @member_name
  end
  def edit
    @users = Users.find(params[:id])
    @member_name = Members.find(@users.id_member).name
    @title = 'Editace uživatele ' + @users.name + ' ' + @users.surname
  end
  def update
    @users = Users.find(params[:id])
    if @users.update_attributes(params[:users])
      flash[:notice] = 'Users was successfully updated.'
      redirect_to('/users/index/'+@users.id_member.to_s)
    else
      render :action => "edit"
    end
  end
end
End
```

PHP frameworky – CodeIgniter

- Oblíbený a **jednoduchý** PHP MVC Framework
- Ke stažení: <http://codeigniter.com/>
(2.2 MB ZIP, verze 2.1.2)
- Zpočátku nebyl vyvíjen komunitou, ale společností EllisLab, Inc. (která později založila komunitu CodeIgniter International)
- Některá řešení jsou intuitivnější než v CakePHP
- Stále podporuje **zastaralé** PHP4

PHP frameworky – Symfony

- Robustní a **poměrně pomalý** open source PHP MVC framework
- Ke stažení: <http://symfony.com/download>
(3.7 MB TGZ, verze 2.1.12 Standard)
- Je vhodný pro **rozsáhlejší komerční** projekty, v některých případech je vhodnější než použití Zend Frameworku

Framework	Zend	Nette	CakePHP	CodeIgniter	DIY	Fusebox	Prado	Symfony
PHP4	ne	ne			ne		ne	ne
MVC		MVP						
Různé DB					ne			
ORM				ne		ne		
Objekt. DB						ne		
Šablonovací systém			ne			ne		ne
Validace					ne	ne		
AJAX				ne				
Moduly				ne	ne			
Autentizační modul				ne	ne	ne		

Shrnutí

- MVC: Model, View, Controller a jejich funkce
- Model-1 není MVC, MVP není MVC
- PHP Frameworky:
 - Výhody užití (rychlost, bezpečnost, koordinace),
Zend, Nette, RoR, CakePHP, ad.
 - Rozdíly mezi frameworky

Použité zdroje

<http://objekty.vse.cz/Objekty/Vzory-uvod#CoPred>

http://cs.wikipedia.org/wiki/N%C3%A1vrhov%C3%BD_vzor

<http://www.root.cz/clanky/velky-test-php-frameworku-2008/>

<http://programujte.com/clanek/2008022000-php-frameworky/>

<http://www.phpframeworks.com/>

<http://www.zdrojak.cz/clanky/nette-framework-mvc--mvp/>

<http://doc.nette.org/>

Použité zdroje

<http://www.zdrojak.cz/clanky/uvod-do-architektury-mvc/>

<http://php-html.net/tutorials/model-view-controller-in-php/>

<http://stackoverflow.com/questions/2056/what-are-mvp-and-mvc-and-what-is-the-difference>

<http://www.zdrojak.cz/clanky/nette-framework-mvc--mvp/#ic=articles-related&icc=uvod-do-architektury-mvc-396>

<http://tournasdimitrios1.wordpress.com/2011/04/13/zend-framework-customizing-the-directory-structure/>

<http://root.cz/clanky/velky-test-php-frameworku-zend-nette-php-a-ror>

<http://www.zdrojak.cz/clanky/prezentacni-vzory-zrodiny-mvc/>

Použité zdroje

<http://nette.org/>

<http://framework.zend.com/>

<http://cakephp.org/>

<http://rubyonrails.org/>

<http://codeigniter.com/>

<http://www.symfony-project.org/>

Děkujeme za pozornost