**Тестовое задание на вакансию PHP-Программист.**

Здравствуйте! Высылаю вам тестовое задание на должность PHP-программист.

Плановый срок решения задания: 3-4 дня. Сообщите, пожалуйста, когда приступите к тестовому заданию, на почту hr@astrio.ru.

Если при выполнении задания у вас возникнут вопросы, обращайтесь на почту hr@astrio.ru.

**1.** Дан массив "категории". Каждая категория имеет следующие параметры:

"id" — уникальный числовой идентификатор категорий

"title" — название категории

"children" - дочерние категории (массив из категорий)

Вложенность категории неограниченна (дочерние категории могу иметь свои вложенные категории и т.д.)

Пример массива :

$categories = array(

array(

"id" => 1,

"title" => "Обувь",

'children' => array(

array(

'id' => 2,

'title' => 'Ботинки',

'children' => array(

array('id' => 3, 'title' => 'Кожа'),

array('id' => 4, 'title' => 'Текстиль'),

),

),

array('id' => 5, 'title' => 'Кроссовки',),

)

),

array(

"id" => 6,

"title" => "Спорт",

'children' => array(

array(

'id' => 7,

'title' => 'Мячи'

)

)

),

);

Необходимо написать функцию searchCategory($categories, $id), которая по идентификатору категории возвращает название категории.

**2**. Даны три таблицы:

таблица `worker` (работник) с данными — id (id работника), first\_name (имя), last\_name (фамилия)

таблица `child` (ребенок) с данными — worker\_id (id работника), name (имя ребенка)

таблица `car` (машина) с данными — worker\_id (id работника), model (модель машины)

Структура таблиц:

CREATE TABLE `worker` (

`id` int(11) NOT NULL,

`first\_name` varchar(100) NOT NULL,

`last\_name` varchar(100) NOT NULL,

PRIMARY KEY (`id`)

) ENGINE=InnoDB AUTO\_INCREMENT=1 DEFAULT CHARSET=utf8;

CREATE TABLE `car` (

`user\_id` int(11) NOT NULL,

`model` varchar(100) DEFAULT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

CREATE TABLE `child` (

`user\_id` int(11) NOT NULL,

`name` varchar(100) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

Необходимо написать один SQL запрос который возвращает: имена и фамилии всех работников, список их детей через запятую и марку машины. Выбрать нужно только тех работников, у которых есть или была машина (если машина была и потом её не стало, то поле model становится **null**).

**3.** Написать функцию, которая на входе принимает массив из открывающихся или закрывающихся тегов (например “<a>”, “</td>” ), а возвращает результат проверки корректности: т.е. является ли принятая функцией последовательность тегов структурой корректного HTML документа. Например, последовательность “<a>”, “<div>”, “</div>”, “</a>”, “<span>”, “</span>” - корректная структура, а последовательность “<a>”, “<div>”, ”</a>” - некорректная структура.

Необходимо использовать нативный php без использования библиотек DOMDocument, Simplexml и тд.

**4.** Написать класс-оболочку хранилища «Box».

В хранилище можно установить данные (setData($key, $value)), получить данные (getData($key)), сохранить данные (save()) и загрузить данные (load())

$key — произвольный идентификатор данных

$value — скаляные данные или массив

Хранилище состоит из:

- Интерфейс описывающий методы установки данных, получения данных, сохранения и загрузки

- Абстрактный класс AbstractBox содержащий реализацию необходимых общих методов

- Класс FileBox расширяющий абстрактный класс AbstractBox. При вызове save() сохраняет заданные в класс данные в файл. При вызове load() достает данные из файла.

- Класс DbBox расширяющий абстрактный класс AbstractBox. При вызове save() сохраняет заданные в класс данные в базу. При вызове load() достает данные из базы.

Функция load не должна ничего возвращать, должна лишь сохранять полученные данные внутри объекта. Для получения данных служит функция getData($key).

При сохранении данных необходимо учитывать ключи новых данных и тех данных что хранятся в базе(файле), если значение с ключом в базе(файле) уже существует, то нужно его значение заменить на новое.

Классы FileBox и DbBox должен быть реализованы таким образом, чтобы нельзя было создать более одного экземпляра каждого из классов.

Общие пожелание по выполнению всех заданий: по 1 и 3 заданию необходимо в отдельном файле создать одну функцию с необходимой логикой, если необходимо какую-либо из функций разбить не несколько, делать все в этом же файле, по 2 заданию, необходимо в отдельном файле написать один sql запрос и больше ничего, по 4 заданию в отдельной папке создать необходимые классы. Никаких index.php, html разметки и тем более стилей добавлять не надо.

Желаем вам удачи в выполнении задания!

Спасибо.