

作业 8-2 实现 Runnable 接口创建多线程

一、案例描述

1、考核知识点

名 称：实现 Runnable 接口创建多线程

2、练习目标

掌握如何通过实现 Runnable 接口方式创建多线程。

➤ 掌握如何使用 Thread 类的有参构造方法创建 Thread 对象。

3、需求分析

在 Java 中只支持单继承，因此通过继承 Thread 类创建线程有一定的局限性，这时可以使用另一种方式，即实现 Runnable 接口来创建线程。通过这种方式需要在 Thread(Runnable target)的构造方法中，传递一个实现了 Runnable 接口的实例对象。接下来在案例中将通过实现 Runnable 接口方式创建线程，并实现多线程分别打印 0~99 的数字的功能。

4、设计思路（实现原理）

- 1) 自定义一个类 Demo，使其实现 Runnable 接口。
- 2) 在 Demo 类中覆写 run()方法，在方法编写一个 for 循环，循环体内打印：当前线程名称：+当前循环次数。
- 3) 编写测试类 Example02，在 Example02 类的 main()方法中，创建一个 Demo 对象，利用 Thread (Runnable target) 构造方法创建 2 个线程对象，分别命名为“蜘蛛侠”和“钢铁侠”，并执行线程对象的 start()方法，同时编写 for 循环，循环内打印“main: ”+当前循环次数。

二、案例实现

```
class Demo implements Runnable {
    public void run() {
        for(int x=0; x<100; x++){
            System.out.println(Thread.currentThread().getName()+"："+x);
        }
    }
}

public class Example02 {
    public static void main(String[] args) {
        Demo d = new Demo();
        Thread t1 = new Thread(d,"蜘蛛侠");
        Thread t2 = new Thread(d,"钢铁侠");
        t1.start();
        t2.start();
        for (int x = 0; x < 100; x++) {
            System.out.println("main:" + x);
        }
    }
}
```

运行结果如图 10-2 所示。

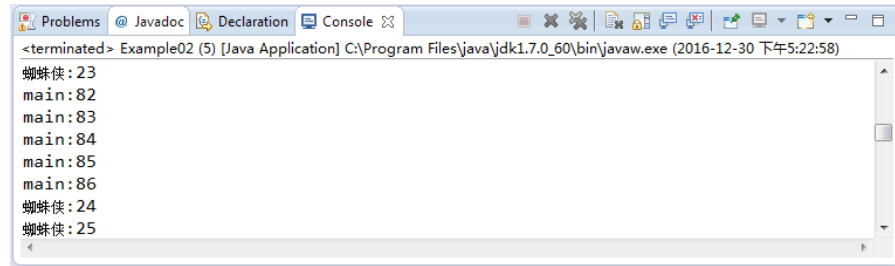


图10-1 运行结果

三、案例总结

1、可以把实现了 `Runnable` 接口并重写 `run()` 方法的实例对象，作为 `Thread` 有参构造方法的参数来创建多线程程序。

2、使用 `Thread` 类的构造方法 `Thread(Runnable target, String name)` 创建线程对象时，还可以给线程指定新名称。

3、思考一下：既然有了继承 `Thread` 类的方式，为什么还要有实现 `Runnable` 接口的方式？

- a) 可以避免由于 Java 的单继承带来的局限性。在开发中经常碰到这样一种情况，就是使用一个已经继承了某一个类的子类创建线程，由于一个类不能同时有两个父类，所以不能用继承 `Thread` 类的方式，那么就只能采用实现 `Runnable` 接口的方式。
- b) 实现接口的方式，适合多个相同程序代码的线程去处理同一个资源的情况，把线程同程序代码、数据有效的分离，很好的体现了面向对象的设计思想。例如：一个售票程序继承了 `Thread` 类，在售票时启动了多个售票程序，但他们不是同一个对象，数据没有共享，这样就会出现票数重复出售的情况；而当售票程序实现 `Runnable` 接口后，多个线程运行同一个售票程序，实现了票数共享的好处。