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

问题：

1. 一个单 CPU 的机器，如何同时执行多个线程？请简述其原理。

2. （线程的创建）有以下代码

```
public class Example implements Runnable {
    public void run() {
        while(true) {
        }
    }
    public static void main(String args[]) {
        Example ex1 = new Example();
        Example ex2 = new Example();
        Example ex3 = new Example();
        ex1.run();
```

```
            ex2.run();
            ex3.run();
        }
    }
```

选择正确答案：

A. 代码编译失败，因为 ex2.run()无法获得执行

B. 代码编译成功，存在 3 个可运行的线程

C. 代码编译成功，存在 1 个可运行的线程

3. （线程的创建）有以下代码

```
class Example implements Runnable {
    public static void main(String args[]) {
        Thread t = new Thread(new Example());
        t.start();
    }
    public void run(int limit) {
        for (int x = 0; x<limit; x++) {
            System.out.println(x);
        }
    }
}
```

选择正确答案：

A. 打印输出，从 0 至 limit

B. 无内容输出，因为没有明确调用 run()方法。

C. 代码编译失败，因为没有正确实现 Runnable 接口

D. 代码编译失败，如果声明类为抽象类，可使代码编译成功。

E. 代码编译失败，如果去掉 implements Runnable，可使代码编译成功。

4. （sleep 方法）有如下代码

```
class Example {
    public static void main(String args[]) {
        Thread.sleep(3000);
        System.out.println( “sleep” );
    }
}
```

选择正确答案：

A. 编译出错

B. 运行时异常

C. 正常编译运行，输出 sleep

D. 正常编译运行，但没有内容输出

5. （线程的创建）创建两个线程，要求如下：

I. 一个线程输出 100 个 1~26，另一个线程输出 100 个 A~Z

II. 一个线程使用继承 Thread 类的写法，另一个线程使用实现 Runnable 接口的写法。

6. （线程的同步）有如下代码

```
class MyThread1 extends Thread{
    Object lock;
```

```
    public MyThread1(Object lock){
        this.lock = lock;
    }
    public void run(){
        synchronized(lock){ //1
            for(int i = 0; i<=10; i++){
            try{
                Thread.sleep( (int)(Math.random()*1000) );
            }catch(Exception e){}
                System.out.println( “$$$” );
            }
        }
    }
}
class MyThread2 extends Thread{
    Object lock;
    public MyThread2(Object lock){
        this.lock = lock;
    }
    public void run(){
        synchronized(lock){ //2
            for(int i = 0; i<=10; i++){
                try{
                    Thread.sleep((int)(Math.random()*1000) );
```

```
                    }catch(Exception e){
                    }
                    System.out.println(“###”);
                }
            }
        }
    }
    public class TestMyThread{
        public static void main(String args[]){
            Object lock = new Object();
            Thread t1 = new MyThread1(lock);
            Thread t2 = new MyThread2(lock);
            t1.start();
            t2.start();
        }
    }
```

问：在//1 和//2 处加上的 synchronized 起什么作用？如果不加 synchronized，运行程序有什么不同的地方？

7. （线程同步）有如下代码

```
class MyThread extends Thread{
    private String data;
    public void run(){
        synchronized(data){
```

```
                for(int i = 0; i<10; i++){
                    try{
                        Thread.sleep((int)(Math.random()*1000));
                    }catch(Exception e){
                    }
                    System.out.println(data);
                }
            }
        }
    }
    public class TestMyThread {
        public static void main(String args[]){
            Thread t1 = new MyThread( “hello” );
            Thread t2 = new MyThread( “world” );
            t1.start();
            t2.start();
        }
    }
```

问：上述代码输出的结果是什么？

A. 先输出 100 个 hello，然后是 100 个 world

B. 先输出 100 个 world，然后是 100 个 hello

C. 线程不同步，因此交替输出 hello 和 world

8. （线程同步）有下面代码

```
1)      class MyThread extends Thread{
2)          private String data;
3)          public MyThread(String data){
4)              this.data = data;
5)          }
6)          public void run(){
7)              for(int i = 0; i<100; i++){
8)                  System.out.println(data);
9)              }
10)         }
11)         }
12)         public class TestMyThread{
13)             public static void main(String args[]){
14)                 Thread t1 = new MyThread(“aaa”);
15)                 Thread t2 = new MyThread(“bbb”);
16)                 t1.start();
17)                 t2.start();
18)             }
19)         }
```

现希望能够同步的输出 aaa 和 bbb，即一次输出 100 个 aaa 或 bbb，输出这两个字符串时没有交互。

为了达到上述目的，要对原代码进行修改。以下哪些修改方式能够得到想要的结果？

A. 把第 6 行改为 public synchronized void run()

B. 把 run 方法中所有的内容都放在 synchronized(data)代码块中

C. 把 run 方法中所有的内容都放在 synchronized(System.out)代码块中

9. （线程综合）代码改错

```
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 {
        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();
        t2.start();
```

```
    }

}
```