1. Чем процесс отличается от потока?
2. Чем Thread отличается от Runnable? Когда нужно использовать Thread, а когда Runnable?
3. Как выполнить две задачи параллельно?
4. Что такое монитор? Как монитор реализован в java?
5. Что такое синхронизация? Какие способы синхронизации существуют в java?
6. Как работают методы Object wait(), notify() и notifyAll()?
7. В каких состояниях может находиться поток?
8. Что такое семафор? Как он реализован в Java?
9. Что обозначает ключевое слово volatile? Почему операции над volatile переменными не атомарны?
10. Расскажи про правила «happens-before» и Java Memory Model (JMM).
11. Для чего нужны Atomic типы данных? Чем отличаются от volatile?
12. Что такое потоки демоны? Для чего они нужны? Как создать поток-демон?
13. Что такое приоритет потока? На что он влияет? Какой приоритет у потоков по умолчанию?
14. Как работает Thread.join()? Для чего он нужен?
15. Чем отличаются методы yield() и sleep()?
16. Как правильно остановить поток? Для чего нужны методы .stop(), .interrupt(), .interrupted(), .isInterrupted().
17. Чем Runnable отличается от Callable?
18. Что такое FutureTask?
19. Что такое deadlock?
20. Что такое livelock?
21. Что такое race condition?
22. Что такое Фреймворк fork/join? Для чего он нужен?
23. Что означает ключевое слово synchronized? Где и для чего может использоваться?
24. Что является монитором у статического синхронизированного класса?
25. Что является монитором у не статического синхронизированного класса?
26. Способы управления потоками java.util.сoncurrent
27. Расскажи про устройство и алгоритм работы ConcurrentHashMap
28. Что знаешь про CopyOnWriteArrayList?
29. Stream API & ForkJoinPool Как связаны, что это такое.
30. Зачем используются thread local переменные?
31. Как реализовать двусторонний обмен данными между потоками?
32. Какими коллекциями пользоваться в многопоточной среде?