软件复用

Performance Management

撰写人：刘蕊

2015-4-27

目录

1. 引言

1.1编写目的

该组件为软件复用课程的课程项目，撰写此文档的目的在于使得用户能够更好的复用该组件。

2. 组件简介

2.1 功能

名称：PM
开发者：韦吾境
功能：这一组件用来接受应用程序的性能指标，每分钟自动生成性能报告，对每个指标求和，性能报告输出到单独文件（报告文件名为：yyyy-mm-dd_hh-mm.log）

2.2 输入

应用程序的性能指标

2.3 输出

每分钟输出性能报告，内容是性能指标之和，输出到 yyyy-mm-dd_hh-mm.log

2.4 参数

sendMsg()：接收程序的性能指标时，检查此次时间与上次接收时间间隔是否在一分钟内，若在一分钟内，就对指标求和，若不在一分钟内，则将性能指标重置为零重新进行累加。
getDateTime()：获取当前时间
printLog()：生成性能报告到文件中，文件名为 yyyy-mm-dd_hh-mm.log

2.5 源码

```
public class PM
{
    private static class SingletonHolder
    {
        public final static Map<String, Integer> instance = new HashMap<String, Integer>();
        public static String prevDateTime = getDateTime(-1);
    }

    // refer to:
    // https://github.com/TJSoftwareReuse/2012T08/blob/master/PM/src/com/team8/PerformanceManagement/PM.java
    public static String getDateTime(int moreMinute)
    {
        Calendar calendar = Calendar.getInstance();
        calendar.add(Calendar.MINUTE, moreMinute);
        SimpleDateFormat dateFormat = new SimpleDateFormat("yyyy-MM-dd_HH-mm");

        return dateFormat.format(calendar.getTime());
    }


    public synchronized static void sendMsg(String key, int value)
    {
        String dateTime = getDateTime(0);
        if (!dateTime.equals(SingletonHolder.prevDateTime))
        {
            SingletonHolder.instance.clear();
            SingletonHolder.prevDateTime = dateTime;
        }

        if(SingletonHolder.instance.containsKey(key))
            SingletonHolder.instance.put(key, value + SingletonHolder.instance.get(key));
        else {
            SingletonHolder.instance.put(key, value);
        }

        printLog(dateTime);
    }


    public static synchronized void printLog(String dateTime)
    {
        try{
            PrintWriter writer = new PrintWriter(dateTime + ".log", "UTF-8");
            writer.println("PM Report of " + dateTime);
            for (String key: (SingletonHolder.instance).keySet())
            {
                writer.println(key + " : " + SingletonHolder.instance.get(key).toString());
            }
            writer.close();
        }catch (IOException e){
            e.printStackTrace();
        }
    }
}
```

3 使用方法

以下是样例程序：

```
public class Main {
    public static void main(String[] args) {
        SimpleTest();
    }

    // a simple test
    public static void SimpleTest()
    {
        testInstance ti0 = new testInstance("ringo", 1, 100, 1000);
        testInstance ti1 = new testInstance("paul", 1, 40, 2000);
        testInstance ti2 = new testInstance("harrison", 1, 60, 1500);
        testInstance ti3 = new testInstance("lennon", 1, 75, 1000);

        System.out.println("Test start.");
        ti0.start();
        ti1.start();
        ti2.start();
        ti3.start();
    }


    private static class testInstance extends Thread
    {
        private int internal;
        private int count;
        String key;
        int value;

        private testInstance(String key, int value, int count, int internal)
        {
            this.key = key;
            this.value = value;
            this.count = count;
            this.internal = internal;
        }

        @Override
        public void run()
        {
            while(count-- > 0)
            {
                try {
                    sleep(internal);
                } catch (InterruptedException e) {
                    e.printStackTrace();
                }
                PM.sendMsg(key, value);
                System.out.println(key + " : " + value + "    sent.");
            }

        }
    }
}
```