1. 内存模型

类加载过程，classloader，字节码， .class文件，二进制文件，双亲委派机制。

线程共享部分，堆（字符串常量池），方法区（元空间）存放类信息，

线程不共享部分，虚拟机栈，本地方法栈（native方法），程序计数器，栈帧（局部变量表，操作数栈，方法返回地址，动态链接）

垃圾回收，新生代，老年代，伊甸区，s0，s1，minorgc和fullgc。对象来到eden，存不下就minorgc，还存不下就进老年代，还存不下就oom。被minorgc后进s0或s1，在其中每活过一次gc年龄加1。超过15进老年代。

死亡判断，引用计数（两对象相互引用）和可达性分析（gcroot），强软弱虚

垃圾收集算法，标记清除（内存碎片），标记复制（内存缩小到一半，适合新生代，其他适合老年代），标记整理

1. mysql
2. 基础

Mysql分为service层和存储引擎层，连接器、数据库连接（TCP长连接），查询缓存，解析sql（解析器），执行器（预处理，优化，执行），索引下推（组合索引a，b，当查询条件a＞1 and b=2 ，索引遇到＞会失效，b的判断交给存储引擎而不是执行器，减少回表次数）。

1. 行，页，区，段，表空间，行格式（变长字段长度列表，null值列表，记录头信息，隐藏的row\_id，trx\_id事务id，roll\_ptr记录上一个版本的指针，undolog，MVCC）
2. MVCC针对快照读，在可重复读隔离级别下，每次开启事务，会生成快照ReadView，里面包含了当前活跃事务id列表，活跃事务id最小值，活跃事务id最大值的下一个事务id，创建该ReadView的事务id。读取一条记录时会判断该记录trx\_id小于min\_id，则该记录可见，如果大于max\_id，则不可见，如果位于这之间，则判读是否存在于ids中，存在则不可见(不能完全解决幻读)，沿着记录指向的undolog找下一条记录，不存在则可见
3. 脏读，不可重复读，幻读，原子性(undo log)，一致性，隔离性(MVCC)，持久性（redo\_log）
4. 全局锁，表锁，行锁（记录锁（x，s型），间隙锁，next-key lock），updata语句如果条件没索引，进行全表扫描，就会锁全表。Sql语句后面加/g可以分析加的什么锁
5. Bufferpool（数据页，索引页，undo页），undolog，redolog，binlog。读取数据先查bufferpool，没有再读磁盘，读完放bufferpool，修改数据先记录undolog（写到undo页），修改bufferpool，标记脏页，记录相应redolog（redolog buffer）。至此更新完成（WAL）。

redolog刷盘（每隔一秒，或者事务提交），redolog写满了会擦除掉已经刷入磁盘的脏页的记录，数据不完整，不能做备份，主从复制（写入binlog，同步binlog（同步复制，异步复制，半同步复制），回放binlog），只能用于掉电等故障恢复。

Binlog刷盘，先写到binlogcache，等事务提交的时候刷盘

1. mybatis二级缓存
2. Spring自动装配
3. Bean生命周期
4. Sql调优
5. 服务变慢排查
6. Redis持久化
7. SpringMVC流程
8. 雪花算法时钟回拨
9. Mq订阅方式