coding test：

coding test目标

请说明项目打包和执行的方式

代码以java书写，采用maven工程，每题有自己的独立类分别位于first、second、third包下；

打包：mvn clean package

执行：intellij idea下直接运行对应的主类

请说明发布的方法

不限语言和平台

如有问题，请随时电话沟通

即使你的项目无法打包执行，请仍然提交您的代码

请确保您的github是公共可访问的

一、问题：

F(n) = F(n-1) + F(n-2), where F1 =1, F2 = 2

F1 =1

F2 =1

F3 =2

F4 =3

F5 =5

F6 =8

F7 =13

F8 =21

F9 =34

F10=55

F11=89

F12=144

可见index=12的F12是第一个包含3个数字的Fn，那么第一个包含1000个数字的Fn的index是多少？

答：4782

见com.pingan.first. FibonacciSeries 类

二、将1，2，3，4，……，99，100的顺序排列，排序成100，1，99，2，98，3，……，51，50这样的非等差数列。要求:空间复杂度o（1），时间复杂度o（n）。

答：

见com.pingan.second. SeriesOrder 类

三、素数是自然数中大于1，且只能被正整数中的1和自身整除的自然数。

比如10以下的素数和为 ：2 + 3 + 5 + 7 = 17

请计算2,000,000以下的所有素数和

答：142913828922

见 com.pingan.third. PrimeNumSum类

四、问答：

排序算法的平均时间复杂度的下限是多少？请尝试证明您的观点

答：以快速排序为例，快排在每次Partition 都很均匀的情况下效率是最高的，其下限的时间复杂度可推论如下：

1. 假设对 n 个关键字进行排序，期望时间T(n)；
2. 在每次Partition均分的情况下, 需要递归 log2n次；
3. 第一次需要n 次比较，然后获得的枢轴将数组一分为二，那么各自还需要T(n/2)的时间；

T(n) ≤ 2T(n/2) + n

1. 依次推n/2 区间、n/4区间、n/8区间耗时如下：

T(n)≤2(2T(n/4)+n/2) +n=4T(n/4)+2n  
T(n)≤4(2T(n/8)+n/4) +2n=8T(n/8)+3n

….

由于递推的深度 = log2n，所以

T(n)≤4(2T(n/8)+n/4) +2n=8T(n/8)+3n = 8T(n/8)+ log2n \* n

同时左边系数2、4、8、、、、 为深度的 2次幂，把系数替换后得

T(n)≤ nT(n/n) + n log2n ≤ nT(1) + n log2n

5) 由于T(1) = 0, 快排在Partition 拆分均匀情况下时间复杂度为 O(nlog2n)

五、目前有线索池系统的需求，需要记录线索来源和线索信息，线索信息的内容除了客户名和多种（至少）一种联系方式外，并不明确信息的数量和内容。另有下单系统，系统中有客户名和一种联系方式，包含供应商信息等确定的60个不同类型字段。线索池的客户名和下单的客户名不一定相同。此外还有代理商网咯系统，会得到线索池分发的信息。

请尝试设计线索池系统，其中下单系统和代理商网络是旧有系统，有一定改造难度。总体目标是实现线索有效性的评价，尽可能提高线索的利用率（供应商对线索的有效利用）。

答：

1. 线索系统除了客户名与联系方式外其它信息数量与内容不明确，存J储上可以考虑Nosql 如：mongodb、elastic search、hbase、同时考虑线索信息与订单等系统数据模糊匹配优先考虑elastic search、mongodb
2. 下单系统和代理商网络是旧有系统有一定改造难度，但他们与线索系统的通讯方式不同，针对代理商系统可以接收线索池分发消息，那么可以考虑采集代理商系统数据，线索系统做数据分析，通过匹配模型提高匹配辨识度，将匹配后的信息推送至代理商系统；而下单系统可以通过接口主动查询线索系统；
3. 综上，线索系统可能需要根据应用方情况提供线索实时匹配查询接口和线索匹配推送服务，实时接口可以通过rpc 方式提供服务，推送延用老系统mq
4. 请简述jvm模型，包括并发模型，请至少说明两项不涉及garbage的内容
5. jvm 体系结构

Java栈

类加载子系统

执行引擎

PC寄存器

本地方法栈

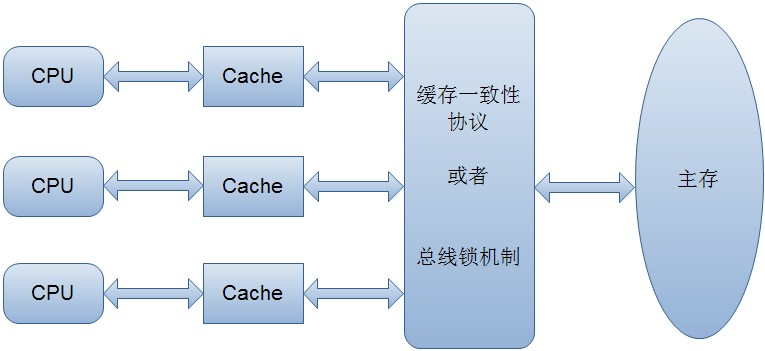
垃圾回收系统

直接内存

堆

方法区

2) java 并发操作主要涉及针对java堆、方法区内的数据共享，通过如下jmm模型可以看出在不采取相应措施情况下，多线程操作共享数据会有线程安全问题；



3）并发模式下，线程安全的解决方式一般有：

a、volatile 解决变量可见性、防止指令重排序

b、synchronized、lock等重量级锁

c、cas 无锁

d、ThreadLocal 线程隔离