**1) What is Metaprogramming in Ruby?**  What metaprogramming methods and techniques do you know? (How to create a new class with methods in runtime?\*\*)

[Metaprogramming](https://en.wikipedia.org/wiki/Metaprogramming) is a technique by which you can write code that *writes* code by itself dynamically at runtime. This means you can define methods and classes during runtime. Crazy, right? In a nutshell, using metaprogramming you can reopen and modify classes, catch methods that don’t exist and create them on the fly, create code that is [DRY](https://en.wikipedia.org/wiki/Don%27t_repeat_yourself) by avoiding repetitions, and more.

A metaclass is essentially a class that Ruby creates and inserts into the inheritance hierarchy to hold class methods, thus not interfering with instances that are created from the class.

Although, to get the metaclass you don’t need to actually reopen Object and add this hack. You can use singleton\_class that Ruby provides.

There’s one more way to create a class method, and that is by using [instance\_eval](http://ruby-doc.org/core-1.9.3/BasicObject.html#method-i-instance_eval)

Singletone

Т.е. мы можем добавить методы отдельному объекту класса *Dog*, которые не будут доступны другим экземплярам этого класса. Чаще всего такие методы называют синглтон-методами, потому что они принадлежат только одному единственному объекту.

snoopy = Dog.new

def snoopy.alter\_ego

"Red Baron"

end

2) What are rails concerns? What is their purpose and why they differ from modules?

3) Do you know about polymorphic associations? Why do we need them?

4) What is JSON API? What are the required parameters in JSON API SPECIFICATION?

Why use JSON API SPECIFICATION?

5) What’s ActiveResourse in rails?

6) What are callbacks in general? Do they exist outside rails and ruby? What purposes are they good for? Why would you want to refuse using callbacks?

7) What is object composition? What is dependency injection?

8) What are service objects? Why would you want to use them?

9) What are decorators? Why would you want to use them (the purpose)?

10) What GOF (gang of four) software design patterns do you know?

11) Can you describe SOLID principles and give a few examples in ruby?

12) Why do we need tests? What is TDD?

13) What is microservices architecture and why do we need to use it?

14) When would you refuse to use microservices?

15) What are asynchronous tasks and why do we need it? Does asynchronous functionality exist only in rails/ruby?

16) What is websockets technology?

17) What is ActiveRecord::Transaction why would you want to use it? What does it return?

If you nest seceral transaction in one big transaction the failure to