1. Из каких основных сущностей состоит Spring-приложение?
2. Что такое инверсия контроля (IoC) и внедрение зависимостей (DI)?
3. Как эти принципы реализованы в Spring?
4. Что такое IoC контейнер?
5. Что такое Bean в спринге?
6. Расскажите про аннотацию @Bean?
7. Расскажите про аннотацию @Component?
8. Чем отличаются аннотации @Bean и @Component?
9. Расскажите про аннотации @Service и @Repository. Чем они отличаются?
10. Расскажите про аннотацию @Autowired
11. Расскажите про аннотацию @Resource
12. Расскажите про аннотацию @Inject
13. Расскажите про аннотацию @Lookup
14. \*Как работает инъекция прототипа в синглтон?
15. Можно ли вставить бин в статическое поле? Почему?
16. Расскажите про аннотации @Primary и @Qualifier
17. Как заинжектить примитив?
18. Как заинжектить коллекцию?
19. Расскажите про аннотацию @Conditional
20. Расскажите про аннотацию @Profile
21. Расскажите про ApplicationContext и BeanFactory, чем отличаются? В каких случаях что стоит использовать?
22. Расскажите про жизненный цикл бина, аннотации @PostConstruct и @PreDestroy
23. Расскажите про скоупы бинов? Какой скоуп используется по умолчанию?
24. Что изменилось в пятом спринге?
25. Расскажите про аннотацию @ComponentScan
26. Как спринг работает с транзакциями?
27. Расскажите про аннотацию @Transactional
28. Расскажите про аннотации @Controller и @RestController. Чем они отличаются?
29. Как вернуть ответ со своим статусом (например, 213)?
30. Что такое ViewResolver?
31. Чем отличаются Model, ModelMap и ModelAndView?
32. Расскажите про паттерн MVC, как он реализован в Spring?
33. Расскажите про паттерн Front Controller, как он реализован в Spring?
34. Что такое AOP? Как реализовано в спринге?
35. PROXY что это, для чего и как применяется в Spring
36. В чем разница между Filters, Listeners and Interceptors?
37. Можно ли передать в запросе один и тот же параметр несколько раз? Как?
38. Как работает Spring Security? Как сконфигурировать? Какие интерфейсы используются?
39. Что такое SpringBoot? Какие у него преимущества? Как конфигурируется? Подробно.
40. Какие задачи решает Spring Data?
41. Какова иерархия интерфейсов/классов репозитория в Spring Data JPA?
42. \*Как преодолеть проблему блокирующих вызовов?
43. Как вызвать транзакционный метод из того же класса?
44. \*Когда отрабатывает @Autowired а когда @Transactional? почему?
45. Почему нельзя отработать @Transactional в другом методе?
46. Структура HTTP запроса, ответа
47. HTTP методы
48. Ключевое отличие методов POST и GET
49. REST стиль
50. Идемпотентность: что это такое, методы
51. Как метод пост сделать идемпотентным?