

咖啡妆

永久域名 <http://coffeef.javaeye.com>

[classloader](#) | [concurrent套件\(转载\)](#)



ericjoe

人穷不能志短...

2009-11-23 通过网页

>>更多闲聊

浏览: 17433 次

性别:

来自: 北京



详细资料

留言簿

搜索本博客

最近访客

>>更多

访客



yejinfei



lijiteng



snowway



mcg

博客分类

全部博客 (107)

生活 (5)

jbp (1)

2009

08 -
08

Callable与Future

关键字: callable与future

java.util.concurrent.Callable与java.util.concurrent.Future类可以协助您完成Future模式。Future模式在请求发生时, 会先产生一个Future对象给发出请求的客户。它的作用类似于代理(Proxy)对象, 而同时所代理的真正目标对象的生成是由一个新的线程持续进行。真正的目标对象生成之后, 将之设置到Future之中, 而当客户端真正需要目标对象时, 目标对象也已经准备好, 可以让客户提取使用。关于Future模式的说明, 可以参考:

<http://caterpillar.onlyfun.net/Gossip/DesignPattern/FuturePattern.htm>

Callable是一个接口, 与Runnable类似, 包含一个必须实现的方法, 可以启动为让另一个线程来执行。不过Callable工作完成后, 可以传回结果对象。Callable接口的定义如下:

```
public interface Callable<V> {
    V call() throws Exception;
}
```

例如, 可以使用Callable完成某个费时的工作, 工作结束后传回结果对象, 例如求质数范例15.24:

范例15.24 PrimeCallable.java

```
package onlyfun.caterpillar;

import java.util.ArrayList;
import java.util.List;
import java.util.concurrent.Callable;

public class PrimeCallable implements Callable<int[]> {
    private int max;

    public PrimeCallable(int max) {
        this.max = max;
    }

    public int[] call() throws Exception {
        int[] prime = new int[max+1];

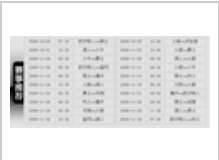
        List<Integer> list = new ArrayList<Integer>();

        for(int i = 2; i <= max; i++)
            prime[i] = 1;

        for(int i = 2; i*i <= max; i++) { // 这里可以改进
            if(prime[i] == 1) {
                for(int j = 2*i; j <= max; j++) {
                    if(j % i == 0)
                        prime[j] = 0;
                }
            }
        }
    }
}
```

- 非技术区 (4)
- 开源 (8)
- 其他技术区 (2)
- java (33)
- swing (0)
- questions_handler (4)
- eclipsercp (0)
- mysql (3)
- oracle (5)
- hibernate (14)
- spring (4)
- struts (6)
- linux (0)
- javaafx (1)
- wbservice (1)
- ibatis (3)
- dwr (2)
- javascript (1)
- 报表 (0)
- jsf (3)
- web (0)
- android (3)
- english (1)

我的相册



2009 nba 搜狐转播列表
共 22 张

我的留言簿 >>更多留言

- sdf
- by ericjoe
- 大哥，请问你能发一下JIVE的源代码我吗？谢谢你。
- email:36486491 ...
- by bqlin1987

其他分类

- 我的收藏 (28)

```
for(int i = 2; i < max; i++) {
    if(prime[i] == 1) {
        list.add(i);
    }
}

int[] p = new int[list.size()];
for(int i = 0; i < p.length; i++) {
    p[i] = list.get(i).intValue();
}

return p;
}
```

程序中的求质数方法是很简单的，但效率不好，这里只是为了示范方便，才使用简单的求质数方法，要更有效率地求质数，可以参考Eratosthenes筛选求质数：
<http://caterpillar.onlyfun.net/Gossip/AlgorithmGossip/EratosthenesPrime.htm>
假设现在求质数的需求是在启动PrimeCallable后的几秒之后，则可以使用Future来获得Callable执行的结果，从而在未来的时间点获得结果，例如范例15.25：
范例15.25 FutureDemo.java

```
package onlyfun.caterpillar;

import java.util.concurrent.Callable;
import java.util.concurrent.ExecutionException;
import java.util.concurrent.FutureTask;

public class FutureDemo {
    public static void main(String[] args) {
        Callable<int[]> primeCallable = new
        PrimeCallable(1000);
        FutureTask<int[]> primeTask =
        new FutureTask<int[]>(primeCallable);

        Thread t = new Thread(primeTask);
        t.start();

        try {
            // 假设现在做其他事情
            Thread.sleep(5000);

            // 回来看看质数找好了吗
            if(primeTask.isDone()) {
                int[] primes = primeTask.get();
                for(int prime : primes) {
                    System.out.print(prime + " ");
                }
                System.out.println();
            } catch (InterruptedException e) {
            } catch (ExecutionException e) {
                e.printStackTrace();
            }
        }
    }
}
```

java.util.concurrent.FutureTask是一个代理，真正执行找质数功能的是Callable对象。使用另一个线程启动FutureTask，之后就可以做其他的事了。等到某个时间点，用isDone()观察任务是否完成，如果完成了，就可以获得结果。一个执行结果如下，显示所有找到的质数：

```
2 3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59 61
67 71 73 79 83 89 97 101 103 107 109 113 127 131
137 139 149 151 157 163 167 173 179 181 191 193
```

- 我的书架 (1)
- 我的论坛主题贴 (1)
- 我的所有论坛贴 (12)
- 我的精华良好贴 (0)
- 我解决的问题 (3)

197	199	211	223	227	229	233	239	241	251	257	263
269	271	277	281	283	293	307	311	313	317	331	337
347	349	353	359	367	373	379	383	389	397	401	409
419	421	431	433	439	443	449	457	461	463	467	479
487	491	499	503	509	521	523	541	547	557	563	569
571	577	587	593	599	601	607	613	617	619	631	641
643	647	653	659	661	673	677	683	691	701	709	719
727	733	739	743	751	757	761	769	773	787	797	809
811	821	823	827	829	839	853	857	859	863	877	881
883	887	907	911	919	929	937	941	947	953	967	971
977	983	991	997								

最近加入圈子

- 英语学习
- Jquery
- javascript研究小组
- lucene爱好者
- 电脑DIY

存档

- 2009-10 (12)
- 2009-09 (8)
- 2009-08 (31)
- 更多存档...

最新评论

- java中调用js脚本
比如一段复杂的json数据,用json转java处理 还不如用js处理,并用这种方 ...
-- by huazhouji
- 非常主播，推荐大家去看看 ...
看了，还不错。这三代人都是音乐天才！据说这部电影创了韩国的收视奇迹？
-- by only_java
- 非常主播，推荐大家去看看 ...
一提韩国就让我联想到 同性恋 人妖 这些玩意
-- by xly_971223
- java中调用js脚本
这个东西，如果运用得到，也许会挺有用的
-- by 逍遥虾
- 非常主播，推荐大家去看看 ...
片子不怎么样 看了一半多点 韩国的主流价值观真的像里面演的一样么？ 那太可怕了 ...
-- by Enn

用户可能需要快速翻页浏览文件，但在浏览到有图片的页数时，由于图片文件很大，导致图片加载较慢，造成用户浏览文件时会有停顿的现象。因此希望在文件开启之后，仍有一个后台作业持续加载图片。这样，用户在快速浏览页面时，所造成的停顿可以获得改善，从而可以考虑使用这里所介绍的功能

22:40	浏览 (249)	评论 (0)	分类: java	相关推荐	classloader	+	concurrent套件(转载)
-------	----------	--------	----------	------	-------------	---	------------------

评论

发表评论

表情图标

字体颜色:

字体大小:

对齐:

提示: 选择您需要装饰的文字, 按上列按钮即可添加上相应的标签

😊

😏

😞

😜

😡

😏

😏

😏

😏

😏

😏

😏

😏

😏

😏

😏

😏

😏

😏

😏

😏

😏

😏

😏

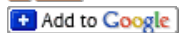
您还没有登录，请登录后发表评论(快捷键 Alt+S / Ctrl+Enter)

评论排行榜

- [java中调用js脚本](#)
- [非常主播，推荐大家去看看！](#)
- [每日一词 持续更新](#)
- [hibernate spring 事务](#)
- [andriod\(1\) 初学 安装 以及相关
问题](#)



RSS



[什么是RSS?]

声明：**JavaEye** 文章版权属于作者，受法律保护。没有作者书面许可不得转载。若作者同意转载，必须以超链接形式标明文章原始出处和作者。

© 2003-2010 JavaEye.com. All rights reserved. 上海炯耐计算机软件有限公司 [沪ICP备05023328号]

量子统计