# Привет!

# Для успешного прохождения отбора в А25 выбери любую подходящую задачу и реши ее (можно и несколько 😄). В случае нехватки информации для решения кейса или возникновения уточняющих вопросов – задавай вопросы в чате.

# Мы оцениваем не объем проделанной работы, а глубину проработки и качество представления результата. Например, если ты хорошо проработал и представил интересное решение даже небольшой проблемы – это гораздо ценнее, чем пройтись по вершкам и ни по одной из проблем не прийти к работающему решению. ВАЖНО! Не следует делать больше, чем мы просим в задании. Это усложнит нам проверку.

# Мы рассчитываем, что решение тестового кейса в совокупности займёт у тебя не более восьми часов. Если процесс затянулся – скорее всего ты выбрал слишком сложный путь либо взял излишне широкий пласт проблем. Попробуй сфокусироваться на чём-то одном или посоветуйся с кураторами в чате Telegram.

# 

# Легенда

У межгалактической команды Y по прокату летающих тарелок есть сайт-одностраничник:

Вид: форма-калькулятор для расчета аренды

Стек: PHP + HTML + jQuery

Требования: PHP 7.4

Команда Y растёт и им уже не хватает одного калькулятора. Просят о доработках.

Выбери любую из ниже представленных задач

## Задача 1 | Реализовать Адаптер

**Проблема:** в index.php используется прямое обращение к БД (через класс sdbh)

* это не отвечает требованиям безопасности
* в случае изменения источника данных возникнет необходимость переписывать все подобные обращения к классу sdbh

**Предлагаемое решение:** убрать прямую зависимость между источником данных (БД) и местами применения этих данных

**Что нужно сделать:**

1. Реализовать адаптер между sdbh и логикой приложения ([Документация](https://alishoff.com/blog/325))
2. Применить методы получения данных из нового класса в index.php и App/calculate.php
3. Добавить метод получения тарифной сетки
4. Добавить вывод тарифной сетки в index.php для выбранного Продукта до момента расчета
5. Реализовать обновление тарифов при смене Продукта (обращаться к серверу через ajax за получением новых данных)

## Задача 2 | Прикрутить календарь

**Проблема:** пользователи ошибаются с количеством дней проката

**Предлагаемое решение:** добавить календарь с выбором даты начала аренды и датой окончания

**Что нужно сделать:**

1. Заменить на форме поле ввода для количества дней на поля для ввода дат начала и окончания аренды
2. Добавить [jQuery UI](https://jqueryui.com/datepicker/) / [Air Datepicker](https://air-datepicker.com/) или любой другой js виджет календаря для выбора дат
3. Реализовать метод подсчета количества дней аренды на основе дат (на PHP)
4. Доработать скрипт расчета стоимости проката (calculate.php)

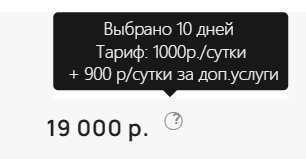
## Задача 3 | Рефакторинг

**Проблема:** скрипт расчета стоимости проката выбивается из общей архитектуры, с ним трудно работать, такой legacy code

**Решение:** рефакторинг кода

**Что нужно сделать:**

1. В calculate.php отрефакторить метод calculate1() по стандартам CleanCode
2. При необходимости, разделить функционал за который отвечает calculate.php (приём данных → обработка → калькуляция → вывод данных) на соответствующие слои в текущей архитектуре проекта
3. Исправить баг: для товара с ID = 4 расчеты неверны
4. Добавить в index.php вывод тарифа, по которому был произведен расчет. Вывод тарифа убрать в tooltip (например, bootstrap) → см. макет



## Задача 4 | Реализовать новую фичу

**Проблема:** отсутствует возможность добавлять новый продукт в БД

**Предлагаемое решение:** добавить форму для добавления данных в БД

**Что нужно сделать:**

1. На странице admin.php сверстать форму для добавления данных по товару *(Доступ к admin.php должен быть только если текущий user→isAdmin. Реализация функционала для User отсутствует, поэтому в рамках данной задачи считаем, что isAdmin всегда true)*
2. Реализовать на backend (App\Application\AdminService.php) методы добавления данных по новому Продукту в БД. Для общения с БД необходимо использовать класс sdbh. Учесть текущую архитектуру проекта.
3. Связать форму и backend через ajax (jQuery)

## Задача 5 | Отправлять заказы по почте

**Проблема:** отсутствует фиксация заказов

**Предлагаемое решение:** отправлять заявки на почту менеджеру

**Что нужно сделать:**

1. Добавить в index.php кнопку “Оставить заявку”, которая должна появляться после расчета итоговой стоимости
2. По нажатию на кнопку отображать модальное окно с полем для ввода номера телефона и кнопкой “Оставить”. Добавить маску ввода для номера телефона
3. По нажатию на кнопку “Оставить” отправляется запрос на сервер с полной информацией о заказе и номер телефона
4. Сервер принимает запрос и отправляет письмо на почту менеджеру. Необходимо учесть текущую архитектуру проекта.

## Задача 6 | Агрегирование данных

**Проблема:** менеджеры не видят данные, которые есть в БД

**Предлагаемое решение:** реализовать дашборд менеджера для вывода статистики

**Что нужно сделать:**

1. На backend реализовать методы получения статистической информации:
   1. Вывести количество Продуктов в базе
   2. Вывод Продуктов, у которых не задано поле TARIFF
   3. Определить самую дорогую и самую дешёвую дополнительную услугу
   4. Вывод Продукта, у которого изменение TARIFF начинается раньше всех

(например, у Продукта 1 в TARIFF задан массив [0 => 1000, 15 => 900], а у Продукта 2 - [0 => 1000, 5 => 900]. Следовательно у Продукта 2 изменение тарифа начинается уже с 5 дня, а у Продукта 1 - только с 15. Значит выводим Продукт 2)

1. Вывести статистические данные в admin.php. *(Доступ к admin.php должен быть только если текущий user→isAdmin. Реализация функционала для User отсутствует, поэтому в рамках данной задачи считаем, что isAdmin всегда true)*
2. Реализовать поиск товара по наименованию:
   1. на admin.php добавить поле ввода для наименования и кнопку для отправки поиска
   2. на backend реализовать метод получения информации (TARIFF и PRICE) о продукте по его наименованию
   3. связку между admin.php и backend реализовать на ajax(jQuery). Переработку TARIFF в удобочитаемый вид реализовать на js

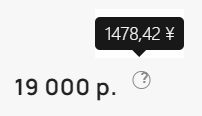
## Задача 7 | Интеграция с API курса валют

**Проблема:** не все клиенты оплачивают услуги в рублях, некоторые - в Юанях, но им приходится дополнительно переходить в конвертер валют, что неудобно

**Предлагаемое решение:** получать текущий курс валют и выводить расчет аренды в Юанях

**Что необходимо сделать:**

1. Воспользоваться любым API (например, https://www.cbr-xml-daily.ru/daily\_json.js) и получать текущий курс валют RUB к CNY
2. Доработать метод расчета стоимости аренды calculate.php и производить расчеты в 2-х валютах
3. Добавить в index.php вывод стоимости в CNY в tooltip (например, bootstrap) → см. макет



## Задача 8 | Прикрутить проект к CMS

1. Необходимо внедрить проект в любую CMS (WP / UMI / другое) так, чтобы можно было использовать административную панель CMS с целью редактирования и просмотра списка Продуктов и дополнительных услуг
2. **Важно!** Учесть правильную кастомизацию и шаблонизацию выбранной CMS, не изменяя её ядро.
3. Опубликовать результат на любом хостинге и прислать ссылку
4. Залить изменения github в формате:
   1. master ветка - чистая CMS
   2. develop ветка - твои изменения

# Архитектура проекта и исходные данные

project.zip - папка проекта с вложениями, где

assets/ - дополнительные файлы для frontend части (css/js/images)

App/ - backend проекта

* App/Domain/ - слой предметной области
* App/Application/ - слой исполнения логики
* App/Infrastructure/ - слой сторонних библиотек
* App/Presentation/ - слой ввода/вывода
* calculate.php - скрипт расчета суммы проката

index.php - главная страница проекта

admin.php - админка

sql.sql - дамп БД

Архив доступен по ссылке -<https://disk.yandex.ru/d/tbGOzC7tF1_ZTA>

# Дока для calculate.php

итоговая стоимость = стоимость выбранного продукта \* количество дней

стоимость продукта зависит от указанного количества дней проката:

данные по тарифам указаны в столбце "TARIFF" в формате:

от N дней => цена

таких периодов может быть несколько.

если данные в "TARIFF" отсутствуют, то данные берутся из "PRICE" и не зависят от кол-ва дней

если выбраны дополнительные услуги, то стоимость каждой из них умножается на выбранное количество дней и прибавляется к итоговой стоимости

# GIT или Как прислать результат\*

1. Создать репозиторий в github
2. Залить исходный код в master ветку
3. Создать develop ветку как клон от master
4. Все изменения вносить в develop ветку
5. Создать PullRequest из develop в master. Не мержить!
6. → Прислать ссылку на PullRequest

*\* Кроме 8 задачи. Для задачи 8 условия сдачи описаны в самой задаче*

# Если остались силы, то можешь сделать дополнительные таски

1. Залить результаты на хостинг и прислать ссылку на сайт. Скрыть сайт от индексации
2. Пропустить html из index.php через валидатор W3 и исправить ошибки