1. Read: https://www.interviewbit.com/multithreading-interview-questions/#class-level-lock-vs-object-level-lock
2. Explain what is volatile in java? (we also use it in Singleton)
3. How to create a new thread(Please also consider Thread Pool approach)?
4. Difference between Runnable and Callable?
5. What is the difference between t.start() and t.run()?
6. Which way of creating threads is better: Thread class or Runnable interface?
7. What are the thread statuses?
8. What is deadlock?
9. How do threads communicate each other?
10. What's the difference between class lock and object lock?
11. What is join() method?
12. what is yield() method
13. What is ThreadPool? How many types of ThreadPool? What is the TaskQueue in ThreadPool?
14. Which Library is used to create ThreadPool? Which Interface provide main functions of thread-pool?
15. How to submit a task to ThreadPool?
16. What is the advantage of ThreadPool?
17. Difference between shutdown() and shutdownNow() methods of executor
18. What is Atomic classes? How many types of Atomic classes? Give me some code example of Atomic classes and its main methods. when to use it?
19. What is the concurrent collections? Can you list some concurrent data structure (Thread-safe)
20. What kind of locks do you know? What is the advantage of each lock?
21. What is future and completableFuture? List some main methods of ComplertableFuture.
22. Type the code by your self and try to understand it. (package com.chuwa.tutorial.t08_multithreading)

23. Write a code to create 2 threads, one thread print 1,3,5,7,9, another thread print 2,4,6,8,10. (solution is in com.chuwa.tutorial.t08_multithreading.c05_waitNotify.OddEventPrinter)
    a. One solution use synchronized and wait notify
    b. One solution use ReentrantLock and await, signal

```
Thread-0: 1
Thread-1: 2
Thread-0: 3
Thread-1: 4
Thread-0: 5
Thread-1: 6
Thread-0: 7
Thread-1: 8
Thread-0: 9
Thread-1: 10

Process finished with exit code 0
```

24. create 3 threads, one thread ouput 1-10, one thread output 11-20, one thread output 21-22. threads run sequence is random. (solution is in com.chuwa.exercise.t08_multithreading.PrintNumber1)

```
Thread-0: 1
Thread-0: 2
Thread-0: 3
Thread-0: 4
Thread-0: 5
Thread-0: 6
Thread-0: 7
Thread-0: 8
Thread-0: 9
Thread-0: 10
Thread-2: 11
Thread-2: 12
Thread-2: 13
```

```
Thread-2: 14
Thread-2: 15
Thread-2: 16
Thread-2: 17
Thread-2: 18
Thread-2: 19
Thread-2: 20
Thread-1: 21
Thread-1: 22
Thread-1: 23
Thread-1: 24
Thread-1: 25
Thread-1: 26
Thread-1: 27
Thread-1: 28
Thread-1: 29
Thread-1: 30
```

25. completable future:
    a. Homework 1: Write a simple program that uses CompletableFuture to asynchronously get the sum and product of two integers, and print the results.
    b. Homework 2: Assume there is an online store that needs to fetch data from three APIs: products, reviews, and inventory. Use CompletableFuture to implement this scenario and merge the fetched data for further processing. (需要找public api去模拟，)
        i. Sign In to Developer.BestBuy.com
        ii. Best Buy Developer API Documentation (bestbuyapis.github.io)
        iii. 可以用fake api https://jsonplaceholder.typicode.com/
    c. Homework 3: For Homework 2, implement exception handling. If an exception occurs during any API call, return a default value and log the exception information.