Dr. Pál László, Sapientia EMTE, Csíkszereda

WEB PROGRAMOZÁS
6.ELŐADÁS

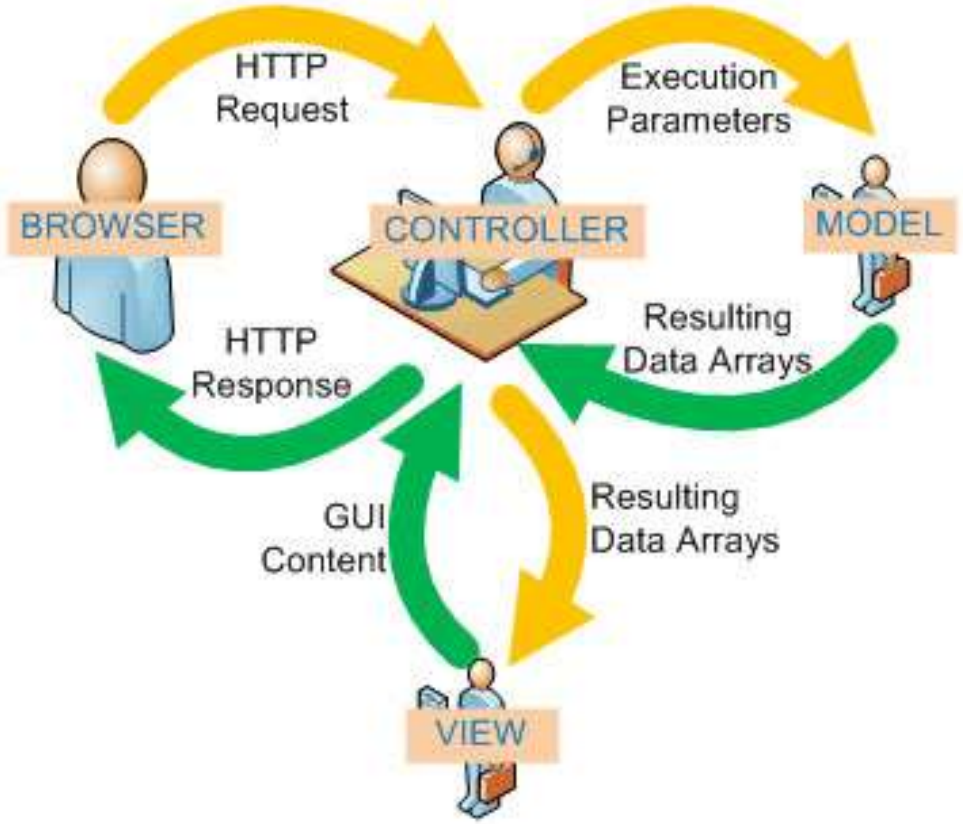
2015-2016

MVC (Modell View Controller) modell

Modell-View-Controller

- Modell
  - Alkalmazás adatai és folyamata
  - Üzleti logika
- Nézet
  - Web design, sablon, HTML, CSS, JavaScript
  - Modell adatait jeleníti meg megfelelően formázva
- Vezérlő
  - Folyamatot irányítja, begyűjti az input adatokat és azokat a modellnek és a nézetnek átadja

MVC architektúra



Modell

- Üzleti logika
- Alkalmazás adatai
  - Tárolás, feldolgozás, absztrahálás
- Független a nézettől és a HTTP kérés feldolgozásától
  - Általában nincs benne HTML és pl. $_GET
- Általában adatbázisokkal dolgozik
  - Adatbázis-elérési absztrakció (réteg)
  - Adatbázis-absztrakció (réteg)

Nézet

- Megjelenítés
  - HTML, CSS, JavaScript, XML, PDF, szöveg, kép
- Nem tartalmazhat logikát, adatfeldolgozást
- Sablonnyelv használata
  - PHP
  - alternatív szintaxis
  - limitált utasításkészlet
- Smarty, stb.

Vezérlő

- Az MVC minta legáltalánosíthatóbb része
- Kérés feldolgozása, adatok továbbítása a modellnek és nézetnek
- Több további részekre bontható
  - bootstrap file
  - Front Controller
  - Action Controller

MVC előnyök

- Fejlesztési munkafolyamatokat támogatja
- Három fő szerepkör van egy fejlesztői csapatban
  - Fejlesztők
    - Modellen dolgoznak (PHP, adatbázis, algritmus, architektúra)
  - Tervezők
    - Nézeten dolgoznak (HTML, CSS, JavaScript, grafikus elemek, mockup, stb.)
  - Integrátorok
    - Vezérlőn dolgoznak, ők kapcsolják össze a modellt a nézettel, kevésbé képzettek, mint a fejlesztők

MVC keretrendszerek

- bizonyos filozófiának megfelelő szabályok gyűjteménye
- ahány keretrendszer, annyi féle
- szabályok korlátokat is jelentenek
- egységesebb alkalmazásfejlesztés
- szétválasztott kód és logika
- meghatározott könyvtárszerkezet
- csoportmunka támogatott
- rétegek szétválasztása

MVC keretrendszerek

- CodeIgniter
- Symfoni2
- Laravel
- Yii
- Zend
- Nette

HelloWorldMVC példa


```php
class Model {

    public $text;

    public function __construct() {

        $this->text = 'Hello world!';
    }

}
```

```php
class Controller {

    private $model;

    public function __construct(Model $model) {

        $this->model = $model;
    }
}
```

```php
class View {

    private $model;

    public function __construct(Model $model) {

        $this->model = $model;
    }

    public function output() {

        return '<h1>' . $this->model->text . '</h1>';
    }

}
```

```php
$model = new Model();
$controller = new Controller($model);
$view = new View($model);
echo $view->output();
```

CurrencyMCV példa


```php
//Model
class CurrencyConverter {

    private $baseValue = 0;
    private $rates = [
        'GBP' => 1.0,
        'USD' => 0.6,
        'EUR' => 0.83,
        'YEN' => 0.0058
    ];

    public function get($currency) {
        if (isset($this->rates[$currency])) {
            $rate = 1 / $this->rates[$currency];
            return round($this->baseValue * $rate, 2);
        } else
            return 0;
    }

    public function set($amount, $currency = 'GBP') {
        if (isset($this->rates[$currency])) {
            $this->baseValue = $amount * $this->rates[$currency];
        }
    }

}
```

CurrencyMCV példa


```php
//View
class CurrencyConverterView {

    private $converter;
    private $currency;

    public function __construct(CurrencyConverter $converter, $currency) {
        $this->converter = $converter;
        $this->currency = $currency;
    }

    public function output() {
        $html = '<form action="?action=convert" method="post">
        <input name="currency" type="hidden" value="' . $this->currency . '" />
        <label>' . $this->currency . ':</label>
        <input name="amount" type="text" value="' . $this->converter->get($this->currency) . '" />
        <input type="submit" value="Convert" />
        </form>';
        return $html;
    }
}
```

CurrencyMCV példa


```php
//Controller
class CurrencyConverterController {

    private $model;

    public function __construct($model) {
        $this->model = $model;
    }

    public function convert($request) {
        if (isset($request['currency']) && isset($request['amount'])) {
            $this->model->set($request['amount'], $request['currency']);
        }
    }
}
```

CurrencyMCV példa


```php
$model = new CurrencyConverter();
$controller = new CurrencyConverterController($model);

//If one of the forms has been submitted, call the relevant controller action
if (isset($_GET['action']))
    $controller->{$_GET['action']}($_POST);

$gbpView = new CurrencyConverterView($model, 'GBP');
echo $gbpView->output();

$usdView = new CurrencyConverterView($model, 'USD');
echo $usdView->output();

$eurView = new CurrencyConverterView($model, 'EUR');
echo $eurView->output();

$yenView = new CurrencyConverterView($model, 'YEN');
echo $yenView->output();
```

CurrencyMCV példa