

Практическая работа № 7.5

«Создание класса Response и класса обработки данных»

Цель работы: закрепление практических навыков работы с функциональным и объектно-ориентированным подходом.

Формируемые компетенции: ПК 5.4 «Производить разработку модулей информационной системы в соответствии с техническим заданием», ПК5.1 «Собирать исходные данные для разработки проектной документации на информационную систему».

Отчет о работе: предоставляется в составе отчета по практике. Студент защищает написанный код очно.

Задачи:

Задача 1.

Создать класс для общей (универсальной) обработки данных. Например, Data. Данный класс должны наследовать классы, связанные с данными: пользователя, постов, комментариев.

Создать следующие методы:

1) метод `validateData` – метод валидации данных. Метод должен возвращать признак сработало или нет хотя-бы одно из правил валидации в любом классе. Для реализации метода можно воспользоваться функцией [get object vars](#). Для корректной работы данного метода необходимо называть атрибуты дочернего класса, связанные с валидацией, с одинаковой «приставкой» (признаком). Например, все атрибуты должны содержать слово – `validate`.

2) метод `loadData` – метод загрузки данных из входного массива в атрибуты дочернего класса. Для реализации метода необходимо использовать [property exists](#);

Задача 2.

Создать класс Response для формирования ссылок и перенаправления клиента на необходимый адрес.

В классе создать следующие методы:

1) в конструктор класса передать экземпляр класса для работы с пользователем, созданного ранее.

2) метод `getLink` – метод для формирования конечного варианта ссылки для перехода на страницу. Метод принимает `url` и массив с параметрами. В методе необходимо

произвести проверку если пользователь авторизован и в массиве параметров отсутствует токен, то его необходимо добавить в массив. Произвести проверку на наличие знака «?» в url. В случае если массив не пустой и знак «?» отсутствует, добавить его. Метод должен возвращать url с параметрами из входного массива в формате «параметр=значение». Параметры между собой соединяются знаком «&;

3) метод `redirect` – метод для перенаправления клиента на определенную страницу. Метод принимает url и массив с параметрами. В методе необходимо вызвать функцию `header` с соответствующим заголовком перенаправления. url перехода сформировать с помощью ранее созданного метода и метода возвращающего адрес хоста запроса;

4) доработать конструктор. Если в url присутствует токен, но пользователь не авторизован, то перенаправить на главную страницу без токена url.

Доработать все страницы и логику в ранее созданных файлах и классах с использованием созданных классов в данной работе. Все ссылки и обработку данных производить через созданные методы.

Проверить корректность работы приложения.