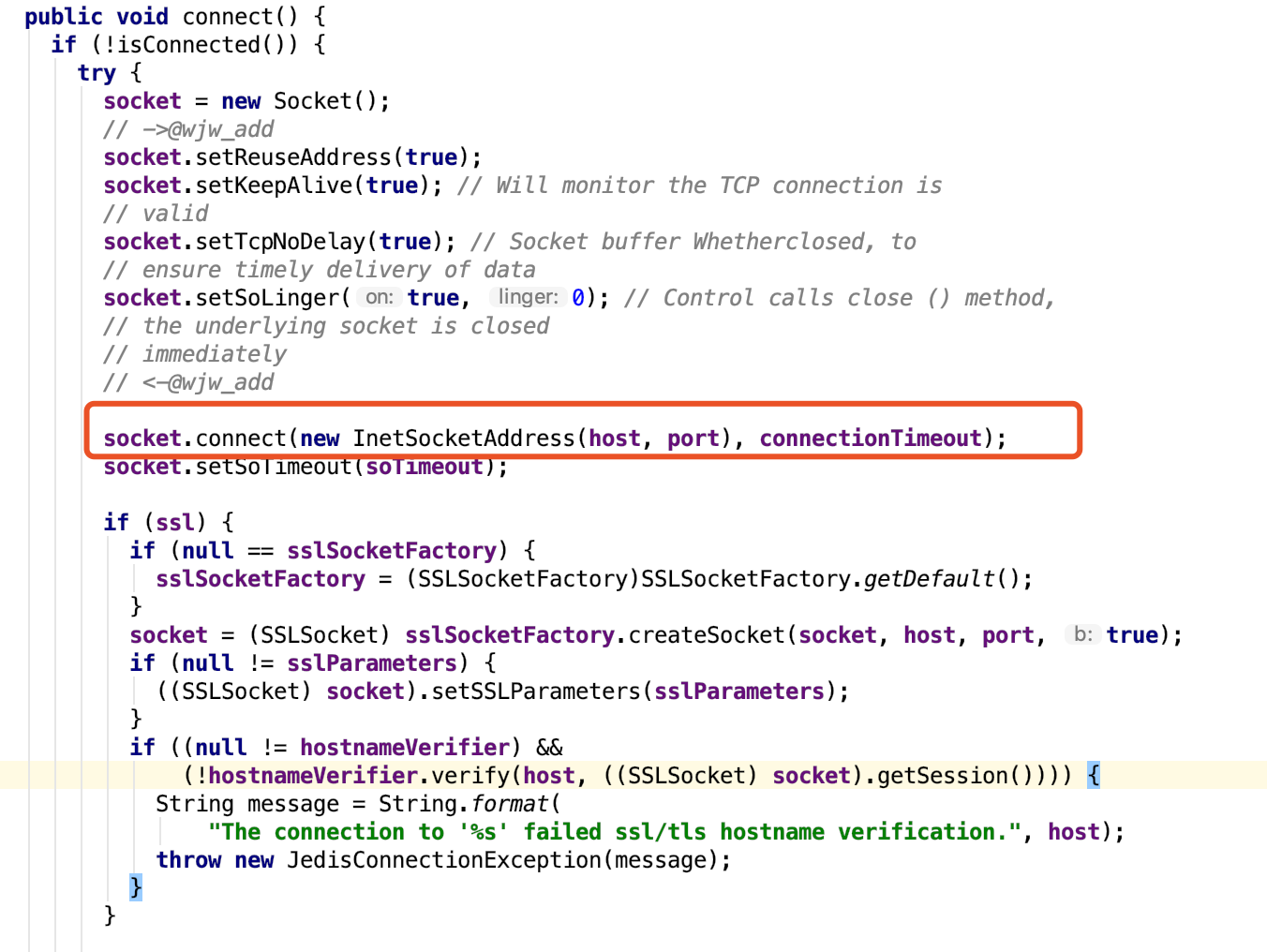
使用jedis连接redis时，如果多个线程共享一个jedis实例，可能会出现多线程并发问题。如下：







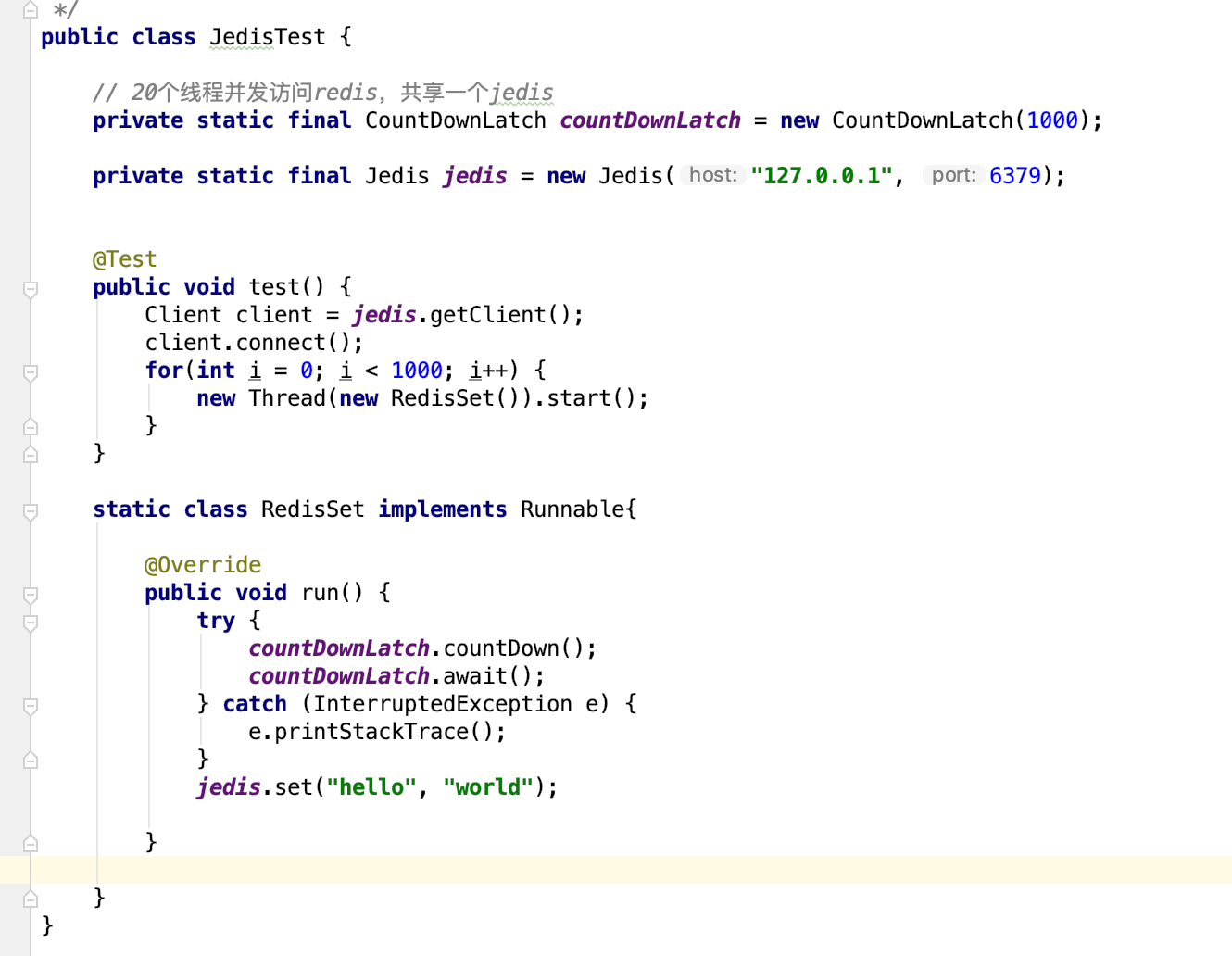
原因是因为我们在执行set命令时，jedis会通过Socket去连接Redis，每次都会去调用connect方法，该方法会进行Socket的初始化，上面出现的问题就是，有一个线程已经调用socket的connect方法了，又有一个线程执行了connect方法，就会出现上面的错误。

想要避免上面的问题，我们可以在使用jedis时，先连接一把，如下：



上面是共享socket出现的问题。

当我们的并发访问量设置为1000时，



又会出现新的问题

如下：



Protocol error: invalid multibulk lengt是因为多线程通过RedisInputStream和RedisOutputStream读写缓冲区的时候引起的问题造成的数据问题不满足RESP协议引起的。举个简单的例子，例如多个线程执行命令,线程1执行 set hello world命令。本来应该发送：

\*3\r\n$3\r\nSET\r\n$5\r\nhello\r\n$5\r\nworld\r\n

但是线程执行写到

\*3\r\n$3\r\nSET\r\n$5\r\nhello\r\n

然后被挂起了，线程2执行了写操作写入了' '，然后线程1继续执行，最后发送到redis服务器端的数据可能就是：

\*3\r\n$3\r\nSET\r\n$5\r\nhello\r\n' '$5\r\nworld\r\n

至于java.net.SocketException: Connection reset或ReadTimeout错误，是因为redis服务器接受到错误的命令，执行了socket.close这样的操作，关闭了连接。服务器会返回复位标志"RST"，但是客户端还在继续执行读写数据操作。

  jedis本身不是多线程安全的，这并不是jedis的bug，而是jedis的设计与redis本身就是单线程相关，jedis实例抽象的是发送命令相关，一个jedis实例使用一个线程与使用100个线程去发送命令没有本质上的区别，所以没必要设置为线程安全的。但是如果需要用多线程方式访问redis服务器怎么做呢？那就使用多个jedis实例，每个线程对应一个jedis实例，而不是一个jedis实例多个线程共享。一个jedis关联一个Client，相当于一个客户端，Client继承了Connection，Connection维护了Socket连接，对于Socket这种昂贵的连接，一般都会做池化，jedis提供了JedisPool。