

Тестовое задание

Задание состоит из 4-х задач.

Как выполнять задание (требование для первых двух)

- Каждая задача должна быть оформлена в виде отдельного *.php файла с именем `task$i.php`, где `$i` — номер задачи. Ответы на вопросы оформляются в отдельных *.txt файлах.
Хорошо: `task1.php`, `task7.txt`
Плохо: `task$1.php`, `task8`
- Решение задач должно быть оформлено в виде функции. Имя функции и формат принимаемых и возвращаемых значений будет указано в задаче.
Хорошо: использовать заданные названия
Плохо: путать или изменять названия
- Кроме функции никакого исполняемого кода в файле быть не должно.
Хорошо: в каждом файле только функция
Плохо: в файлах содержится вызов функции или еще что-то
- После исполнения код в `stdout` не должно что-то появиться.
+Хорошо: не используются `echo` / `print` / `print_r` / `var_dump` / `var_export` и т.д.
Плохо: после исполнения файлов появляется что-то в выводе
- Файл должен корректно обрабатываться синтаксическим анализатором PHP
Хорошо: `php task1.php` не выводит никаких ошибок
Плохо: код содержит ошибки PHP, используются переменные без инициализации
- Если в результате ожидаются `TRUE` или `FALSE`, это именно булевы значения, а не строки
- Выполнение заданий проверяется автоматически, поэтому невыполнение условий выполнения задач, приведет к ошибкам выполнения теста
- Написать unit-тест к задаче.

Задачи

1. Структура данных

Дано: выражение вида `[5] * 3 - (4 - 7 * [3-6])`.

Написать функцию `checkBrackets($s)`, которая принимает на вход строку с выражением и возвращает `TRUE` или `FALSE`. Если скобки расставлены правильно — `TRUE`, в противном случае — `FALSE`.

Пример неправильно расставленных скобок: `(5 * 3 [6) - 6]`

2. Задача на логику

Счастливые пропуски OWOX.

Предположим, что номер пропуска состоит из 6 цифр. Пропуск считается счастливым, если сумма первых трех цифр равняется сумме вторых трех цифр.

Написать функцию `luckyTickets($k)`, которая посчитает и вернет возможное количество таких пропусков, где k — число цифр в билете (четное, 2,4,6...). Билет 000000 считается.

Примеры счастливых пропусков: 933591, 030300, 113311.

Обратите внимание: количество цифр в пропуске может быть как больше, так и меньше 6. Гарантируемое условие - количество цифр четное.

3. Задача на ООП

Дано: 2 типа занятий по английскому языку – Speaking и Grammar. У них есть общие свойства – длительность и тариф: фиксированный или почасовой (200грн. за занятие, 100 грн. за час).

Задача: Написать систему подсчета стоимости занятий.

- система должна рассчитывать стоимость массива занятий, учитывая, что каждый элемент может быть экземпляром типа занятие и иметь тот или иной тариф;
- система должна быть расширяема – новые типы занятий и тарифы должны легко вводиться;
- назвать выбранные шаблоны проектирования;
- реализовать код базовых классов такой системы;
- привести пример реализации подсчета и пример того, как систему можно расширить;
- при желании дополнить диаграммой классов.

4. Задача по MySQL

Есть БД со следующей структурой данных:

```
CREATE TABLE `products` (  
  `id` INT(11) NOT NULL AUTO_INCREMENT,  
  `name` VARCHAR(128) NOT NULL,  
  `description` TEXT NOT NULL,  
  `price` decimal(10,2) unsigned DEFAULT NULL,  
  PRIMARY KEY (`id`)  
) ENGINE = MyISAM;
```



```
CREATE TABLE `clients` (  
    `id` INT(11) NOT NULL AUTO_INCREMENT,  
    `name` VARCHAR(100) NOT NULL,  
    `email` varchar(64) NOT NULL,  
    PRIMARY KEY (`id`)  
) ENGINE = MyISAM;
```

```
CREATE TABLE `orders` (  
    `id` smallint( 11 ) NOT NULL AUTO_INCREMENT ,  
    `client_id` INT( 11 ) NOT NULL ,  
    `product_id` INT( 11 ) NOT NULL ,  
    `amount` DECIMAL(12, 2) NOT NULL,  
    `created` datetime NOT NULL,  
    `ip` VARCHAR(15) NOT NULL,  
    `client_phone` INT(10) NOT NULL,  
    PRIMARY KEY (`id`)  
) ENGINE = MyISAM;
```

Необходимо изменить/оптимизировать БД согласно следующим требованиям:

0) Каждая таблица имеет более 1 млн записей.

1) Заказ может иметь в составе несколько разных наименований товаров, каждого наименования также может быть более одного.

2) Создание заказа должно проходить через транзакцию.

3) Иметь быструю выборку заказов, которые содержат конкретные товары или принадлежат конкретному пользователю.

4) Должна соблюдаться третья нормальная форма БД.

После доработки БД, необходимо написать запрос по выборке 10 заказов(id заказа, дата создания, count_product - кол-во товаров в заказе, avg_product_price - средняя цена товара в заказе), у которых более одного товара в заказе, отсортированные по дате создания.