# 线程基础习题

1. 线程的定义：

数据+程序+CPU //线程就是一条在程序代码中独立执行的路径

1. 给出runnable的定义：

对于线程的一系列操作的API //一个runnable就是一段封装在对象中的代码序列，它的类实现了Runnable接口

1. Thread类和Runnable接口完成了什么？

//Thread提供了一个底层操作系统线程架构的统一接口；Runnable接口为关联了Thread对象的线程提供执行代码

1. 指出创建一个runnable对象的两种方式：

（1）.实现接口并new // 创建一个实现了接口的匿名类；使用lambda表达式，

1. 指出关联一个runnable到一个Thread对象的两种方式：

（1）.创建一个Thread对象并以参数的形式传给runnable的构造器中

//（2）继承Thread类并重写run方法.

1. 指出5种Thread状态：

//线程名称；线程存活标识；线程的执行状态；线程的优先级；线程是否为守护线程的标识

1. 判断对错：默认线程的名字会以Thd-作为前缀。

错，

1. 如何给线程赋予非默认的名称：

thread.setName(name) ,name为你想要赋予的名称

1. 如何确定线程是死是活？

thread.isAlive（）；

1. 指出Thread.State枚举的所有常量：

RUNNABLE正在运行、TERMINATED已被终止 //补充：NEW线程未开始执行；BLOCKED线程被阻塞，正在等待一个监听锁；WAITING正在等待另外一个线程执行特定的操作

1. 如何获取当前线程的执行状态：

thread.getState();

1. 给出优先级的定义：

重要程度或者先后顺序

1. 如何通过setPriority（）来影响应用程序跨操作系统的可移植性：

//使用该函数会影响应用程序在操作系统之间的可移植性，因为不同的调度器会采取不同的方式来处理优先级

1. 确定Thread的void setPriority（int priority）方法参数的取值范围：

//介于Thread.MIN\_PRIORITY和Thread.MAX\_PRIORITY之间。

1. 判断对错：当应用程序的最后一个非守护线程死亡之后，守护线程也会自动死亡以使得应用程序退出。 对
2. 在一个正在运行或者已经死亡的Thread对象上调用Thread的void start（）方法会发生什么：

不会启动该线程，因为它已经死亡了，无法复活。 //会抛出一个异常

1. 如何在windows上停止一个无法种终止的程序？

//同时按住Ctrl键和C键

1. 确定组成Thread中断机制的所有方法。

//void interrupt();static boolean interrupted();boolean isInterrupted

1. 判断对错：Boolen isInterrupt（）方法清除了线程的中断状态。

错，该方法只返回是否中断；void interrupt会设置为中断，static Boolean interrupted会返回是否中断并清除中断

1. 当线程中断了，该线程会如何反应？

//抛出InterruptedException异常

1. 给出忙循环的定义：

//设计用来消耗时间的一条循环语句

1. 指出让一条线程等待另一个线程直至死亡的Thread方法：

void jion();

1. 指出让一条线程睡眠的Thread方法：

thread.sleep（time），time为睡眠时间，单位为ms

1. 编写一个名为IntSleep的应用程序，其创建一条后台线程，不断地打印出Hello,之后睡眠100ms。在睡眠了2S之后，默认主线程应该中断后台线程，此线程在打印出interrupt之后跳出循环

