

MVC

Model-View-Controller

Design Patterns

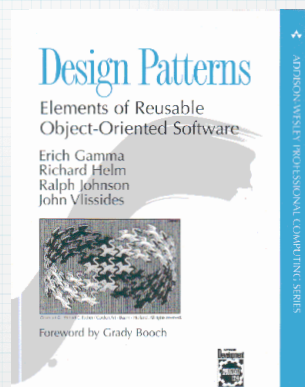
- Certain programs reuse the same basic structure or set of ideas
- These regularly occurring structures have been called **Design Patterns**

Design Patterns

Design Patterns: Elements of Reusable Object-Oriented Software

Addison-Wesley

Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides



Patterns

		Behavioral Patterns
		Chain of responsibility
		Command
		Interpreter
		Iterator
		Mediator
		Memento
		Observer
		State
		Strategy
		Template method
		Visitor
Structural Patterns		
Adapter		
Bridge		
Composite		
Decorator		
Façade		
Flyweight		
Proxy		
Proxy Clone pattern		
Creational patterns		
Abstract Factory		
Builder		
Factory Method		
Prototype		
Singleton		

MVC - design pattern

A way of structuring an application into different component parts

First described by Trygve Reenskaug working at Xerox research labs (1979)

Influential paper: Applications Programming in Smalltalk-80 - How to use Model-View-Controller

Using MVC

Model

Classes that are core to the application

person
invoice
car
booking

View

Classes that are key to the presentation / interaction with the user

HTML classes
forms

Controller

Code that uses the Model and View

Using MVC in PHP

Model

Class style

View

Class style

Controller

Imperative style

Using MVC in PHP

Model

PHP Classes
SQL
No HTML

View

PHP classes for display
HTML
No SQL

Controller

No SQL or HTML

Example: BadJokeCity.com in MVC

The screenshot displays several overlapping browser windows from BadJokeCity.com, illustrating the MVC pattern:

- Top Window:** Shows the main site header with navigation links (Main page, Cow, FAQ, User Login) and a "Today's Bad Joke" section.
- Middle Window:** Shows a "User log in page" with input fields for "Username:" and "Password:".
- Bottom Window:** Shows the "Authenticated" state with a message: "Welcome user user1, you have logged in." and buttons for "Enter joke" and "Log out page".

BadJokeCity.com in MVC

Model

Construct model classes for the application

View

Controller

Model

Joke
jokeNo
jokeTitle
jokeBody
displayed
userNo
get_JokeBody
get_JokeTitle

Joke class with same properties as joke table

A **Joke** object will be constructed with a date (generally todays date)

```
$todaysJoke=new Joke("2007-04-25");
```

Use the **get_** methods to print jokes

```
print $todaysJoke->get_JokeTitle();  
print $todaysJoke->get_jokeBody();
```

Model

A **Joke** object is constructed with a date (generally todays date)

```
$todaysJoke=new Joke("2007-04-25");
```

Needs a **SQLDate** class to get todays date in **SQL** date format

Model

SQLDate
todaysDate
get_SQLDate

SQLDate class with one property - todays date in SQL **YY-MM-DD** format

A **SQLDate** object is constructed - then **get_SQLDate()** returns todays date as a string

```
$today= new SQLDate();  
print $today->get_SQLDate();
```

Model

Joke
jokeNo
jokeTitle
jokeBody
displayed
userNo
get_JokeBody
get_JokeTitle

When the **Joke** object is constructed a SQL process occurs to get the joke from the **jokedb** table

The generic **Database** class can help here

Model

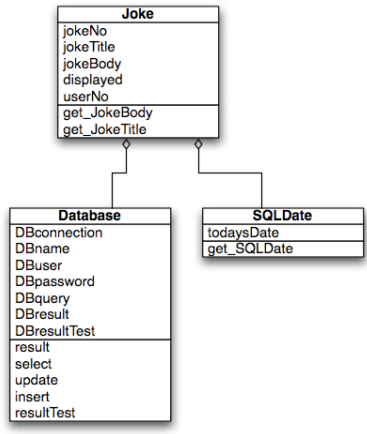
Database
DBconnection
DBname
DBuser
DBpassword
DBquery
DBresult
DBresultTest
result
select
update
insert
resultTest

DBconnection, **DBname**, **DBuser** and **DBpassword** are used in the constructor to open a connection to the **mysql** server

Methods **select**, **update**, **insert** are used to construct the SQL statement in **DBquery**

The query is then returned via **DBresult**

Model



An aggregate relationship where instances of Joke use instances of Database and SQLDate

Model

The Joke constructor uses the displayeddate and a Database object to look for todays joke ...

```
function Joke($displayedDate=""){
    ...
    $database=new Database($DBconnection,$DBdatabasename,$DBusername,
        $DBpassword);
    $database->select("*","joke,jokeday","joke.jokeno=jokeday.jokeno and
        displayedDate='".$displayedDate."'");
    $dateResult=$database->result();
    ...
}
```

Model

The results are then placed in the appropriate Joke properties

```
...
$dateRow=mysql_fetch_array($dateResult);

$this->jokeNo=$dateRow["jokeNo"];
$this->jokeTitle=$dateRow["jokeTitle"];
$this->jokeBody=$dateRow["jokeBody"];
$this->displayed=$dateRow["displayed"];
$this->userNo=$dateRow["userNo"];
}
```

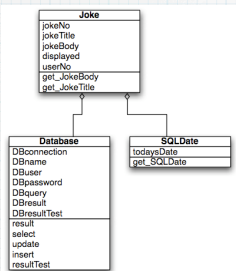
Model

The appropriate `get_` methods are used to get the `joke title` and `joke body` out of the object

```
...  
function get_JokeBody(){  
    return ($this->jokeBody);  
}  
  
function get_JokeTitle(){  
    return ($this->jokeTitle);  
}
```

BadJokeCity.com in MVC

Model



View

Now to construct
View Classes

Base this new class
on the generic HTML
class

Controller

Class - HTML

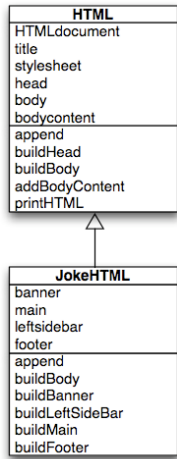
HTML
HTMLdocument
title
stylesheet
head
body
bodycontent
append
buildHead
buildBody
addBodyContent
printHTML

`title` and `stylesheet` are set in the constructor (in the `title` and `stylesheet` properties)

`addBodyContent` is used to add lines of HTML to the body (stored in `bodycontent`)

`printHTML` returns (as a string) the HTML which can be printed

Class - HTML

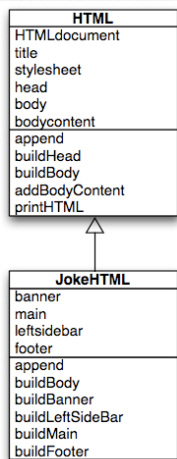


title and **stylesheet** are set in the constructor (in the title and stylesheet properties)

addBodyContent is used to add lines of HTML to the body (stored in bodycontent)

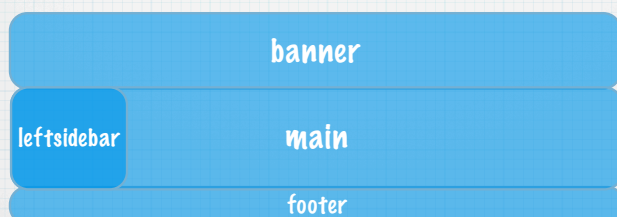
printHTML returns (as a string) the HTML which can be printed

Class - JokeHTML is a descendant of HTML

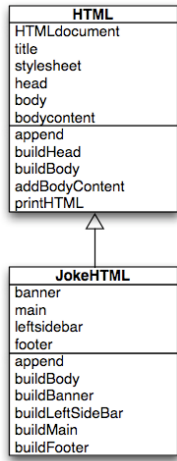


JokeHTML has extra properties for each of the displayed parts of the joke HTML

Joke page elements - each in its own div

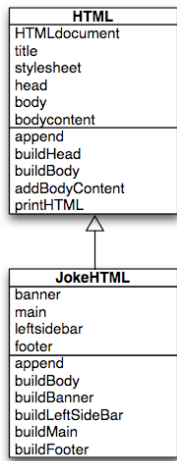


Class JokeHTML



build... are required methods to construct the appropriate banner, main, leftsidebar and footer parts

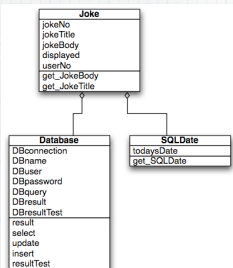
Class JokeHTML



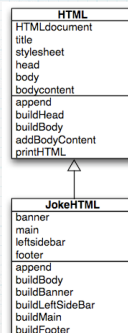
Finally both the **append** and **buildBody** methods are overridden with new **JokeHTML** versions

BadJokeCity.com in MVC

Model



View



Controller

Final step - build the controller

Controller (Imperative style)

Generally need to do the following:

- | |
|--------------------------------------|
| Include class resources |
| Check passed POST or GET parameters |
| Create / Use Objects |
| Generate response (HTML) |
| Perhaps pass control to another page |

Use sessions or database to pass values around

index.php

```
<?
require_once("../resources/class.Joke.php");
require_once("../resources/class.SQLDate.php");
require_once("../resources/class.JokeHTML.php");

$today= new SQLDate();
$todayJoke=new Joke($today->get_SQLDate());
$content= new JokeHTML("BadJokeCity.com", "../resources/bluestyles.css");
$content->buildBanner("BadJokeCity.com","The first place to look for bad jokes",
    "Todays Bad Joke");

$row_0 = array("Main page", "index.php");
$row_1 = array("FAQ", "FAQ.php");
$row_2 = array("User Login", "../user/login.php");
$arrayStructure=array($row_0,$row_1,$row_2);

$content-> buildLeftSideBar($arrayStructure);

$content-> buildMain($todayJoke->get_JokeTitle(),$todayJoke->get_jokeBody());

$content-> buildFooter($arrayStructure);

print $content->printHTML();
?>
```

index.php

```
<?
require_once("../resources/class.Joke.php");
require_once("../resources/class.SQLDate.php");
require_once("../resources/class.JokeHTML.php");

$today= new SQLDate();
$todayJoke=new Joke($today->get_SQLDate());
$content= new JokeHTML("BadJokeCity.com", "../resources/bluestyles.css");
$content->buildBanner("BadJokeCity.com","The first place to look for bad jokes",
    "Todays Bad Joke");

$row_0 = array("Main page", "index.php");
$row_1 = array("FAQ", "FAQ.php");
$row_2 = array("User Login", "../user/login.php");
$arrayStructure=array($row_0,$row_1,$row_2);

$content-> buildLeftSideBar($arrayStructure);

$content-> buildMain($todayJoke->get_JokeTitle(),$todayJoke->get_jokeBody());

$content-> buildFooter($arrayStructure);

print $content->printHTML();
?>
```

Import the
required classes

index.php

```
<?
require_once("../resources/class.Joke.php");
require_once("../resources/class.SQLDate.php");
require_once("../resources/class.JokeHTML.php");

$today= new SQLDate();
$today= new SQLDate();
$content= new JokeHTML("BadJokeCity.com", "../resources/bluestyles.css");
$content= new JokeHTML("BadJokeCity.com", "The first place to look for bad jokes",
"Today's Joke");
$row_0 = array("Main page", "index.php");
$row_1 = array("FAQ", "FAQ.php");
$row_2 = array("User Login", "../user/login.php");
$arrayStructure=array($row_0,$row_1,$row_2);

$content-> buildLeftSideBar($arrayStructure);

$content-> buildMain($today->get_JokeTitle(),$today->get_jokeBody());

$content-> buildFooter($arrayStructure);

print $content->printHTML();
?>
```

index.php

```
<?
require_once("../resources/class.Joke.php");
require_once("../resources/class.SQLDate.php");
require_once("../resources/class.JokeHTML.php");

$today= new SQLDate();
$today= new SQLDate();
$today= new Joke($today->get_SQLDate());
$content= new JokeHTML("BadJokeCity.com", "../resources/bluestyles.css");
$content= new JokeHTML("BadJokeCity.com", "The first place to look for bad jokes",
"Today's Joke");
$row_0 = array("Main page", "index.php");
$row_1 = array("FAQ", "FAQ.php");
$row_2 = array("User Login", "../user/login.php");
$arrayStructure=array($row_0,$row_1,$row_2);

$content-> buildLeftSideBar($arrayStructure);

$content-> buildMain($today->get_JokeTitle(),$today->get_jokeBody());

$content-> buildFooter($arrayStructure);

print $content->printHTML();
?>
```

index.php

```
<?
require_once("../resources/class.Joke.php");
require_once("../resources/class.SQLDate.php");
require_once("../resources/class.JokeHTML.php");

$today= new SQLDate();
$today= new SQLDate();
$today= new Joke($today->get_SQLDate());
$content= new JokeHTML("BadJokeCity.com", "../resources/bluestyles.css");
$content= new JokeHTML("BadJokeCity.com", "The first place to look for bad jokes",
"Today's Joke");
$row_0 = array("Main page", "index.php");
$row_1 = array("FAQ", "FAQ.php");
$row_2 = array("User Login", "../user/login.php");
$arrayStructure=array($row_0,$row_1,$row_2);

$content-> buildLeftSideBar($arrayStructure);

$content-> buildMain($today->get_JokeTitle(),$today->get_jokeBody());

$content-> buildFooter($arrayStructure);

print $content->printHTML();
?>
```

index.php

```
<?
require_once("../resources/class.Joke.php");
require_once("../resources/class.SQLDate.php");
require_once("../resources/class.JokeHTML.php");

$today= new SQLDate();
$todayJoke=new Joke($today->get_SQLDate());
$content= new JokeHTML("BadJokeCity.com", "../resources/bluestyles.css");
$content->buildBanner("BadJokeCity.com","The first place to look for bad jokes",
    "Todays Bad Joke");

$row_0 = array("Main page", "index.php");
$row_1 = array("FAQ", "FAQ.php");
$row_2 = array("User Login", "../user/login.php");
$arrayStructure=array($row_0,$row_1,$row_2);

$content-> buildLeftSideBar($arrayStructure);

$content-> buildMain($todayJoke->get_JokeTitle(),$todayJoke->get_jokeBody());

$content-> buildFooter($arrayStructure);

print $content->printHTML();
?>
```

Add the banner content

index.php

```
<?
require_once("../resources/class.Joke.php");
require_once("../resources/class.SQLDate.php");
require_once("../resources/class.JokeHTML.php");

$today= new SQLDate();
$todayJoke=new Joke($today->get_SQLDate());
$content= new JokeHTML("BadJokeCity.com", "../resources/bluestyles.css");
$content->buildBanner("BadJokeCity.com","The first place to look for bad jokes",
    "Todays Bad Joke");

$row_0 = array("Main page", "index.php");
$row_1 = array("FAQ", "FAQ.php");
$row_2 = array("User Login", "../user/login.php");
$arrayStructure=array($row_0,$row_1,$row_2);

$content-> buildLeftSideBar($arrayStructure);

$content-> buildMain($todayJoke->get_JokeTitle(),$todayJoke->get_jokeBody());

$content-> buildFooter($arrayStructure);

print $content->printHTML();
?>
```

Build the menu array

index.php

```
<?
require_once("../resources/class.Joke.php");
require_once("../resources/class.SQLDate.php");
require_once("../resources/class.JokeHTML.php");

$today= new SQLDate();
$todayJoke=new Joke($today->get_SQLDate());
$content= new JokeHTML("BadJokeCity.com", "../resources/bluestyles.css");
$content->buildBanner("BadJokeCity.com","The first place to look for bad jokes",
    "Todays Bad Joke");

$row_0 = array("Main page", "index.php");
$row_1 = array("FAQ", "FAQ.php");
$row_2 = array("User Login", "../user/login.php");
$arrayStructure=array($row_0,$row_1,$row_2);

$content-> buildLeftSideBar($arrayStructure);

$content-> buildMain($todayJoke->get_JokeTitle(),$todayJoke->get_jokeBody());

$content-> buildFooter($arrayStructure);

print $content->printHTML();
?>
```

Build the leftsidebar
using the array

index.php

```
<?
require_once("../resources/class.Joke.php");
require_once("../resources/class.SQLDate.php");
require_once("../resources/class.JokeHTML.php");

$today= new SQLDate();
$todayJoke=new Joke($today->get_SQLDate());
$content= new JokeHTML("BadJokeCity.com", "../resources/bluestyles.css");
$content->buildBanner("BadJokeCity.com","The first place to look for bad jokes",
    "Todays Bad Joke");

$row_0 = array("Main page", "index.php");
$row_1 = array("FAQ", "FAQ.php");
$row_2 = array("User Login", "../user/login.php");
$arrayStructure=array($row_0,$row_1,$row_2);

$content-> buildLeftSideBar($arrayStructure);

$content-> buildMain($todayJoke->get_JokeTitle(),$todayJoke->get_jokeBody());

$content-> buildFooter($arrayStructure);

print $content->printHTML();
?>
```

Build the main content using the \$todayJoke object

index.php

```
<?
require_once("../resources/class.Joke.php");
require_once("../resources/class.SQLDate.php");
require_once("../resources/class.JokeHTML.php");

$today= new SQLDate();
$todayJoke=new Joke($today->get_SQLDate());
$content= new JokeHTML("BadJokeCity.com", "../resources/bluestyles.css");
$content->buildBanner("BadJokeCity.com","The first place to look for bad jokes",
    "Todays Bad Joke");

$row_0 = array("Main page", "index.php");
$row_1 = array("FAQ", "FAQ.php");
$row_2 = array("User Login", "../user/login.php");
$arrayStructure=array($row_0,$row_1,$row_2);

$content-> buildLeftSideBar($arrayStructure);

$content-> buildMain($todayJoke->get_JokeTitle(),$todayJoke->get_jokeBody());

$content-> buildFooter($arrayStructure);

print $content->printHTML();
?>
```

Build the Footer content

index.php

```
<?
require_once("../resources/class.Joke.php");
require_once("../resources/class.SQLDate.php");
require_once("../resources/class.JokeHTML.php");

$today= new SQLDate();
$todayJoke=new Joke($today->get_SQLDate());
$content= new JokeHTML("BadJokeCity.com", "../resources/bluestyles.css");
$content->buildBanner("BadJokeCity.com","The first place to look for bad jokes",
    "Todays Bad Joke");

$row_0 = array("Main page", "index.php");
$row_1 = array("FAQ", "FAQ.php");
$row_2 = array("User Login", "../user/login.php");
$arrayStructure=array($row_0,$row_1,$row_2);

$content-> buildLeftSideBar($arrayStructure);

$content-> buildMain($todayJoke->get_JokeTitle(),$todayJoke->get_jokeBody());

$content-> buildFooter($arrayStructure);

print $content->printHTML();
?>
```

Print the HTML out using the printHTML method

Demo

```
<?
require_once("../resources/class.Joke.php");
require_once("../resources/class.SQLDate.php");
require_once("../resources/class.JokeHTML.php");

$today= new SQLDate();
$todayJoke=new Joke($today->get_SQLDate());
$content= new JokeHTML("BadJokeCity.com", "../resources/bluestyles.css");
$content->buildBanner("BadJokeCity.com","The first place to look for bad jokes",
    "Todays Bad Joke");

$row_0 = array("Main page", "index.php");
$row_1 = array("FAQ", "FAQ.php");
$row_2 = array("User Login", "../user/login.php");
$arrayStructure=array($row_0,$row_1,$row_2);

$content-> buildLeftSideBar($arrayStructure);

$content-> buildMain($todayJoke->get_JokeTitle(),$todayJoke->get_jokeBody());

$content-> buildFooter($arrayStructure);

print $content->printHTML();
?>
```

Professional PHP

Model

User defined
classes

View

Smarty class based
templates

Controller

Class style