Java服务器端Socket线程池计算机等级考试 PDF转换可能丢失图片或格式，建议阅读原文 https://www.100test.com/kao_ti2020/643/2021_2022_Java_E6_9C_8D_E5_8A_A1_c97_643992.htm 编辑特别推荐: 全国计算机等级考试（等考）指定教材 全国计算机等级考试学习视频 全国计算机等级考试网上辅导招生 全国计算机等级考试时间及科目预告 百考试题教育全国计算机等级考试在线测试平台 全国计算机等级考试资料下载 全国计算机等级考试论坛 import java.util.Vector. import java.net.*. import java.io.*. public class ThreadPool { public static final int MAX_THREADS = 100. public static final int MAX_SPARE_THREADS = 50. public static final int MIN_SPARE_THREADS = 10. public static final int WORK_WAIT_TIMEOUT = 60 * 1000. protected Vector pool. //存放空闲线程 protected MonitorRunnable monitor. //A monitor thread that monitors the pool for idel threads. protected int maxThreads. //Max number of threads that you can open in the pool. protected int minSpareThreads. //Min number of idel threads that you can leave in the pool. protected int maxSpareThreads. //Max number of idel threads that you can leave in the pool. protected int currentThreadCount. //Number of threads in the pool. protected int currentThreadsBusy. //Number of busy threads in the pool. protected boolean stopThePool. //Flag that the pool should terminate all the threads and stop. /** * Construct */ public ThreadPool() { maxThreads = MAX_THREADS. maxSpareThreads = MAX_SPARE_THREADS. minSpareThreads = MIN_SPARE_THREADS. currentThreadCount = 0.

```
currentThreadsBusy = 0. stopThePool = false. } /** * 启动线程池 */
public synchronized void start() { adjustLimits(). //调整最大和最
小线程数及最大和最小多余线程数.
openThreads(minSpareThreads). //打开初始线程 monitor = new
MonitorRunnable(this). //Runnable对象实例 //A monitor thread
that monitors the pool for idel threads. } public void
setMaxThreads(int maxThreads) { this.maxThreads = maxThreads.
} public int getMaxThreads() { return maxThreads. } public void
setMinSpareThreads(int minSpareThreads) { this.minSpareThreads
= minSpareThreads. } public int getMinSpareThreads() { return
minSpareThreads. } public void setMaxSpareThreads(int
maxSpareThreads) { this.maxSpareThreads = maxSpareThreads. }
public int getMaxSpareThreads() { return maxSpareThreads. } /** *
线程池管理方法. * 当空闲队列线程中没有空闲线程时,则增加
处理(空闲)线程数量. * 如果线程数量已达到最大线程数,则新
的连接进行等待. * 当请求到来,且有空闲线程时调用处理线程
进行具体业务处理. * @param r ThreadPoolRunnable */
```