// 多线程查找数据

**int** corePoolSize = 5;

// 最大线程数

**int** maximumPoolSize = 5;

// 超过 corePoolSize 线程数量的线程最大空闲时间

**long** keepAliveTime = 3;

// 以秒为时间单位

TimeUnit unit = TimeUnit.*SECONDS*;

// 创建工作队列，用于存放提交的等待执行任务

BlockingQueue<Runnable> workQueue = **new** ArrayBlockingQueue<Runnable>(5);

ThreadPoolExecutor threadPoolExecutor = **null**;

SelectBottleNumByType selectAllBottleDatQuery1 = **new** SelectBottleNumByType(

bottleMapper, 1, level, companyAreaCode);

SelectBottleNumByType selectAllBottleDatQuery2 = **new** SelectBottleNumByType(

bottleMapper, 2, level, companyAreaCode);

SelectBottleNumByType selectAllBottleDatQuery3 = **new** SelectBottleNumByType(

bottleMapper, 3, level, companyAreaCode);

SelectBottleNumByType selectAllBottleDatQuery4 = **new** SelectBottleNumByType(

bottleMapper, 4, level, companyAreaCode);

SelectBottleNumByType selectAllBottleDatQuery5 = **new** SelectBottleNumByType(

bottleMapper, 5, level, companyAreaCode);

List<Future<Integer>> futureTaskList = **new** ArrayList<Future<Integer>>(5);

// 创建线程池

**try** {

threadPoolExecutor = **new** ThreadPoolExecutor(corePoolSize,

maximumPoolSize, keepAliveTime, unit, workQueue,

**new** ThreadPoolExecutor.AbortPolicy());

futureTaskList.add(threadPoolExecutor

.submit(selectAllBottleDatQuery1));

futureTaskList.add(threadPoolExecutor

.submit(selectAllBottleDatQuery2));

futureTaskList.add(threadPoolExecutor

.submit(selectAllBottleDatQuery3));

futureTaskList.add(threadPoolExecutor

.submit(selectAllBottleDatQuery4));

futureTaskList.add(threadPoolExecutor

.submit(selectAllBottleDatQuery5));

} **finally** {

**assert** threadPoolExecutor != **null**;

// 关闭线程池

threadPoolExecutor.shutdown();

}

**int** zhengchang = 0;

**int** guojian = 0;

**int** guoqi = 0;

**int** baofei = 0;

**int** quanbu = 0;

**try** {

zhengchang = futureTaskList.get(0).get();

guojian = futureTaskList.get(1).get();

guoqi = futureTaskList.get(2).get();

baofei = futureTaskList.get(3).get();

quanbu = futureTaskList.get(4).get();

} **catch** (NumberFormatException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

} **catch** (InterruptedException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

} **catch** (ExecutionException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}