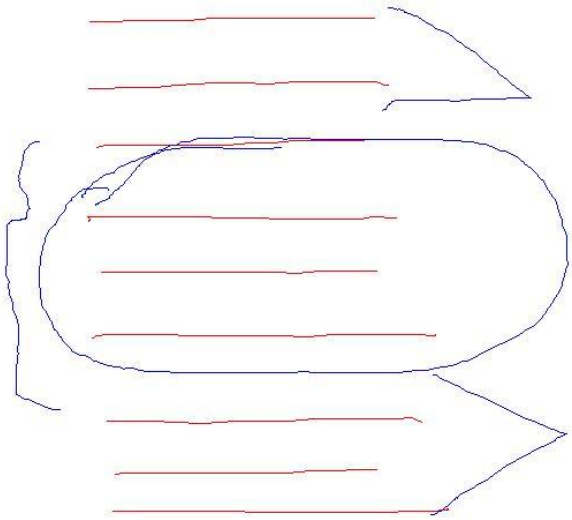
Java SE Lesson 16

1. 如果某个 synchronized 方法是 static 的，那么当线程访问该方法时，它锁的并不是 synchronized 方法所在的对象，**而是 synchronized 方法所在的对象所对应的 Class 对象，因为 Java 中无论一个类有多少个对象**，这些对象会对应唯一一个 Class 对象，因此当线程分别访问同一个类的两个对象的两个 static，synchronized 方法时，他们的执行顺序也是顺序的，也就是说一个线程先去执行方法，执行完毕后另一个线程才开始执行。
2. synchronized 块，写法：
   synchronized(object)
   {

   }
   表示线程在执行的时候会对 object 对象上锁。
3. synchronized 方法是一种粗粒度的并发控制，某一时刻，只能有一个线程执行该 synchronized 方法;synchronized 块则是一种细粒度的并发控制，只会将块中的代码同步，位于方法内、synchronized 块之外的代码是可以被多个线程同时访问到的。



4. 死锁（deadlock）
5. wait 与 notify 方法都是定义在 Object 类中，而且是 final 的，因此会被所有的 Java 类所继承并且无法重写。这两个方法要求在调用时线程应该已经获得了对象的锁，因此对这两个方法的调用需要放在 synchronized 方法或块当中。**当线程执行了 wait 方法时，它会释放掉对象的锁**。
6. 另一个会导致线程暂停的方法就是 Thread 类的 sleep 方法，它会导致线程睡眠指定的毫秒数，**但线程在睡眠的过程中是不会释放掉对象的锁的**。