Тест по Ruby on Rails

1. Назначение контроллера в MVC
   1. проверка корректности пользовательских данных
   2. обеспечение механизма сохранения данных пользователя в базу данных
   3. обеспечение связи между пользователем и системой
2. Назначение метода `before\_action` в Rails-контроллерах
   1. регистрация методов вызываемых до запуска всех или некоторых экшенов контроллера
   2. инициализация соединения с базой данной
   3. изменение порядка вызова методов
3. Имеется следующий контроллер:

class UsersController < ActionController::Base

def create

User.create(users\_params)

end

private

def users\_params

params.require(:user).permit(:login, :name)

end

end

Модель:

class User < ActiveRecord::Base

validates :login, :name,

presence: true

end

Таблица в базе:

users:

login (varchar(255)) | name (varchar(255)) | admin (boolean, default: false)

Что произойдет при передаче следующих параметров:  
 `{user: {login: 'vasya', name: 'pupkin', admin: true}}`?

* 1. контроллер выдаст ошибку с HTTP-статусом 422
  2. объект класса User будет сохранен в базу со значениями:  
     login: 'vasya', name: 'pupkin', admin: false
  3. объект класса User будет сохранен в базу со значениями:  
     login: 'vasya', name: 'pupkin', admin: true
  4. будет выдана ошибка валидации
  5. Свой вариант: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
     \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
     \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. Сколько экшенов зарегистрирует строка `resources :users`
   1. 1
   2. 5
   3. 7
   4. 10
2. В каком слое расположить код проверки прав доступа?
   1. Model
   2. View
   3. Controller
   4. свой вариант: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
      \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
      \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
3. Наиболее часто используемая в Ruby on Rails библиотека для работы с базой данных
   1. action\_pack
   2. model
   3. active\_record
   4. sprockets
4. Имеется ActiveRecord модель `User`. Укажите **способы** получения объекта данного класса, для которых значение колонки `login` равно строке `admin`?
   1. User.find(login: 'admin')
   2. User.find\_by(login: 'admin')
   3. User.where(login: 'admin').first
   4. User.where.login.eq('admin')
5. Что выдаст функция `save` ActiveRecord-модели при наличии ошибок валидаций
   1. true
   2. false
   3. nil
   4. вызовет исключение
6. Что выдаст функция `save!` ActiveRecord-модели при наличии ошибок валидаций
   1. true
   2. false
   3. nil
   4. вызовет исключение
7. Имеются сдедующие модели:

class User

has\_one :avatar

end

class Avatar

belongs\_to :user

end

Для организации соответствующих связей необходимо:

* 1. В таблицу users добавить колонку avatar\_id
  2. В таблицу avatars добавить колонку user\_id
  3. В таблицу users добавить колонку avatar\_id, а в таблицу avatars колонку user\_id
  4. Нужна дополнительная таблица связи avatars\_users с колонками avatar\_id и user\_id

1. Для чего нужны валидации модели?
   1. для запрета установки некорректного значений указанным параметрам модели
   2. для преобразования переданных в модель данных в соответствующий тип
   3. для проверки корректности состояния объекта до сохранения в базу данных
2. Для чего нужны callback-функции модели?
3. Какие из перечисленных методов ActiveRecord пропускают валидации и callback-функции?
   1. update\_all
   2. update\_columns
   3. update\_attributes
   4. touch
   5. update
4. Что такое проблема N+1?
5. Что не так со кодом `User.where('login=params[:name] or name=params[:name]')` и как его исправить?