**POCO C++库学习和分析 -- 任务**

**1. 任务的定义**

        任务虽然在Poco::Foundation库的目录结构中被单独划出，其实也可以被看成线程的应用，放在线程章节。首先来看一下Poco中对于任务的描述：

* task主要应用在GUI和Seerver程序中，用于追踪后台线程的进度。
* 应用Poco任务时，需要类Poco::Task和类Poco::TaskManager配合使用。其中类Poco::Task继承自Poco::Runnable，它提供了接口可以便利的报告线程进度。Poco::TaskManager则对Poco::Task进行管理。
* 为了完成取消和上报线程进度的工作：

                  a. 使用者必须从Poco::Task创建一个子类并重写runTask()函数  
                  b. 为了完成进度上报的功能，在子类的runTask()函数中，必须周期的调用setProgress()函数去上报信息  
                  c. 为了能够在任务运行时终止任务，必须在子类的runTask()函数中，周期性的调用isCancelled()或者sleep()函数，去检查是否有任务停止请求  
                  d. 如果isCancelled()或者sleep()返回真，runTask()返回。

* Poco::TaskManager通过使用Poco::NotificationCenter 去通知所有需要接受任务消息的对象

        从上面描述可以看出，Poco中Task的功能就是能够自动汇报线程运行进度。

**2. 任务用例**

        Task的应用非常简单，下面是其一个使用例子：

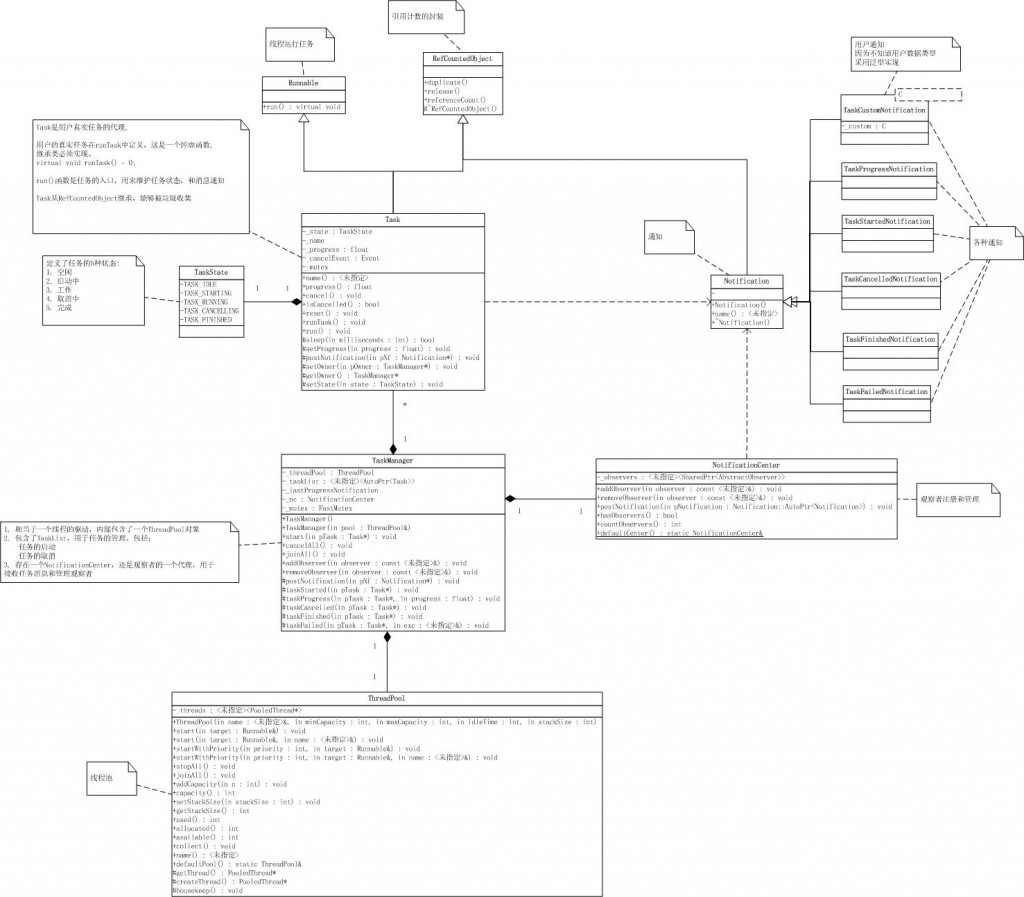
**[cpp]** [view plaincopy](http://blog.csdn.net/arau_sh/article/details/8620810)

1. #include "Poco/Task.h"
2. #include "Poco/TaskManager.h"
3. #include "Poco/TaskNotification.h"
4. #include "Poco/Observer.h"
6. **using** Poco::Observer;
7. **class** SampleTask: **public** Poco::Task
8. {
9. **public**:
10. SampleTask(**const** std::string& name): Task(name)
11. {}

14. **void** runTask()
15. {
16. **for** (**int** i = 0; i < 100; ++i)
17. {
18. setProgress(**float**(i)/100); // report progress
19. **if** (sleep(1000))
20. **break**;
21. }
22. }
23. };
25. **class** ProgressHandler
26. {
27. **public**:
28. **void** onProgress(Poco::TaskProgressNotification\* pNf)
29. {
30. std::cout << pNf->task()->name()
31. << " progress: " << pNf->progress() << std::endl;
32. pNf->release();
33. }
34. **void** onFinished(Poco::TaskFinishedNotification\* pNf)
35. {
36. std::cout << pNf->task()->name() << " finished." << std::endl;
37. pNf->release();
38. }
39. };
41. **int** main(**int** argc, **char**\*\* argv)
42. {
43. Poco::TaskManager **tm**;
44. ProgressHandler pm;
45. **tm**.addObserver(
46. Observer<ProgressHandler, Poco::TaskProgressNotification>
47. (pm, &ProgressHandler::onProgress)
48. );
49. **tm**.addObserver(
50. Observer<ProgressHandler, Poco::TaskFinishedNotification>
51. (pm, &ProgressHandler::onFinished)
52. );
53. **tm**.start(**new** SampleTask("Task 1")); // tm takes ownership
54. **tm**.start(**new** SampleTask("Task 2"));
55. **tm**.joinAll();
56. **return** 0;
57. }

**3. Task类图**

        最后给出Poco中Task的类图。



        在类图中，我们可以看到Task类继承自RefCountedObject，这主要是为了Task类的管理。TaskManger对Task实现了自动的垃圾收集。

（版权所有，转载时请注明作者和出处   <http://blog.csdn.net/arau_sh/article/details/8620810>）