2016-09-28日工作日报

1. **已完成的工作内容**

* 遍历List集合:

- 方法1：通过for循环和get()方法配合实现遍历

- 方法2：通过迭代器Iterator实现遍历

- 所有集合接口和类都没有提供相应遍历方法，而是由Iterator实现集合遍历

- Collection 接口的iterate()方法返回一个Iterator，然后通过Iterator接口的两个方法可实现遍历

- boolean hasNext(): 判断是否存在另一个可访问的元素

- Object next(): 返回要访问的下一个元素

* 泛型<T>最常见的用途是创建集合类;- 泛型集合可以约束集合内的元素类型 ;- 典型泛型集合List<T>，HashMap<K,V>

- <T>、<K,V>表示该泛型集合中的元素类型.

* 线程是程序执行中的一个执行路径（子任务）。- 多线程是指程序中包含多条执行路径。也是指在一个进程中可同时执行两个或两个以上的线程。
* 进程： 是一个程序在其自身的地址空间中的一次执行活动，是程序独立运行的基本单位。
* Java线程的生命周期中一共有五个状态：新建状态 : 创建一个新的线程对象(new Thread)。当一个线程处于创建状态时，系统不为它分配资源。就绪状态: Java通过start方法启动处于新建状态的线程对象，使其进入就绪状态。处于就绪状态的线程已经具备了执行资格，将进入线程队列等待系统为其分配CPU，一旦获得了CPU，线程就进入运行状态，并调用自己的run方法。运行状态:处于就绪状态的线程被调度并获得CPU资源后即进运行状态，每一个Thread类及其子类的对象都有一个run()方法，当线程对象被调度执行的时候，它将自动调用本对象的run()方法。阻塞状态: 一个正在执行的线程如果在某些特殊情况下，如被人为挂起或它的CPU时间片耗尽时，将让出CPU并暂时中止自己的执行，进入阻塞状态。阻塞时它不能进入排列队列，只有当引起阻塞的原因被消除时，线程才可以转入就绪状态，重新进到线程队列中排队等待CPU资源，以便从原来终止处开始继续执行。终止状态;当线程的运行代码全部执行完毕时，线程就进入终止状态，当我们执行Java线程对象的stop()方法后线也进入终止状态。CPU再也不会为该线程分配执行时间了。
* Throwable: Error：通常出现重大问题如：运行的类不存在或者内存溢出等；不编写针对代码对其处理。Exception：在运行时函数出现的一些情况，可以通过try、catch、finally。
* 异常处理的格式:

try{

//需要检测的代码；

}catch(异常类 变量) {

//异常处理代码；

}catch(异常类 变量) {

//异常处理代码；}finally{

//一定会执行的代码；}

* final、finally、finalize的区别: final：如果一个变量或方法参数被final修饰，就表示它只能被赋值一次，但是JAVA虚拟机为变量设定的默认值不记作一次赋值。finally：只能跟在try/catch语句中，并且附带一个语句块，表示最后执行。如果父类抛出多个异常，那么重写(覆盖)方法必须抛出那些异常的一个子集，不能抛出新的异常； finalize：是gc操作的运行机制中得一部分。

1. **工作成果**
2. **未完成工作**
3. **未完成原因**
4. **遇到的问题及解决方案**