* **Programming and OOP questions**

1. What is OOP? What are the 3 principles of OOP and their meaning?
2. What is the difference between static and dynamic typing in programming languages?
3. What is difference between strong and weak typing in programming languages?
4. What is the difference between explicit and implicit typing in programming languages?
5. The fundamental difference between script and "ordinary" languages?
6. What is *namespace* and why use them?
7. What is the difference between thread and process
8. Explain, what is caching?
9. Where is it faster to add an element in the array (hash) in the middle or at the end?
10. What are the 3 levels of access control for methods in the class? Explain each.
11. What is a class? What is an object (class instance)? What is the difference?
12. What is a constructor? What is an object initialization?
13. What is MVC? Why use it?
14. How do MVC components interact with each other? The pattern of the pattern in general.
15. What is REST? Why use it?
16. What is CRUD?
17. Rules for building a REST API.
18. Where should the logic be in the application?
19. What are migrations for?
20. What is a Strong Params?
21. What is KISS, DRY, SOLID, YAGNI?
22. What design patterns do you know?
23. What is a singleton class?
24. What is a Monkey pathcing?
25. What is ORM?

* **Ruby questions**

1. What data types are used in Ruby? What is an array? hash? line? number? time? symbol?
2. Why is it in Ruby 1660/100 ≠ 16.6?
3. Why in Ruby 24.0 \* 0.1 ≠ 2.4?
4. What structures are there in ruby?
5. What is loop, while, map, each ?
6. What is the difference between each and map?
7. What loops and iterators are there in Ruby?
8. What variables are there, where are they used, where are they available (visibility fields)?
9. What is a variable with one @ and a variable with two @@?
10. What is the difference between *require* and *require\_relative*?
11. What is a gem? Why we need it?
12. How to create getter and setter methods in ruby?
13. What is attr\_reader, attr\_writer and attr\_accessor?
14. What does the keyword *self* mean?
15. What are singleton methods and singleton classes?
16. What is a module in ruby? What is the difference between a class and a module?
17. How is inheritance organized in Ruby?
18. What is the difference between include and extend? What is prepend?
19. Implementing multiple inheritance in ruby?
20. What are the ways to call methods in ruby?
21. What is proc, lambda, block? And what are the differences between them?
22. Threading in ruby?
23. What is safe navigation?
24. If in the module do extend self what will happen?

* **Ruby on Rails questions**

1. What is a ActiveRecord and what tools does it provide for working with objects?
2. What are the Model, View, Controller levels in Rails responsible for?
3. How does routing work? What are resource routes? How are they formed?
4. What are scopes? How to use?
5. What is a ActiveJob? When to use it?
6. What is a Asset Pipeline?
7. What is a serializer and what is it for? Where is it used? What is its main task?
8. What are validations? How to write your validations? Why do we need validations? Where are validations applied? Validation Examples
9. What is a Rack?
10. What is Haml, Slim? What are the advantages in your opinion, their use?
11. What does the multiple example.html.erb file extensions mean?
12. What is ERB? Can you decrypt the abbreviation?
13. Know and tell the folder structure of the Rails application
14. What relationships for model binding in a Rails application do you know?
15. Give examples of the use of has\_many, belongs\_to, has\_and\_belongs\_to\_many, has\_one, has\_many: through?
16. What are polymorphic bonds?
17. What is i18n (internationalization)?
18. What is dependent communication?
19. What is t.references?
20. What is scaffolding? Why is it used and where is it used?
21. How is caching implemented in rails?
22. Imagine that there is a huge table. users. How can you go through its elements as quickly as possible?
23. What is N+1 problem? How can I solve the N + 1 problem in Rails?
24. What are Service Objects, Form Objects, View Objects, Query Objects, what are they for?

* **Web development questions**

1. What happens after you enter the site name in the browser and press Enter? Explain in detail.
2. What are processes? How to work with them?
3. What is the difference between HTTP and HTTPS?
4. What is HTTP OPTIONS?
5. What encryption does HTTPS have?
6. What are sessions? What are they for?
7. How does HTTP work, what are the headers, how do cookies work, how does HTTPS work?

* **Application Testing questions**

1. Why do we need tests
2. What are TDD and BDD?
3. What are specs, factories, moki, stubs, musta\_matchers?
4. What are unit tests, integration tests, functional tests?
5. What gems for testing do you know?

* **Git questions**

1. What is a VCS? What is git? Why use it?
2. How to create a repository, connect an external repository?
3. How to download a remote repository?
4. What is commit? How to view commit history?
5. What file states exist on a Git system**?**
6. What is a branch in Git?
7. Differences between merge and rebase commands?
8. How to download the latest changes from a specific branch?
9. How to send my changes to a remote repository?
10. How to add changes to an already created commit? change the name of such a commit?
11. How to delete a branch locally and from a remote repository?
12. What are pull requests? How to create them?
13. How to transfer changes from one branch to another (2 ways)?
14. What is an SSH key for?
15. git commit - in what cases write -am, -a and -m?

* **Database questions**

1. What is a relational database?
2. What is a table, a tuple? What is a primary key?
3. How are relationships between tables implemented? What is a foreign key?
4. How does the SELECT statement work?
5. What are the types of JOIN? How does everyone work?
6. How do the INSERT, UPDATE, DELETE statements work?
7. What are indexes? What are they used for? Pros, cons?
8. What types of indices are there?
9. What are transactions?
10. Tell us about transaction isolation levels.
11. What is a ACID properties?
12. What is replication, what is it for?
13. What is database normalization and denormalization?

* **English questions**

1. Tell me about yourself
2. What did you like or dislike about your previous job?
3. What major challenges and problems did you solve/face? How did you handle them?
4. What is your greatest strength?
5. What is your greatest weakness?
6. How do you handle stress and pressure?
7. Describe a difficult work situation or project and how you overcame it?
8. Why do you want to go to this lab? Why do you want to learn Ruby?
9. What are your goals for the future ?
10. How did you hear about Ruby/Rails?
11. What do you know about our company?
12. What are tour hobbies?
13. What are three skills or traits you wish you had?
14. What qualities make a good leader?
15. Describe your perfect company?
16. What do you expect from these courses?
17. Tell me about your specialty, how is your study at the university?
18. Who are your heroes?
19. What is the last book you read?
20. Do you like to travel? What countries have you been to?
21. If you could go to any country in the world, where would you go and why?
22. Where did you study English?
23. Have you had any experience working in a team, if so, tell me about it?
24. Do you like sport? What is your favorite sport and why?
25. What kind of films do you like and why? What is your favorite movie?