

第六次作业参考答案

第一题：

```
import java.text.*;
import java.util.Date;

public class FormatTest
{
    public static void main(String[] args)
    {
        SimpleDateFormat sdf=new SimpleDateFormat("[yyyy/MM/dd][HH:mm:ss]");

        while(true)
        {
            System.out.println(sdf.format(new Date()));
            try
            {
                Thread.sleep(1000);
            } catch (InterruptedException e)
            {
                e.printStackTrace();
            }
        }
    }
}
```

第二题

2. (1) 使线程进入睡眠状态 200ms
(2) 发生异常抛出
(3) 实例化线程类
(4) 等待其他线程运行结束

3. 提示：缺少同步，造成多个线程对共享变量操作非有序性，导致最终结果不一致。在 `getNumber` 和 `add` 中加入同步关键字即可。

第三题

提示：使用 TCP (Socket) 获取指定网页文本，再使用正则表达式匹配即可。通常情况下，网站的端口为 80。

第五题

这个问题是个比较复杂的问题，涉及到 Java 内存模型，喜欢钻研的同学可以认真研究一下。

`i++`和`++i` 不是原子操作。可以使用多线程对共享变量进行这两种操作进而验证，但是要注意以下的关键点。

在 Java 内存模型中，每个线程会有自己的运行栈（不懂的同学去研究操作系统），栈中会包含一个或多个方法帧（帧中包含操作栈，动态连接，局部参数区，返回地址...），在涉及到共享变量时，线程会将堆（对象的主要保存位置）中的该共享变量的值拷贝到本地进行操作，便于提高运行速度，操作完成后将值写回。这样当涉及到多个线程时，**各个线程的推进的速度是不一样的，各个线程也不能准确的观察到共享变量的变化**，即共享变量值发生变化没有被立刻被反映出来，这样也就无法真正验证 `i++`和`++i` 是不是原子操作这个问题。**这个问题可以通过使用 `volatile` 关键字对共享变量进行修饰，但是要注意该关键字语义是一个线程对共享变量修改后，该共享变量的变化对其他线程立即可见。**

`volatile` 还有另外一种语义：

禁止指令重排（涉及到计算机的组成原理的 CPU 指令乱序发射问题），有兴趣的同学可以研究一下。