**学习总结**

姓名 董金伟 专业 计算机科学与技术 班级 2班 日期 2022/2/16

1. 当天所学习主要内容（做一总结）
2. 线程池：
3. 好处：

线程池就是事先将线程对象放到一个容器中，当使用的时候不用new线程，而是直接在线程池中取线程即可，这样可以创建线程的时间，提高代码执行效率。

1. 线程池的种类：

单线程线程池、固定大小线程池、可缓存的线程池、支持周期执行的固定大小线程池

Executors类中的静态方法创建线程池。

单线程线程池：

所有任务的执行顺序按照任务的提交顺序执行。

newSingleThreadExecutor();

固定大小线程池：

每次提交一个任务就是一个线程，直到线程达到线程池的最大大小。

newFixedThreadExecutor();

可缓存的线程池：

线程池不会对线程池中的线程个数限制。

newCachedThreadExecutor();

支持周期执行的固定大小线程池：

线程池中线程个数固定，可以周期性的执行操做。

newScheduledThreadExecutor();

1. 线程池的使用

创建线程池；

执行execute方法将线程添加到线程池；

执行shutdown关闭线程池；

1. volatile
2. 并发编程中的三个概念：
3. 原子性：

一个操作或者多个操作要么全部执行，执行过程中不会被任何的因素打断；要么就全部不执行。

如：银行账户转帐。

通过同步块、同步方法实现原子性。

1. 可见性：

多个线程访问同一个变量时，一个线程修改了这个变量，其他的线程能够立即看到修改后的值。

1. 有序性：

程序执行的顺序按照代码的先后顺序执行。

指令重排：处理器为了提高程序的运行效率，可能会对输入的代码进行优化，它不保证各个语句的执行先后顺序同代码中的顺序一致，但是它会保证程序最终执行结果和代码顺序执行的结果保持一致。

指令重排不会影响单线程的执行结果，但可能会影响多线程的执行结果。

要想程序正确执行，必须保证原子性、可见性、有序性。只有有一个没有保证，就有可能导致程序执行不正确。

1. volatile的作用

volatile修饰全局变量。

1. 保证了不同线程对该变量进行操作时的可见性。
2. 禁止指令重排。不能将 在 对volatile变量访问的前面的语句放在其后执行，也不能将 在 对volatile变量访问的后面的语句放在前面执行。

C、不保证原子性。

需要保证操作是原子性操作，才能保证使用volatile关键字的程序在并发时能够正确执行。

1. 学习中所存在的问题及解决办法

问题：在完成冰淇淋作业时，wait方法抛出一个异常。InterruptedException。

解决：wait需要在同步代码块中使用，反复检查，发现忘记讲代码放到同步代码块中。

3．学习收获及建议

线程的6种状态；

1. 新建
2. 可运行
3. 阻塞
4. 等待
5. 潮湿等待
6. 终止

四种线程池：

1. 单线程线程池：顺序执行newSingleThreadExecutor();
2. 固定大小线程池：每提交一个任务就是一个线程，直到设置的线程大小newFixedThreadExecutor();
3. 可缓存的线程池：不会限制线程的数newCachedThreadExecutor();
4. 支持周期的固定大小线程池：周期的执行操作newScheduledThreadExecutor();

并发中的概念：

1. 原子性：代码完整执行，不可被分割
2. 可见性：一个线程对对象进行操作。另一个线程就可以看到变化
3. 有序性：多线程中要避免指令重排

volatile可以保证的线程之间的可见性；禁止了指令重排；但是无法保证原子性

原子性可以使用同步代码块实现。