Web программирование
Web applications

Игорь Родионов
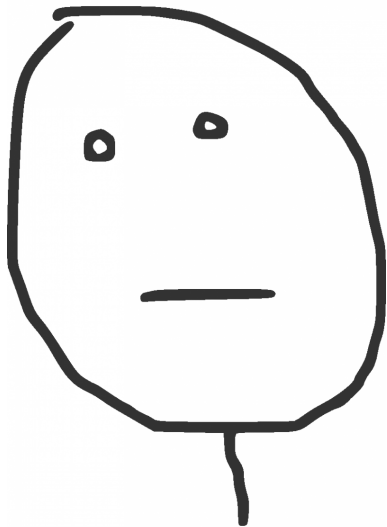
Омский Государственный Технический Университет
кафедра Информатики и вычислительной техники
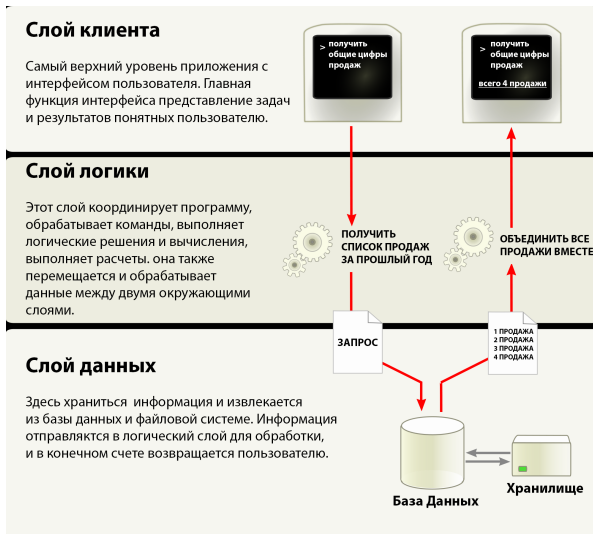
ОмГТУ, 2014.

Web application

Веб-приложение — клиент-серверное приложение, в котором клиентом выступает браузер, а сервером — веб-сервер.

Concurrency

Этот неловкий момент, когда для объяснения материала курса, надо рассказать другой курс.

3-thier architecture

Слой клиента

Самый верхний уровень приложения с интерфейсом пользователя. Главная функция интерфейса представление задач и результатов понятных пользователю.

> получить общие цифры продаж

> получить общие цифры продаж

всего 4 продажи

Слой логики

Этот слой координирует программу, обрабатывает команды, выполняет логические решения и вычисления, выполняет расчеты. она также перемещается и обрабатывает данные между двумя окружающими слоями.

ПОЛУЧИТЬ СПИСОК ПРОДАЖ ЗА ПРОШЛЫЙ ГОД

ОБЪЕДИНИТЬ ВСЕ ПРОДАЖИ ВМЕСТЕ

ЗАПРОС

1 ПРОДАЖА
2 ПРОДАЖА
3 ПРОДАЖА
4 ПРОДАЖА

Слой данных

Здесь хранится информация и извлекается из базы данных и файловой системе. Информация отправляется в логический слой для обработки, и в конечном счете возвращается пользователю.

База Данных   **Хранилище**

Вопросы?

Systems

- HTTP abstraction

- Session management

- Routing

- Data storage operations

- Business logic

- Templates

Concurrency
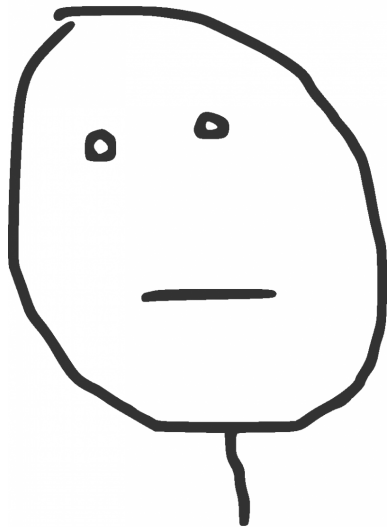
Этот неловкий момент, когда для объяснения материала курса, надо рассказать другой курс.

Systems

- Compilation / Interpretation

- Static / Dynamic typing

- Memory management

- Programming paradigm

PHP

- Interpretation

- Dynamic typing

- Garbage collector

- Procedure & OOP

Ruby

- Interpretation

- Dynamic typing

- Garbage collector

- Procedure & OOP & Functional

Node js

- Interpretation

- Dynamic typing

- Garbage collector

- Procedure & OOP & Functional

Java

- Compile

- Static typing

- Garbage collector

- OOP

Systems

- HTTP abstraction

- Session management

- Routing

- Data storage operations

- Business logic

- Templates

Questions

Вопросы?

HTTP Abstraction - PHP

```php
1  <?php
2
3  $get_param = $_GET['q'];
4  $post_param = $_POST['user_name'];
5  $is_ajax = isset($_REQUEST['ajax']) ? $_REQUEST['ajax'] :
        0;
6  $user_agent = $_SERVER['HTTP_USER_AGENT'];
7
8  foreach ($_FILES as $file) {
9    file_save($file);
10 }
```
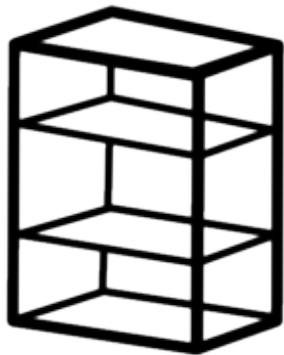
HTTP Abstraction - PHP

```php
1  if (!isset($_COOKIE['time'])) {
2    $_COOKIE['time'] = time();
3  }
4
5  header('Location: http://www.example.com/');
6  http_response_code(303);
```

HTTP Abstraction - Ruby

```ruby
1  req = Rack::Request.new(env)
2  req.post?
3  req.params["data"]
4
5  response = Rack::Response.new
6  response.header['Content-Type'] = 'text/html'
7  response.write view
8  response.finish
```

HTTP Abstraction - Ruby

```ruby
1  def GET
2    if @env["rack.request.query_string"] == query_string
3      @env["rack.request.query_hash"]
4    else
5      p = parse_query(query_string)
6      @env["rack.request.query_string"] = query_string
7      @env["rack.request.query_hash"] = p
8    end
9  end
```

HTTP Abstraction - Node js

```
1  var http = require('http');
2
3  http.createServer(function (req, res) {
4     res.writeHead(200, {'Content-Type': 'text/plain'});
5     res.end('Hello World\n');
6  }).listen(1337, '127.0.0.1');
```

HTTP Abstraction - Java

```java
public class HelloWorldServlet extends HttpServlet {

    public void doGet(HttpServletRequest req,
            HttpServletResponse resp) throws ServletException,
            IOException {
        resp.setContentType("text/html;charset=utf-8");

        PrintWriter pw = resp.getWriter();
        pw.println("<H1>Hello, world!</H1>");

    }
}
```

Вопросы?

Session - PHP

```php
<?php

session_start();
if (!isset($_SESSION['count'])) {
    $_SESSION['count'] = 0;
} else {
    $_SESSION['count']++;
}

//----------------------------------------
session_destroy();
```

Session - Java

```
1  public class SessionSample extends HttpServlet {
2    public void doGet (HttpServletRequest request, HttpServletResponse response)
3        throws ServletException, IOException {
4
5    // Step 1: Get the Session object
6
7        boolean create = true;
8        HttpSession session = request.getSession(create);
9
10   // Step 2: Get the session data value
11
12       Integer ival = (Integer)
13       session.getAttribute ("sessiontest.counter");
14       if (ival == null) ival = new Integer (1);
15       else ival = new Integer (ival.intValue () + 1);
16       session.setAttribute ("sessiontest.counter", ival);
17   }
18 }
```
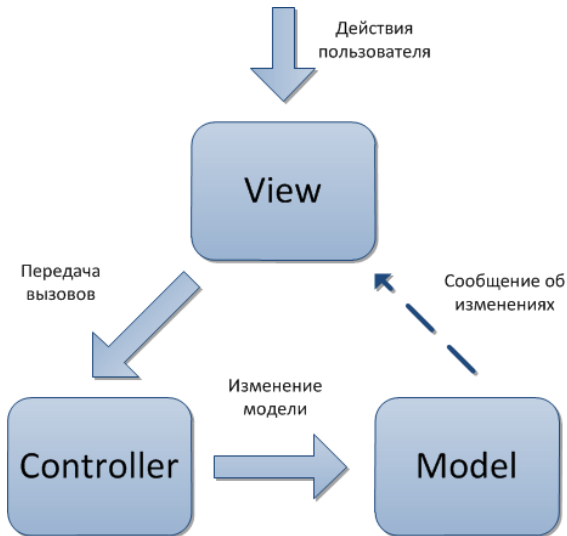
Вопросы?

Concurrency
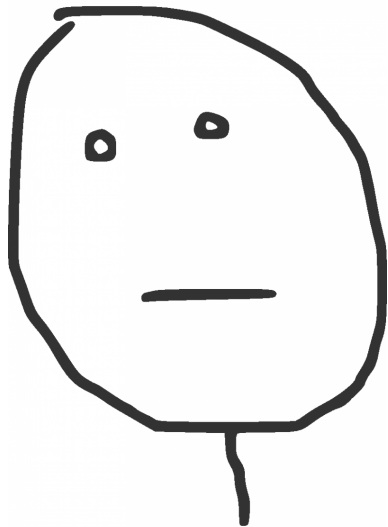
Этот неловкий
момент, когда для
объяснения
материала курса,
надо рассказать
другой курс.

Model View Controller

Model View Presenter

MVC - PHP

```php
1  class Advert_Controller_List extends Controller_Common {
2    public function run() {
3      $this->getView()->loadI18n('Common/General', 'Advert/List');
4
5      if (!$this->checkAccess()) {
6        return $this->createNotification()->setType('alert')->run();
7      }
8
9      $list = new Module_Advert_List(
10       $this->getRequest(), $this->getMapper('Advert/Advert'), new
                Pagination_Manager(10, 10, $this->getRequest())
11     );
12
13     $this->getView()->adverts = $list->getList();
14     $this->getView()->pagination = $list->getPagination();
15
16     $this->getView()->id_category = $this->getRequest()->getRequest('
            id_category');
17     $this->getView()->id_user = $this->getRequest()->getRequest('id_user
            ');
18
19     return $this->getView();
20   }
```

Вопросы?
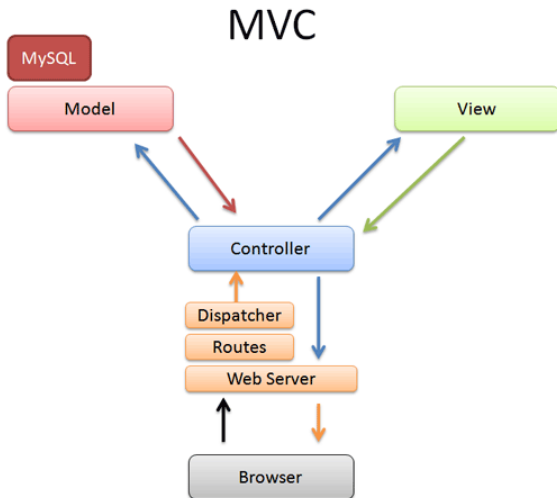
Routing

http://site.com/shop/computers - категория "Компьютеры"
http://site.com/shop/computers/23 - товар с ID = 23

```
'shop/category/<category:[\w_\/-]+>/<id:[\d]+>'=>'shop/show',
'shop/category/<category:[\w_\/-]+>'=>'shop/category',
'shop'=>'shop/index',
```
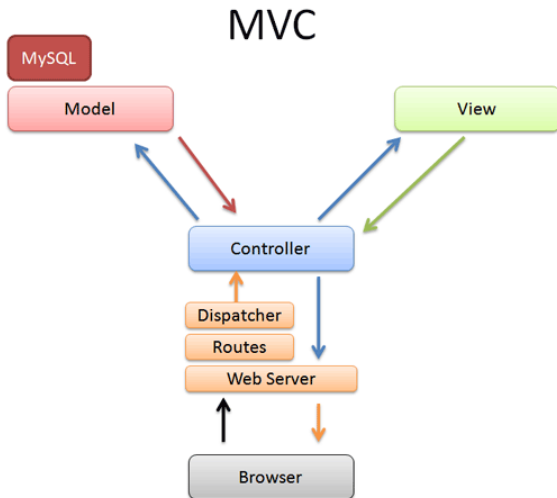
Routing

```php
1  <?php
2
3  class ShopController extends Controller
4  {
5    public function actionCategory($category)
6    {
7
8    }
9
10   public function actionShow($category, $id)
11   {
12
13   }
14 }
```

Вопросы?

MVC - PHP

Model - ORM

```
1  class Person < ActiveRecord::Base
2      attr_accessible :age, :first_name, :last_name
3  end
4
5  Person.new first_name: "Andrew", last_name: "Burgess", age: 22
6
7  /**
8    INSERT INTO "people" ("age", "created_at", "first_name", "last_name", "
          updated_at")
9    VALUES (?, ?, ?, ?, ?) [["age", 30], ["created_at", Fri, 15 Feb 2013 16:02:18
          UTC +00:00],
10   ["first_name", "John"], ["last_name", "Doe"],
11   ["updated_at", Fri, 15 Feb 2013 16:02:18 UTC +00:00]]
12
13 **/
14
15 Person.offset(0).limit(10).find_by_last_name(params[:search_term])
```

Model View Controller
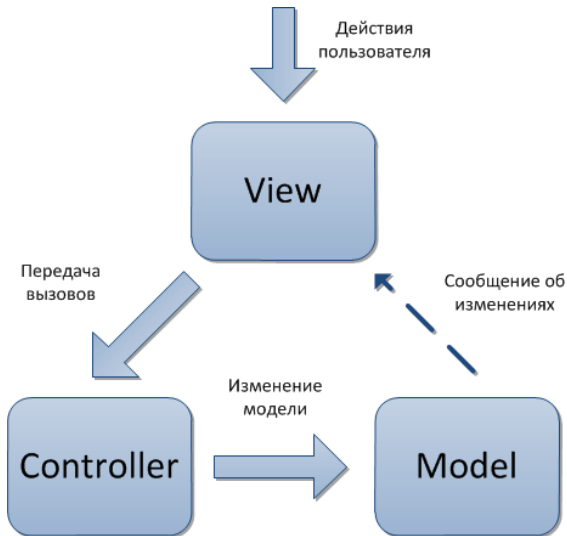
View - Templates

```
1  // create an smarty object of our customized class
2  $smarty = new Smarty() ;
3
4  // basic assignment for passing data to template file
5  $smarty->assign('title', 'Test Title');
6  $smarty->assign('description', 'Test Description');
7
8  // show the template
9  $smarty->display('index.tpl');
```

View - Templates

```
 1  <html>
 2  <head>
 3    <title>Info</title>
 4  </head>
 5  <body>
 6  Test Article:
 7  Title: {$title}
 8  Description: {$description}
 9  </body>
10  </html>
```

Вопросы?