# Тестовое задание. Технический писатель

В качестве тестового задания предлагаем вам описать алгоритм, который реализуется в предложенном методе. Необходимо описать, в каком формате надо подавать входные данные, а также описать словами тот результат, которого добивается данный алгоритм в зависимости от входных данных.

# Решение

## Алгоритм

Алгоритм проверяет соответствие правила заданным значениям, в зависимости от чего возвращает результат: 1 (соответствует) или 0 (не соответствует) и добавляет параметры правила в массив параметров из шаблона маршрута.

В начале определяем результат выполнения алгоритма, присваивая значение 0.

Если класс CDebug существует и доступен, записываем в лог информацию о типе правила.

В зависимости от входных данных алгоритм предоставляет следующие результаты:

a) Если тип – «or», проверяем по отдельности каждое вложенное правило, если хотя бы для одного вложенного правила проверка прошла, возвращаем 1, иначе – 0.

б) Если тип – «and», проверяем по отдельности каждое вложенное правило, если хотя бы для одного вложенного правила проверка не прошла, возвращаем 0, иначе – 1.

в) Если тип – «request\_var» или «request\_uri», то:

   в.1) Если «request\_var», сохраняем в переменную var результат вызова метода \_REQUEST, в который передаем название правила.

   в.2) Если «request\_uri», сохраняем в переменную var результат вызова метода \_SERVER, в который передаем «REQUEST\_URI».

   в.3) Если определено регулярное выражение, выполняем проверку:

       в.3.1) Если var соответствует регулярному выражению, то результат равен 1,

               в.3.1.1) Если определены параметры по умолчанию, передаем их в «routing\_parameters».

             в.3.1.2) Если определен список параметров, то в каждый элемент «routing\_parameters» передаем декодированную строку url.

    в.4) Если регулярное выражение не определено:

        в.4.1) Проверяем, если var равно значению правила или значение правила равно «any» и var не пустое, то результат равен 1,

в.4.2) Если класс CDebug существует и доступен, записываем в лог результат выполнения проверки и информацию о правиле.

г) Если тип принимает любое другое значение:

   г.1) Если класс CDebug существует и доступен, записываем в лог, что такое правило неизвестно.

д) В ином случае выполняем описанный выше алгоритм проверки для каждого вложенного правила: если проверка прошла, возвращаем 1, иначе – 0.

   д.1) Если класс CDebug существует и доступен, записываем в лог результат выполнения проверки.

е) Возвращаем результат – 1 или 0.

## Входные данные

Входные данные – объект rules, имеющий свойства:

* type – тип правила, может принимать значения «or», «and», «request\_var», «request\_uri»,
* name – название правил,
* rules – вложенный список правил,
* regexp – регулярное выражение,
* default\_parameters – параметры по умолчанию,
* parameters – список передаваемых параметров,
* value – значение правила

Входные данные могут подаваться в формате json. Затем json декодируется в object с помощью функции json\_decode и передается в check\_route\_rules.