

Chapter 12 多线程

Key Point :

- 线程的概念
- 线程的创建
- 线程的状态转换
- 线程间数据共享
- 线程的同步

答案：

1. 参考答案：

一个单 CPU 在同时处理多个线程时，遵循“宏观上并行，微观串行”。

2. C

run 方法可以直接调用，但是直接调用 run 方法并不产生新线程。

3. C

Runnable 接口中的 run 方法没有参数。

4. A

Thread.sleep()方法会抛出已检查异常 java.lang.InterruptedException，必须要处理。

5. 参考：TestThread.java

6. 参考答案：

加上 synchronized 表示对 lock 对象加锁，这样能保证输出 10 个\$\$\$之后再输出 10 个###。如果不加上 synchronized 的话，则程序运行时输出的顺序每次执行都不同。

7. C

两个线程都是对属性 data 加锁，然而，t1 对象的 data 属性是字符串对象“hello”，t2 对象的 data 属性是字符串对象“world”，因此不是同一个对象，因此 t1 线程和 t2 线程之间没有交互和加锁、解锁的操作，因此不会有同步的效果。

8. C

- I. A 错误，因为对 run 方法表示对当前对象加锁，具体到这个例子，分别表示对 t1 对象和 t2 对象加锁。由于加锁不是同一个对象，因此不会产生同步的效果。
- II. B 错误，原因如上一题所说。

9. 参考答案：

```
class MyThread1 implements Runnable{  
    public void run() {  
        for(int i = 0; i<100; i++){  
            this.sleep((int)(Math.random()*1000));  
            System.out.println( "hello" );  
        }  
    }  
}
```

```
class MyThread2 extends Thread{

    public void run() throws Exception { // run 方法不能抛出任何异常

        for(int i = 0; i<100; i++){

            this.sleep((int)(Math.random()*1000));

            System.out.println( "world" );

        }

    }

}

public class TestMyThread{

    public static void main(String args[]){

        Runnable t1 = new MyThread1();

        Thread t2 = new MyThread2();

        t1.start(); //不能对 t1 调用 start 方法 , 应该利用 t1 创建一个 Thread 对象

        t2.start();

    }

}
```