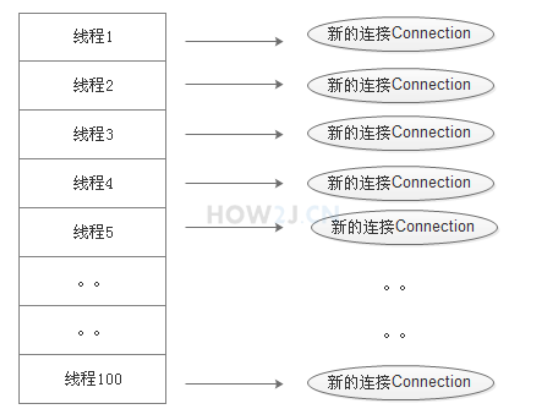
**一，传统方式的缺点**

参考DAO笔记，我们会发现，每个线程在执行SQL语句时，都会取得一个Connection对象，并且在使用完毕后，关闭Connection

**缺点：**

**1，创建和关闭Connection是比较耗时间的，当多线程并发时，多个Connection不断创建关闭，系统会变得很卡顿**

**2，同时，一个数据库支持的Connection总数是有限的，如果并发量过大，那么数据库的Connection总数会被消耗光，后续线程发起的数据库连接请求就会失败**



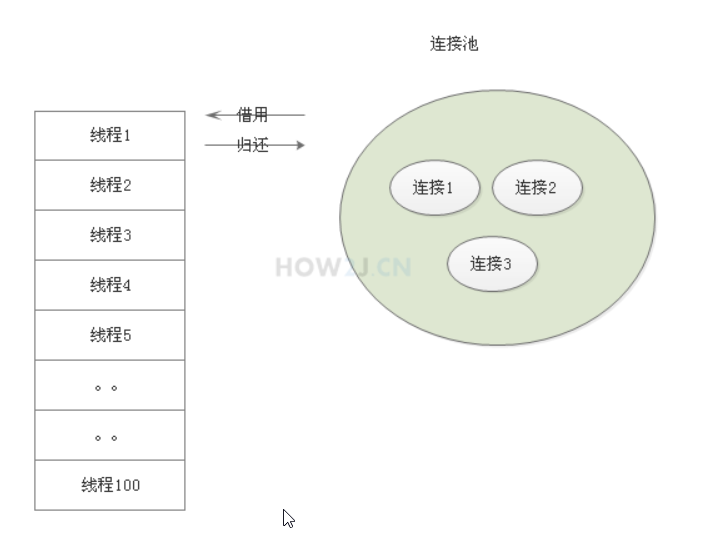
**二，数据库连接池**

**数据库连接池——创建数据库连接池时，同时也会创建一定数量的Connection**

**多线程任务中，一个线程如果需要连接数据库，那么这个线程不需要自己创建Connection，而是向数据库连接池借用Connection，使用完后还给数据库连接池**

**如果数据库连接池里面的Connection都被借用了，那么后续的线程如果想要用Connection，就只能等待，直到有Connection被归还**

**整个过程，Connection都不会被被关闭，而是不断循环使用，节约效率**

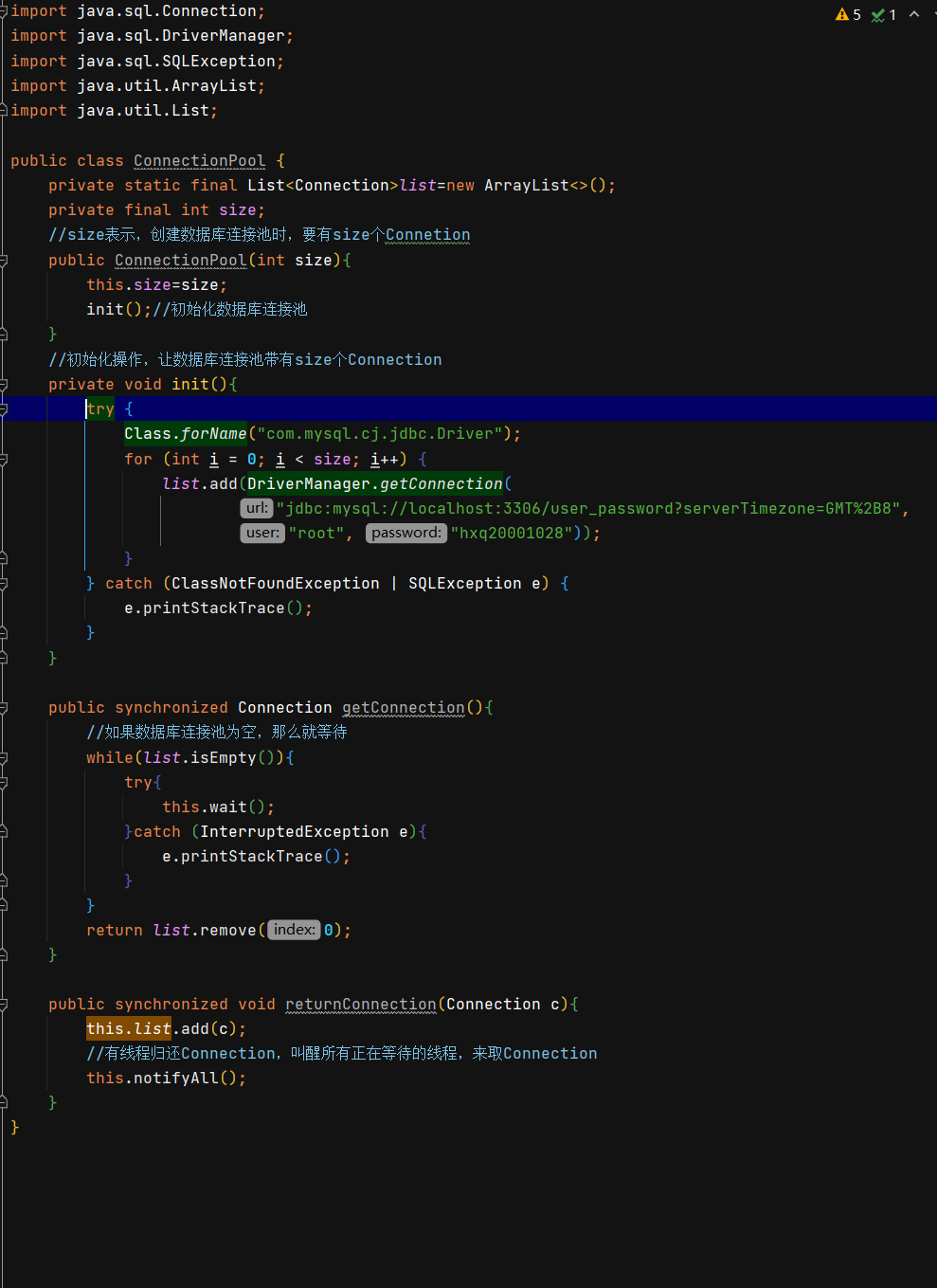


**三，使用数据库连接池**

**实现DataSource接口**

**市面上比较知名的数据库连接池——DBCP，C3P0，Druid**

**使用数据库连接池后，我们就不需要编写数据库连接代码了**



1，创建数据库连接池实例对象时，会初始化 size个Connection连接

2，如果一个线程在调用getConnection( )时，没有Connection了，那么这个线程会进入wait状态，当有线程归还Connetion时，会调用 notifyAll( )来唤醒所有 wait的线程，让它们来取Connection

**1，为什么getConnection( )和returnConnection( )要设置为 synchronized**

**因为如果getConnection( )不为synchronized，两个线程都是调用这个方法，但是只剩下一个Connection，那么就有一个会拿到null**