

02 (14.02.22)

<https://s3-us-west-2.amazonaws.com/secure.notion-static.com/890a90ee-bf58-4384-9c19-e8123bae7971/02.pdf>

Подходы к разработке web-приложений

1. Основанные на программировании или скриптах:
внешние программы, расширения web-сервера (плагины к nginx, lua скрипты)
2. Основанные на использовании шаблонов web-страниц, включающих вставки кода скриптов и спец. серверных тегов (.php, .pl)
3. Объектные среды (web-frameworks)

Программные подходы

web-приложением является внешняя программа, составленная на некотором универсальном языке программирования высокого уровня.

Проблемы:

1. Ориентация на написание кода: разметка HTML встраивается в логику работы программы при помощи операторов вывода
2. Ограниченные возможности web-дизайнеров, требуется постоянное вмешательство программиста

динамическое формирование страницы - работа внешней программы.

первая технология - **CGI (Common Gateway Interface)** - набор правил, которым должна следовать программа, для выполнения на разных HTTP-серверах и ОС.

при поступлении в web сервер HTTP-запроса, который включает ссылку не на статическую страницу, а на CGI программу, создается новый процесс, в котором запускается требуемая прикладная программа.

входные данные передаются в качестве параметров функции, или через набор фиксированных переменных среды. выходные данные (HTML-страница) возвращаются с помощью STDOUT.

Проблемы:

1. Производительность (для каждого HTTP-запроса web-сервер запускает новый процесс)
2. Составление и отладка CGI-программ требует достаточного опыта программирования
3. Программный код и код разметки полностью перемешаны

Fast CGI - попытка объединить переносимость CGI-приложений с эффективностью их работы (позволяет не закрывать процессы, связанные со скриптами, а использовать их для обработки новых запросов программ)

ISAPI - библиотека функций, с помощью которой можно создавать web-приложения в виде DLL-модулей (динамически подключаемых библиотек), формирующих HTML-страницы.

Java Servlet API - связывает web-сервер с виртуальной машиной Java Virtual Machine (JVM) (переносима между разными web-серверами, ОС, комп. платформами).

Подходы на основе шаблонов

HTML-файлы с дополнительными “тегами” (серверными), которые задают методы включения динамически формируемого контента.

файл шаблона - HTML-код, который описывает общую структуру страницы, и дополнительные **серверные теги**.

Server-Side Includes (SSI), Cold Fusion, Active Server Pages (ASP), Java Server Pages (JSP).

Вставка в шаблон **SSI** инструкций выполняется с помощью следующего формата:

<!-- instruction attribute='value' attr2='Value2' -->

```
<html>
<head><title>hello</title></head>
<body>
  <!-- #exec cgi http://mysite.org/cgi-bin/zip-ssi.cgi -->
</body>
</html>
```

Cold Fusion:

```
<cfoutput>
  <h1>#rsPage.FLD_PAGETITLE#</h1>
  #rsPage.FLD_PAGECONTENT#
</cfoutput>
```

PHP Hypertext Preprocessor:

```
<b>
<?php if ($xyz >= 3 ) { print $myHeading; }
  else {
?>
  DEFAULT HEADING
<?php } ?>
</b>
```

Active Server Pages:

```
<% @LANGUAGE = VBScript %>
<% Set conn = Server.CreateObject("ADODB.Connection")
conn.open("Data Source=mydata; User ID=myname;Password=*****")
Set results = Server.CreateObject("ADODB.RecordSet")
Set results.ActiveConnection = conn
query = "SELECT X, Y, Z FROM TABLE1 WHERE X > 23"
results.Open query %>
<html> <head><title>Active Server Page Example</title></head>
<body id="c09-body-0003" bgcolor="#ffffff" >
<table>
  <tr>
    <td align="center">X</td> <td align="center">Y</td> <td align="center">Z</td>
  </tr>
<%
  While Not results.EOF
    Response.Write "<tr>" Response.Write "<td>" &results(" X") & " </td>"
    Response.Write "<td>" &results(" Y") & " </td>"
    Response.Write "<td>" &results(" Z") & " </td>" Response.Write "</tr>"
  Wend
%>
</table>
</body>
</html>
```

Java Server Pages:

```
<%@ page import="java.io.*" %>
<%! private CustomObject myObject; %>
<h1>My Heading</h1>
<%
    for(int i = 0; i < myObject.getCount(); i++) { %>
        <p>Item #<%= i %> is ' <%= myObject.getItem(i) %>' . </p>
    } %>
```

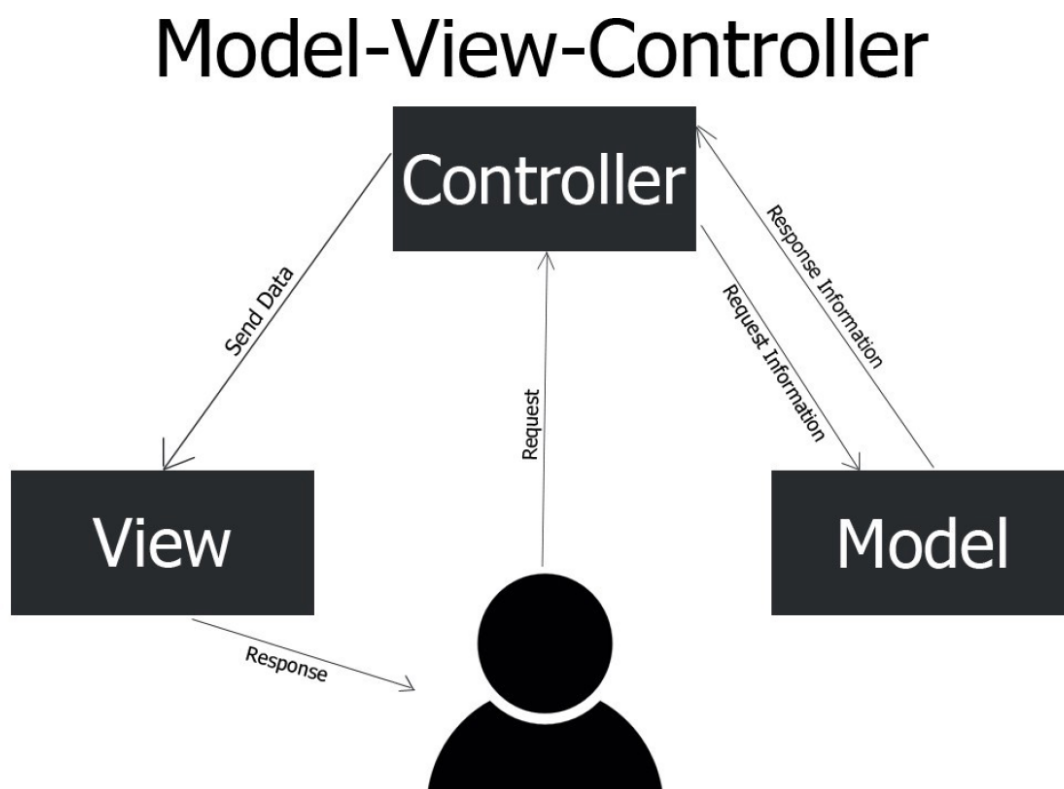
Подходы на основе объектных сред

большее сходство с разработкой ООП ПО

поддерживается принцип отделения содержания от представления. Модули контента отделены от модулей отображения данного содержания.

Подходы:

- Набор специальных web-страниц, связанных с описаниями классов, объекты которых создаются и используются при вызове (ASP.Net Web Forms, JSP)
- Набор классов по шаблону MVC (Model - View - Controller) (Spring, ASP.Net MVC)



Model - бизнес-логика web-приложения (Эти классы отвечают за обработку данных (сущностей), размещение их в БД, чтение из БД, а также за взаимодействие между самими объектами).

View - интерфейс взаимодействия с пользователями (UI) (формируют HTML страницы, показывающие пользователям данные из модели).

Controller - получает данные из запроса к серверу и передают в модель для обработки и сохранения.

Движок Pug / EJS

<https://metanit.com/web/nodejs/4.1.php>

Template Engines

ex <https://expressjs.com/en/resources/template-engines.html>

Express

Паттерн MVC

<https://metanit.com/web/nodejs/7.1.php>