

# Тестовое задание для РНР разработчика

## Библиотека для создания консольных команд

Необходимо разработать библиотеку, обеспечивающую обработку ввода-вывода (I/O) при работе в консоли с возможностью реализации собственных команд конечным разработчиком, который использует эту библиотеку.

Требуется реализовать обработку входящих аргументов запуска в соответствии со следующим соглашением:

- название команды передается первым аргументом в произвольном формате;
- аргументы запуска передаются в фигурных скобках через запятую в следующем формате:
  - одиночный аргумент: `{arg}`
  - несколько аргументов: `{arg1,arg2,arg3}` ИЛИ `{arg1} {arg2} {arg3}` ИЛИ `{arg1,arg2}{arg3}`
- параметры запуска передаются в квадратных скобках в следующем формате:
  - параметр с одним значением: `[name=value]`
  - параметр с несколькими значениями: `[name={value1,value2,value3}]`

Функциональность библиотеки включает в себя:

- регистрацию необходимых команд в приложении;
- возможность установить название и описание каждой команды;
- обработку ввода пользователя (парсинг аргументов и выявление имени команды, аргументов и параметров);
- выполнение заданной логики с возможностью вывода информации в консоль.

При запуске приложения без указания конкретной команды необходимо выводить список всех зарегистрированных в нём команд и их описания.

При запуске любой из команд с аргументом `{help}` необходимо выводить описание команды.

*Ограничения: для сокращения трудозатрат достаточно обеспечить корректную работу библиотеки на ОС семейства Linux.*

## Приложение на базе разработанной библиотеки

На базе разработанной библиотеки необходимо создать простое консольное приложение для демонстрации его работы.

Требования к приложению:

- единая точка входа для запуска из консоли ОС семейства Linux;
- возможность регистрации неограниченного количества команд в приложении.

Необходимо создать и зарегистрировать в приложении консольную команду, которая на вход принимает неограниченное количество аргументов и параметров и выводит их на экран в читаемом для пользователя виде.

## Пример работы конечной команды

```
$/usr/bin/php app.php command_name {verbose,overwrite} [log_file=app.log] {unlimited}  
[methods={create,update,delete}] [paginate=50] {log}
```

Called command: command\_name

### Arguments:

- verbose
- overwrite
- unlimited
- log

### Options:

- log\_file
  - app.log
- methods
  - create
  - update
  - delete
- paginate
  - 50

## Требования к решению

- решение должно быть выполнено без использования фреймворков и сторонних библиотек, однако допускается использование composer для загрузки PSR-контрактов, фреймворков для Unit-тестирования, а также любых иных целей, не связанных с установкой сторонних библиотек;
- обязательно использование ООП с соблюдением принципов SOLID;
- рекомендуемая версия PHP: 7.4 - 8.1;
- решение необходимо разместить на одном из открытых сервисов для хранения кода (GitHub / Gitlab / BitBucket).

## Критерии оценки

- грамотная и современная архитектура, отделяемость логики библиотеки от реализации приложения и консольных команд;
- аккуратный документированный код, соблюдение PSR стандартов при оформлении кода;
- работоспособность решения;
- будет плюсом: покрытие Unit-тестами, конфигурация для запуска в Docker.