

Veebiprogrammeerimine

Jelena Kuzmina

ООП

- Пример создания класса

- Свойства

- Конструктор

- Get-методы

```
<?php
class Categoria
{
    // property declaration
    private $id ;
    private $nimetus;
    private $kirjeldus;

    //construct
    function __construct($id,$nimi,$tekst)
    {
        $this->id=$id;
        $this->nimetus=$nimi;
        $this->kirjeldus=$tekst;
    }

    // method declaration
    public function getId() {
        return $this->id;
    }

    public function getNimetus() {
        return $this->nimetus;
    }

    public function getKirjeldus() {
        return $this->kirjeldus;
    }

    public function displayId() {
        echo $this->id;
    }

    public function displayNimetus() {
        echo $this->nimetus;
    }

    public function displayKirjeldus() {
        echo $this->kirjeldus;
    }
}
```

```

<?php
include_once 'categoria.php';

class ReadFromFile
{
    public static function readtxt()
    {
        $allcat=array();
        $file=file('andmed.txt');
        foreach($file as $line)
        {
            $info=explode('|',$line);
            $allcat[]=new Categoria($info[0],$info[1], $info[2]);
        }
        return $allcat;
    }

    public static function createLinks()
    {
        //метод создают ссылки с названиями категорий

        $cats=self::readtxt(); // self-для вызова статических данных в классе
        foreach ($cats as $c)
        {
            echo '<a href="index.php?id='.$c->getId().'">'.$c->getNimetus().'</a> <br>';
        }
    }

    public static function InfoByID($id)
    {
        $cats=self::readtxt(); // self-для вызова статических данных в классе
        foreach ($cats as $c)
        {
            if ($c->getId()==$id )
            {
                echo '<p >'.$c->getKirjeldus().'</p> ';
            }
        }
    }
}
?>

```

Класс для работы с категориями

- Статические методы, вызываются без создания объекта

Index.php

- Вызов статических методов

```
<?php
include_once 'readfromfile.php';

ReadFromFile::createLinks();

if (isset($_GET['id']))
{
    ReadFromFile::InfoByID($_GET['id']);
}

?>
```



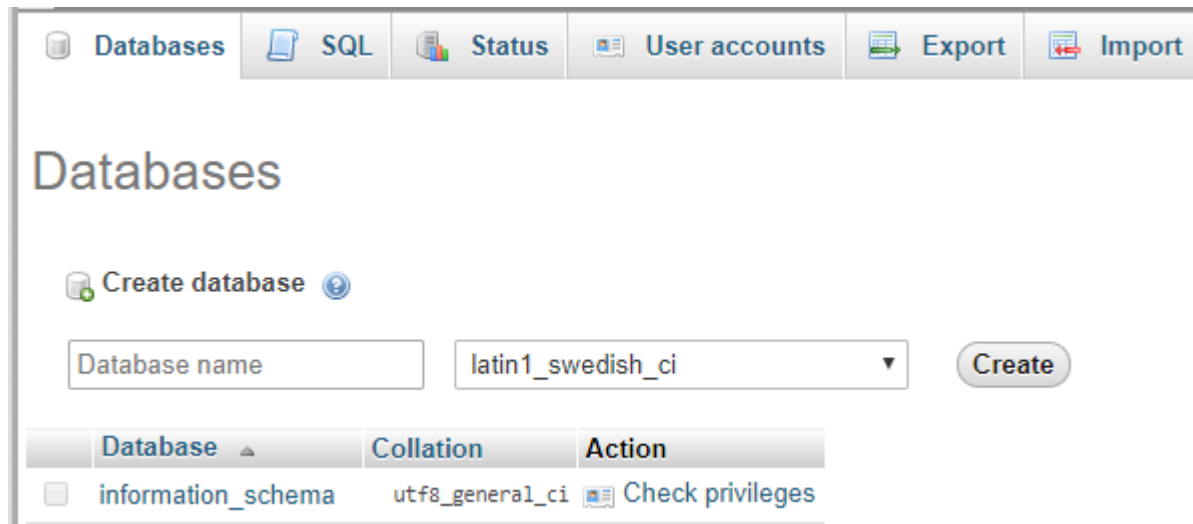
свободная реляционная система управления базами данных.

Обычно MySQL используется в качестве сервера, к которому обращаются локальные или удалённые клиенты.

Входит в состав серверов WAMP, AppServ, LAMP и в портативные сборки серверов Денвер, **XAMPP**, VertrigoServ.


MySQL в XAMPP

- <http://localhost/phpmyadmin/>
- Создание базы



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

MySQL- Создание таблицы

 **Create table**

Name:

Number of columns:

Table name: **Add** **column(s)** **Go**

Structure

Name	Type	Length/Values	Default	Collation	Attributes	Null	Index	A..I	Comments	Virtuality	Move column	MIME type
<input type="text"/> <small>Pick from Central Columns</small>	INT	<input type="text"/>	None	<input type="text"/>	<input type="text"/>	<input type="checkbox"/>	---	<input type="checkbox"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="text"/> <small>Pick from Central Columns</small>	INT	<input type="text"/>	None	<input type="text"/>	<input type="text"/>	<input type="checkbox"/>	---	<input type="checkbox"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="text"/> <small>Pick from Central Columns</small>	INT	<input type="text"/>	None	<input type="text"/>	<input type="text"/>	<input type="checkbox"/>	---	<input type="checkbox"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="text"/> <small>Pick from Central Columns</small>	INT	<input type="text"/>	None	<input type="text"/>	<input type="text"/>	<input type="checkbox"/>	---	<input type="checkbox"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

Table comments:

Collation:

Storage Engine: InnoDB

PARTITION definition:

Partition by: ()

Partitions:

Preview SQL **Save**

Типы данных MySQL

Символьные типы

- **CHAR**: представляет строку фиксированной длины.
- **VARCHAR**: представляет строку переменной длины.
- **TINYTEXT**: представляет текст длиной до 255 байт.
- **TEXT**: представляет текст длиной до 65 КБ.
- **MEDIUMTEXT**: представляет текст длиной до 16 МБ
- **LARGETEXT**: представляет текст длиной до 4 ГБ

- **TINYINT**: представляет целые числа от -127 до 128, занимает 1 байт
- **BOOL**: фактически не представляет отдельный тип, а является лишь псевдонимом для типа `TINYINT(1)` и может хранить два значения 0 и 1. Однако данный тип может также в качестве значения принимать встроенные константы **TRUE** (представляет число 1) и **FALSE** (предоставляет число 0).

Также имеет псевдоним **BOOLEAN**.

- **TINYINT UNSIGNED**: представляет целые числа от 0 до 255, занимает 1 байт
 - **SMALLINT**: представляет целые числа от -32768 до 32767, занимает 2 байта
 - **SMALLINT UNSIGNED**: представляет целые числа от 0 до 65535, занимает 2 байта
-
- **MEDIUMINT**: представляет целые числа от -8388608 до 8388607, занимает 3 байта
 - **MEDIUMINT UNSIGNED**: представляет целые числа от 0 до 16777215, занимает 3 байта
 - **INT**: представляет целые числа от -2147483648 до 2147483647, занимает 4 байта
 - **INT UNSIGNED**: представляет целые числа от 0 до 4294967295, занимает 4 байта
 - **BIGINT**: представляет целые числа от -9 223 372 036 854 775 808 до 9 223 372 036 854 775 807, занимает 8 байт
 - **BIGINT UNSIGNED**: представляет целые числа от 0 до 18 446 744 073 709 551 615, занимает 8 байт

- **DECIMAL**: хранит числа с фиксированной точностью. Данный тип также имеет псевдонимы **NUMERIC, DEC, FIXED**.
- **FLOAT**: хранит дробные числа с плавающей точкой одинарной точности от $-3.4028 * 10^{38}$ до $3.4028 * 10^{38}$, занимает 4 байта.
- **DOUBLE**: хранит дробные числа с плавающей точкой двойной точности от $-1.7976 * 10^{308}$ до $1.7976 * 10^{308}$, занимает 8 байт.

Данный тип также имеет псевдонимы **REAL** и **DOUBLE PRECISION**, которые можно использовать вместо DOUBLE.

- **Типы для работы с датой и временем**
- **DATE:** хранит даты с 1 января 1000 года до 31 декабря 9999 года (с "1000-01-01" до "9999-12-31"). По умолчанию для хранения используется формат `yyyy-mm-dd`. Занимает 3 байта.
- **TIME:** хранит время от -838:59:59 до 838:59:59. По умолчанию для хранения времени применяется формат `"hh:mm:ss"`. Занимает 3 байта.
- **DATETIME:** объединяет время и дату, диапазон дат и времени - с 1 января 1000 года по 31 декабря 9999 года (с "1000-01-01 00:00:00" до "9999-12-31 23:59:59"). Для хранения по умолчанию используется формат `"yyyy-mm-dd hh:mm:ss"`. Занимает 8 байт
- **TIMESTAMP:** также хранит дату и время, но в другом диапазоне: от "1970-01-01 00:00:01" UTC до "2038-01-19 03:14:07" UTC. Занимает 4 байта
- **YEAR:** хранит год в виде 4 цифр. Диапазон доступных значений от 1901 до 2155. Занимает 1 байт.

Тип Date может принимать даты в различных форматах, однако непосредственно для хранения в самой бд даты приводятся к формату "yyyy-mm-dd".

Некоторые из принимаемых форматов:

- yyyy-mm-dd - 2018-05-25
- yyyy-m-dd - 2018-5-25
- yy-m-dd - 18-05-25

В таком формате двузначные числа от 00 до 69 воспринимаются как даты в диапазоне 2000-2069. А числа от 70 до 99 как диапазон чисел 1970 - 1999.

- yyyymmdd - 20180525
- yyyy.mm.dd - 2018.05.25

Для времени тип Time использует 24-часовой формат. Он может принимать время в различных форматах:

- hh:mi - 3:21 (хранемое значение 03:21:00)
- hh:mi:ss - 19:21:34
- hhmiss - 192134

Примеры значений для типов DATETIME и TIMESTAMP:

- 2018-05-25 19:21:34
- 2018-05-25 (хранемое значение 2018-05-25 00:00:00)

Структура созданной таблицы

Server: 127.0.0.1 » Database: koolitused » Table: kategooria

Browse Structure SQL Search Insert Export Import Privileges Operations

Table structure Relation view

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1	id	bigint(20)	UNSIGNED	No	None		AUTO_INCREMENT	Change Drop More
<input type="checkbox"/>	2	Nimetus	varchar(50)	utf8_bin	No	None			Change Drop More
<input type="checkbox"/>	3	Kirjeldus	varchar(255)	utf8_bin	No	None			Change Drop More

☐ Check all With selected: Browse Change Drop Primary Unique Index Fulltext

Print Propose table structure Track table Move columns Normalize

Add column(s)

Indexes

Action	Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
Edit Drop	PRIMARY	BTREE	Yes	No	id	3	A	No	
Edit Drop	id	BTREE	Yes	No	id	3	A	No	

Create an index on columns

Partitions

Добавление данных

←

Server: 127.0.0.1 » Database: koolitused » Table: kategoria

Browse

Structure

SQL

Search

Insert

Export

Import

Privileges

Operations

Tr

Column	Type	Function	Null	Value
id	bigint(20) unsigned			
Nimetus	varchar(50)			
Kirjeldus	varchar(255)			

Go

☒ Ignore

Column	Type	Function	Null	Value
id	bigint(20) unsigned			
Nimetus	varchar(50)			
Kirjeldus	varchar(255)			

Go

Insert as new row

and then

Go back to previous page

?

Go

Preview SQL

Reset

Continue insertion with rows

```

include_once 'categoria.php';

class DBMySQL
{
    private static $servername = "localhost";
    private static $username = "root";
    private static $password = "";
    private static $db = "koolitused";

    public static function getAllCategories()
    {
        $allcat=array();

        // Create connection
        $conn = mysqli_connect(self::$servername, self::$username, self::$password, self::$db);

        // Check connection
        if (!$conn) {
            die("Connection failed: " . mysqli_connect_error());
        }
        else
        {
            if ($result = $conn->query("SELECT id,Nimetus,Kirjeldus FROM kategooria"))
            {
                while ($obj = $result->fetch_object()) {
                    $allcat[]=new Categoria($obj->id,$obj->Nimetus, $obj->Kirjeldus);
                }

                /* free result set */
                $result->close();
            }
        }
        $conn->close();

        return $allcat;
    }
}

```

- Получение данных из базы

```
public static function getCategoryById($id)
{
    // Create connection
    $conn = mysqli_connect(self::$servername, self::$username, self::$password, self::$db);

    // Check connection
    if (!$conn) {
        die("Connection failed: " . mysqli_connect_error());
    }
    else
    {
        if ($result = $conn->query("SELECT id,Nimetus,Kirjeldus FROM kategoria WHERE id="(string)$id))
        {
            while ($obj = $result->fetch_object()) {
                $cat=new Categoria($obj->id,$obj->Nimetus, $obj->Kirjeldus);
            }

            /* free result set */
            $result->close();
        }
    }
    $conn->close();

    return $cat;
}
```


Вывод информации на страницу

```
echo "Данные из MySQL <br>";

$cats=DBMySQL::getAllCategories();
foreach ($cats as $c)
{
    echo '<a href="index.php?iddb='.$c->getId().'">'.$c->getNimetus().'</a> <br>';
}

if (isset($_GET['iddb']))
{
    $c=DBMySQL::getCategoriaById($_GET['iddb']);
    echo '<p >'.$c->getKirjeldus().'</p> ';
}
```

Новостной портал

проект

Пример (Delfi.ee)

Стартовая страница

[Kõik uudised](#) / [Eesti](#) / [Arvamus](#) / [Arvamuskonkurss](#) / [Maailm](#) / [Krimi](#) / [Liiklus](#) / [Delfi Lood](#) / [Delfi Tasku](#) / [Arhiiv](#) / [Delfi kaart](#) / [Kliima](#) / [METS](#)



Kohus lõpetab valeütluste andmises süüdistatud Mary Krossi kriminaalasja avaliku huvi puudumisel (270)

FOTOD KOHTUST | Harju maakohtus algas protsess Mary Krossi üle, keda süüdistatakse valeütluste andmises (303)



ILMATARKADE SUUR TALVEENNUSTUS: Kas meid ootab ees lõputu sügis? (62)

FOTOD | Lumelörtsi ootuses: meenutus ootamatust lumetormist seitse aastat tagasi (9)

VIDEO ja FOTOD | Huvasti, sügisilmad: talv on tulekul! Tartus sajab lund (31)

Все новости

[Kõik uudised](#) / [Eesti](#) / [Arvamus](#) / [Arvamuskonkurss](#) / [Maailm](#) / [Krimi](#) / [Liiklus](#) / [Delfi Lood](#) / [Delfi Tasku](#)

Viimased uudised



Varalahkunud George Michaeli elukaaslase suur paljastus: laulja oli HIV positiivne!

31.10.2019 10:24 Kroonika



Inbank on tänavu teeninud 6,57 miljoni eurot kasumit

31.10.2019 10:21 Ärileht



Rallimänedžer: Toyota võimalused Ogier endale saada on 50-50

31.10.2019 10:20 Sport



Kas meie toit on päriselt nii naturaalne, kui me arvame?

31.10.2019 10:20 Forte

Поиск

ALATES:

KUNI:

31.10.2018

31.10.2019


VALI PERIOOD

KÕIK KANALID

KÕIK KATEGOORIAD

OTSI


31.10.2018 - 31.10.2019



Parim maamajandusõppur: riik peaks toetama peretalusid

31.10.2019 00:01


Maaelht



Tutvu kandidaatidega! Kellest saab 2019. aasta parim taimekasvataja?

29.10.2019 07:10


Maaelht



Lase hea maitsta, sest on november!

28.10.2019 15:42

Reisijuht



Публикация

Varasem otsus vaadati ümber: Postimees Grupp hakkab siiski Vene telekanalite reklaamimüügiga tegelema (27)



Riho Nagel
reporter-toimetaja



Postimees

Foto: Madis Veltman

Postimees Grupi juhatuse esimehe kohusetäitja Andrus Raudsalu kinnitusel hakkab ettevõtte siiski Vene telekanalite reklaamimüügiga tegelema, kuigi augusti keskel loobuti sellest toimetuse vastuseisu tõttu.

Комментари

Sinu nimi



Sinu kommentaar



Vanemad



Anonüümne (23)

Registreeritud (5)



mmm

31.10.2019 09:49:07

Sobimatu



Raha paneb rattad käima. Kui ikka ollakse 9 miljoniga miinuses, siis näidatakse kas või Põhja-Korea reklaami.

1 шаг

- Создать свой дизайн сайта
- Продумать структуру базы данных
- Создать базу данных в MySQL
- Заполнить базу «нормальными» данными

Database.php класс для работы с бд

```
<?php

class database {

    private $conn;
    private $host;
    private $user;
    private $password;
    private $baseName;

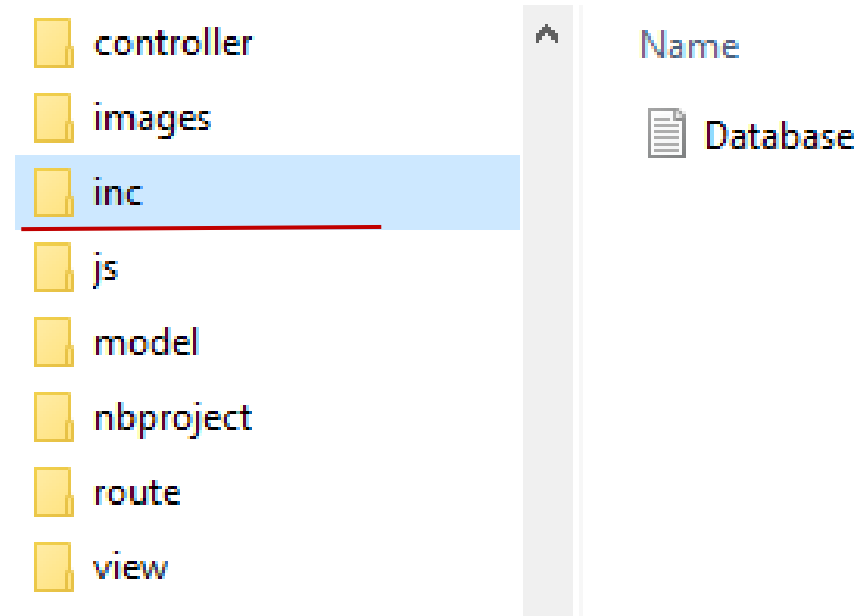
    function __construct() {

        $this->host = 'localhost';
        $this->user = 'root';
        $this->password = '';
        $this->baseName = 'newsportal';
        $this->connect();

    }

    function __destruct() {
        $this->disconnect();
    }

}
```



Соединение с базой

```
function connect() {  
    try {  
        $this->conn = new PDO(  
            'mysql:host=' . $this->host . '  
            . ';dbname=' . $this->baseName . '  
            $this->user,  
            $this->password,  
            array(PDO::MYSQL_ATTR_INIT_COMMAND => 'SET NAMES utf8'));  
    } catch (Exception $e) {  
        die('Connection failed : ' . $e->getMessage());  
    }  
  
    return $this->conn;  
}  
  
function disconnect() {  
    if ($this->conn) {  
        $this->conn = null;  
    }  
}
```

Запросы для SELECT

```
function getOne($query) {  
    $stmt = $this->conn->prepare($query);  
    $stmt->execute();  
    $stmt->setFetchMode(PDO::FETCH_ASSOC);  
    $response = $stmt->fetch();  
    return $response;  
}
```

```
function getAll($query) {  
    $stmt = $this->conn->prepare($query);  
    $stmt->execute();  
    $stmt->setFetchMode(PDO::FETCH_ASSOC);  
    $response = $stmt->fetchAll();  
    return $response;  
}
```

Запрос на изменения и получения последнего ид

```
function executeRun($query) {  
    $response = $this->conn->exec($query);  
    return $response;  
}  
  
function getLastId() {  
    $lastId = $this->conn->lastInsertId();  
    return $lastId;  
}  
-}  
- ?>
```

Index.php

```
<?php
require 'model/Category.php'; // подключаем нужные модели

include_once 'inc/Database.php'; // класс для работы с базой

include_once 'controller/Controller.php'; // контроллер
include_once 'route/routing.php'; // маршрутизатор

echo $response;
?>
```

Результат маршрутизации


MVC


Model-View-Controller (MVC), «Модель-Представление-Контроллер», «Модель-Вид-Контроллер») — схема разделения данных приложения, пользовательского интерфейса и управляющей логики на три отдельных компонента: модель, представление и контроллер — таким образом, что модификация каждого компонента может осуществляться независимо

- **Модель** (*Model*) предоставляет данные и реагирует на команды контроллера, изменяя своё состояние.
- **Представление** (*View*) отвечает за отображение данных модели пользователю, реагируя на изменения модели.
- **Контроллер** (*Controller*) интерпретирует действия пользователя, оповещая модель о необходимости изменений.


MVC

- Создать
 - Модели
 - Контроллеры
 - Вид

 controller

 model

 view

 index.php

Модель категории

model/category.php

```
<?php
class Category{

    public static function getAllCategory() {
        $query = "SELECT * FROM category" ;
        $db = new Database();
        $arr = $db->getAll($query);
        return $arr;
    }

    // далее добавляются так же все методы по работе с категориями
}
```


Представление категории

view/category.php

```
<?php  
  
echo "<li class='submenuunit'><a href='all'>ALL</a></li><br>";  
foreach($arr as $value){  
    echo "<li class='submenuunit'><a href='category?id=".$value['id']."'>".$value['name'].'</a></li><br>";  
}  
  
?>
```

Контроллер

controller/controller.php

```
<?php

class Controller {

    public static function StartSite() {
        include_once 'view/start.php';
    }

    public static function AllCategory() {
        $arr = Category::getAllCategory();
        include_once 'view/category.php';
    }

    public static function error404() {
        include_once 'view/error404.php';
    }
}
```

Layout.php шаблон страницы

```
<!DOCTYPE html>
<html>
  <head>
    <title> NEWSPORTAL</title>
    <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css" >
    <link rel="stylesheet" type="text/css" href="style.css">
    <link href="https://fonts.googleapis.com/css?family=Noto+Serif" rel="stylesheet">
    <meta charset="utf-8">
  </head>
  <body>
    <nav class="one">
      <ul class="topmenu">
        <li><a href="#">Категории<i class="fa fa-angle-down"></i></a>
          <ul class="submenu">
            <?php
              Controller::AllCategory();
            >
          </ul>
        </li>
        <li><a href="iwww">Инфо</a></li>
        <li><a href="index">Главная</a></li>
      </ul>
    </nav>
    <section>
      <div class = ''>
        <?php
          if(isset($content)){
            echo $content;
          }
          else{
            echo '<h1>Content is gone!</h1>';
          }
        >
      </div>
    </section>
  </body>
</html>
```

Вызов контроллера, для вывода всех категорий

Выводит контент страницы

View/start.php

```
<?php
ob_start();
?>
<h1>Start page</h1>
<br>
<?php
$content = ob_get_clean();
include_once 'view/layout.php';
?>
```

Вид стартовой страницы

Собирает контент страницы

- **Ob_start()**- Включение буферизации вывода
- Содержимое этого внутреннего буфера может быть скопировано в строковую переменную, используя **ob_get_contents()**.
- Для вывода содержимого внутреннего буфера следует использовать **ob_end_flush()**.
- В качестве альтернативы можно использовать **ob_end_clean()** для очистки содержимого буфера.

Routing.php

```
<?php
    $host = explode('?', $_SERVER['REQUEST_URI'])[0];
    $num = substr_count($host, '/');
    $path = explode('/', $host)[$num];

    if($path == '' OR $path == 'index' OR $path == 'index.php'){
        $response = Controller::StartSite();
    }

    else{
        $response = Controller::error404();
    }

?>
```

- **explode** — Разбивает строку с помощью разделителя
- **substr_count** — Возвращает число вхождений подстроки

Error404.php

```
<?php
ob_start();
?>
Error 404
<?php
$content = ob_get_clean();
include_once 'view/layout.php';
?>
```


НОВОСТИ

- Создать модель с функциями:
 - AllNews
 - Last10News
 - NewsByCategoryID
 - NewsByID

Создать представления для вывода новостей (Класс со статическими методами)

Изменить View/start.php и создать необходимые дополнительные страницы для просмотра всех новостей , новостей по категориям, просмотр выбранной новости

Дополнить Routing.php