

Практическая работа № 7.4

«Реализация авторизации пользователя»

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.

Доработать класс User.

Создать следующие методы:

1) метод `validateLogin` – метод валидации данных при авторизации пользователя. Метод должен возвращать признак сработало или нет хотя-бы одно из правил валидации. Для реализации метода можно воспользоваться функцией [get object vars](#);

2) метод `login` – метод «логирования» пользователя. В методе необходимо произвести проверку пароля и логина введенных пользователем и из БД. Также, используя метод `load`, произвести загрузку данных полученных из бд в атрибуты класса. Установить признак пользователя и его роль. Для установки роли использовать метод `isAdmin()`, который будет разработан далее. Сформировать токен для пользователя из набора случайных символов и записать его в БД. Метод должен возвращать признак успешной операции, либо возвращать информацию о некорректности данных (например, через атрибут валидации пароля);

3) метод `identity` – метод идентификации пользователя. Метод принимает `id` пользователя. Если `id` не передан, то данные по пользователю загружаются на основе токена.

4) Также, используя метод `load`, произвести загрузку данных полученных из бд в атрибуты класса;

5) метод `isAdmin` – метод определения прав пользователя. Вернуть признак совпадения логина и пароля текущего пользователя с данными администратора;

6) в методе `load` добавить формирование значения атрибута `isAdmin` с помощью метода, разработанного в пункте 7;

7) метод `logout` – метод «разлогирования» пользователя. Если пользователь авторизован в системе, то произвести выход его из системы, удалив токен в БД;

8) произвести доработку конструктора, если есть токен, необходимо вызвать метод `identity`.

Задача 2.

В коде страницы авторизации пользователя настроить форму, указав метод и файл обработчик формы.

В файле инициализации страницы авторизации, произвести проверку на метод запроса `Post` и загрузить данные от формы в экземпляр класса `User`. Убедиться, что данные были успешно загружены.

Произвести валидацию полученных данных. Сформировать отображение ошибок валидации на странице авторизации и загрузку в поля формы ранее введенных данных. В случае успеха произвести авторизацию пользователя, при успешном выполнении авторизации произвести перезагрузку на главную страницу с формированием токена в `url`.

Проверить, что происходит авторизация пользователя по токену.

Проверить корректность работы приложения.