

возможность конфигурировать работу сервера в отдельных папках, не предоставляя доступа к главному конфигурационному файлу. В каталоге “pages” находятся подкаталоги с названием страниц, которые есть на сайте. В каждом из таких подкаталогов содержится PHP файл, который генерирует контент и подключает соответствующий вид из папки “view”. В каталоге view содержатся php файлы, содержащие html разметку с php вставками для вывода содержимого определённой переменной. В этой папке находятся css стили и javascript файлы. В каталоге functions содержатся файлы с php функциями, которые используются в других частях приложения. Файл “connection.php” содержит константы для конфигурации подключения к базе данных. Файл “route.php” играет роль маршрутизатора. Он разбирает URL запрос и подключает требуемый php файл из папки pages.

**Е.В. Мельников** (УО «ГГУ имени Ф. Скорины», Гомель)  
Науч. рук. **В.А. Дробышевский**, старший преподаватель

## **ПРОГРАММНАЯ ЛОГИКА ВЕБ-САЙТА ДЛЯ ГОМЕЛЬСКОГО ГОРОДСКОГО НАУЧНО-ПРОСВЕТИТЕЛЬСКОГО ОБЩЕСТВЕННОГО ОБЪЕДИНЕНИЯ «ОРАКУЛ»**

При разработке веб-сайта было решено использовать процедурный стиль вместо ОПП. Поскольку данный проект имеет довольно простую структуру, то было решено пренебречь такими особенностями ООП как удобная командная разработка и масштабирование (что необходимо крупным проектам) в пользу производительности. Однако это не означает, что весь код собран в одной куче. Принципы MVC соблюдены. Далее описывается, как работает данный проект. Сначала все запросы поступают на файл index.php, который находится в корне проекта. В нём определяются главные константы и route.php:

```
<?php
$root = getcwd() . '/';
define('APP', $root . 'application/');
define('CORE', $root . 'core/');
define('PAGES', APP . 'pages');
define('VIEWS', APP . 'views/');
define('VIEWS_REL', '/application/views/');
define('FUNCTIONS', APP . 'functions/');
include CORE.'route.php';
```

php?>

Далее управление переходит в route.php. Там подключаются файл functions.php, который содержит необходимые для работы приложения функции. Затем в этом файле разбирается URL на составные части. Затем проверяется, есть ли папка с названием, которое указано в URL

и конструируется строка, которая содержит путь к файлу `index.php` для этой папки. Например, `...pages/home/index.php`.

```
include CORE.'functions/main_functions.php';
$path = rtrim(parse_url($_SERVER['REQUEST_URI'], PHP_URL_PATH), '/');
$page_path = PAGES . $path;
if(is_dir($page_path)) { $page_path .= '/index'; }
$page_path .= '.php';
if(is_readable($page_path)) {include $page_path; }
else { header('HTTP/1.1 404 Not Found');
      header("Status: 404 Not Found");
      view('error404');
      exit(); }
```

В файле `index.php` конкретной страницы формируются данные, которые нужны для вывода на страницу. Данные берутся из базы данных. Затем вызывается функция `view`, в которую в качестве параметров передается название файла разметки и данные как ассоциативный массив. Причем в качестве данных также передается разметка страницы.

```
$data['content'] = view('contact_view');
$data['somedata'] = get_some_data();
echo view('layout_view', $data);
```

Функция `view` извлекает данные из ассоциативного массива как отдельные самостоятельные переменные. Затем включает буферизацию. Конструирует строку-путь для определенного `view`, используя для этого константу из самого главного `index.php` и переданную в эту функцию как аргумент строку, содержащую название файла разметки. Затем подключает этот файл. После чего очищает буфер.

```
function view($view, $data = null){
    if($data !== null){ extract($data); }
    ob_start();
    include_once VIEWS . $view . '.php';
    return ob_get_clean(); }
```

Несмотря на то, что в коде не используется ООП, на удобство разработки это никак не влияет. Всё разбито на части и находится в отдельных файлах, которые соединяются друг с другом во время выполнения.

**К.А. Мельников** (УО «ГГУ имени Ф. Скорины», Гомель)

Науч. рук. **М.С. Данильченко**, старший преподаватель

## **АВТОМАТИЗАЦИЯ МОНИТОРИНГА СОСТОЯНИЯ ВИРТУАЛЬНОЙ МАШИНЫ JAVA**

В производственных условиях важно, чтобы виртуальная машина Java поддерживалась в работоспособном состоянии, ведь она используется для запуска созданных приложений и лучшим вариантом контроля этого будет использование приложения, которое бы самостоятельно считывало нужные показатели работоспособности и делало их анализ.