

廈門大學



信息学院软件工程系

《JAVA 程序设计》实验报告

实验 13

姓名：黄勛

学号：22920212204392

学院：信息学院

专业：软件工程

完成时间：2023.5.23

一、实验目的及要求

- 熟悉多线程编程

二、实验题目及实现过程

实验环境：Windows 10 21H2、jdk17、javafx scene builder、utf-8 编码

题目一

(一) 实验题目

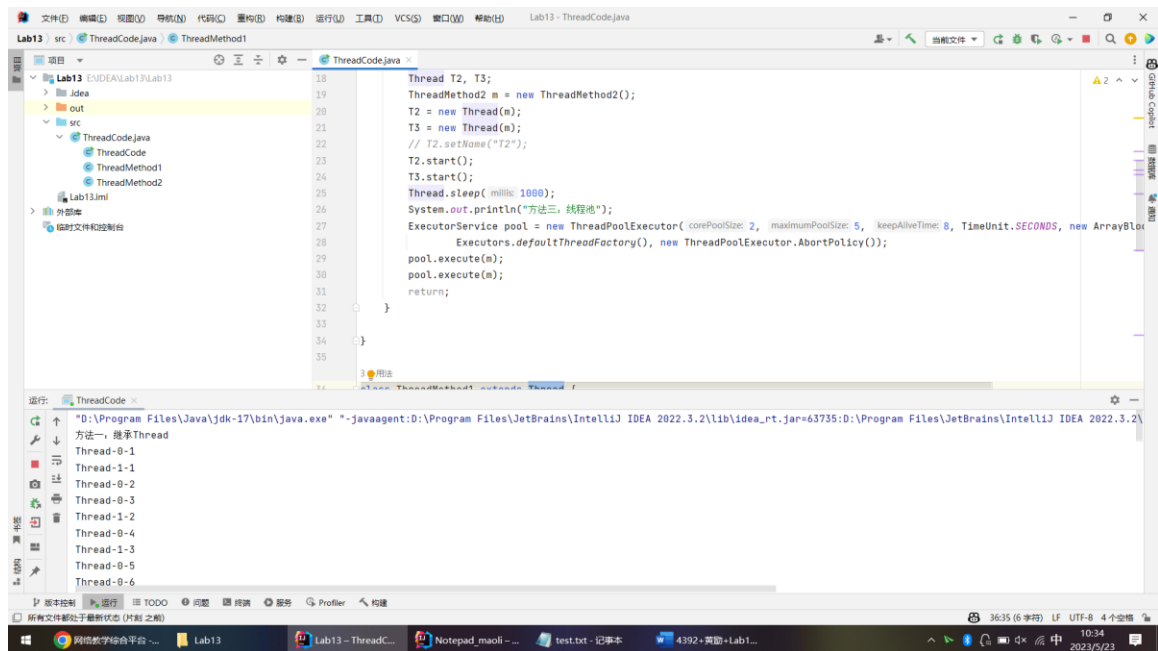
- ◆ 请用三种方法完成：2 个线程同时循环输出“线程名-数字”，其中数字从 1 变化到 10。
- ◆ 提示：(1) 继承 Thread (2) 实现 Runnable 接口，用 Thread 方法启动线程
- ◆ (3) 用线程池管理

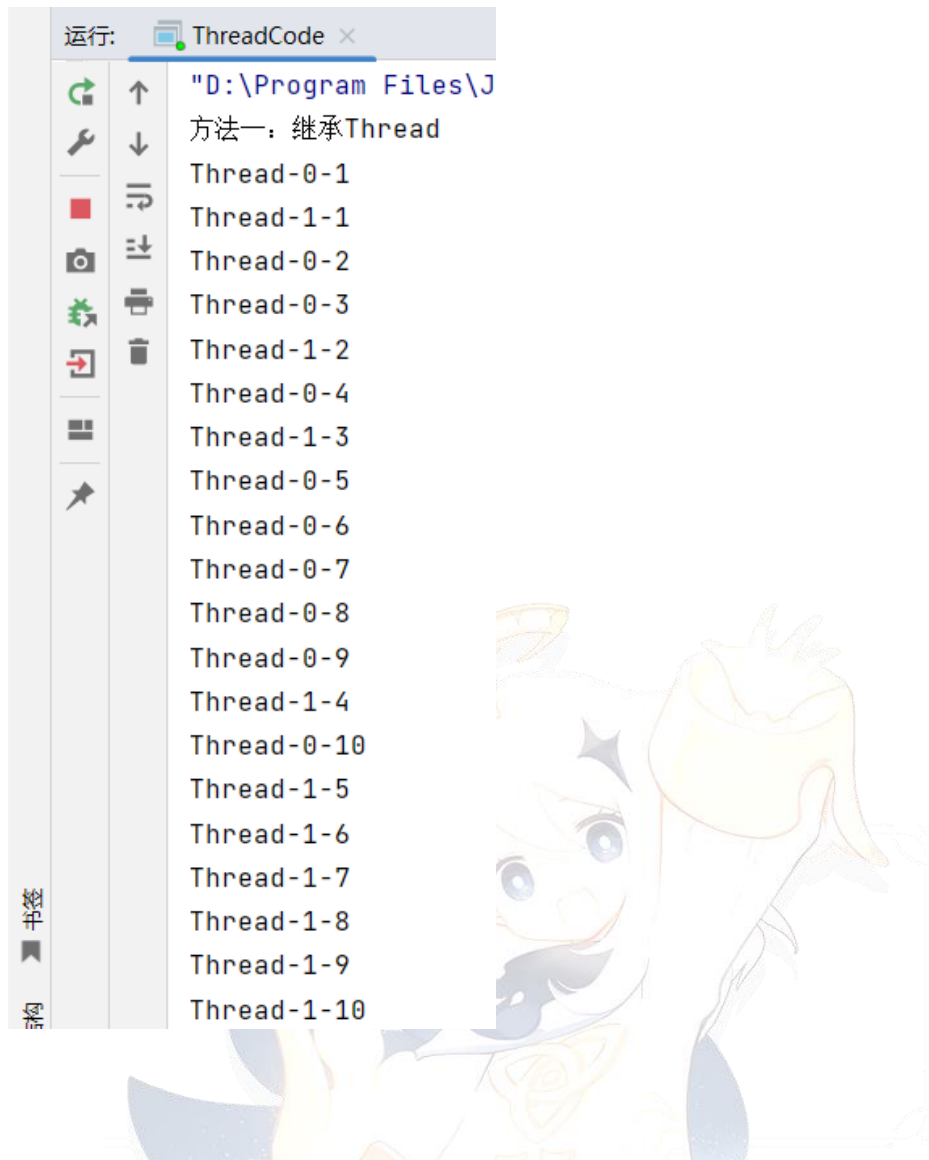
(二) 实现过程

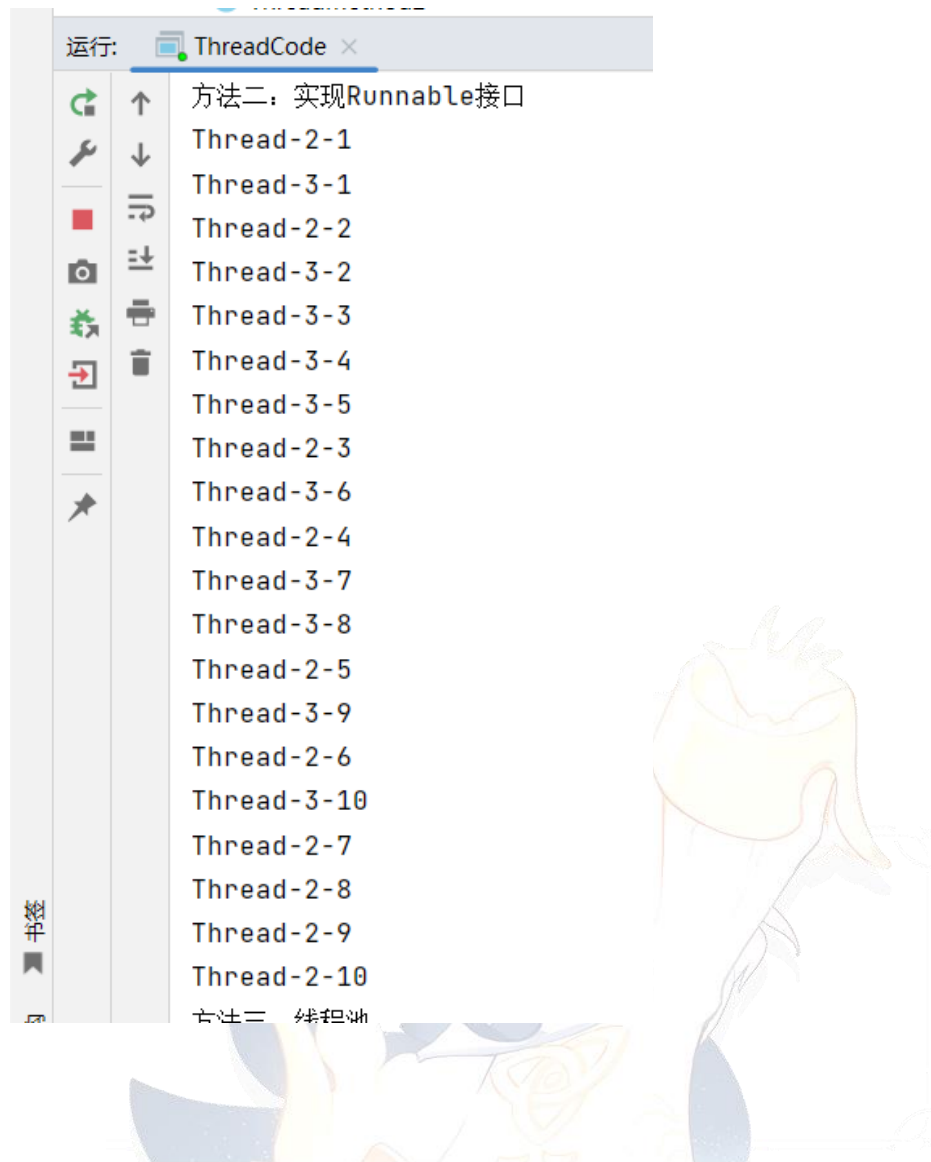
思路：首先让用设计了四个类，ThreadCode 类包含 main 方法是主线程，首先通过继承了 Thread 类的 ThreadMethod1 创建两个新线程，ThreadMethod1 里重写了 run 方法，实现循环输出“线程名-数字”，通过 Thread.currentThread.getName 方法获取线程名字。第二个方法是 ThreadMethod2 类实现 Runnable 接口并重写 run 方法，主线程使用 ThreadMethod2 对象创建两个新的线程。第三个方法是使用 ThreadPoolExecutor 创建线程池，通过 execute 方法实现 ThreadMethod2 对象的任务。

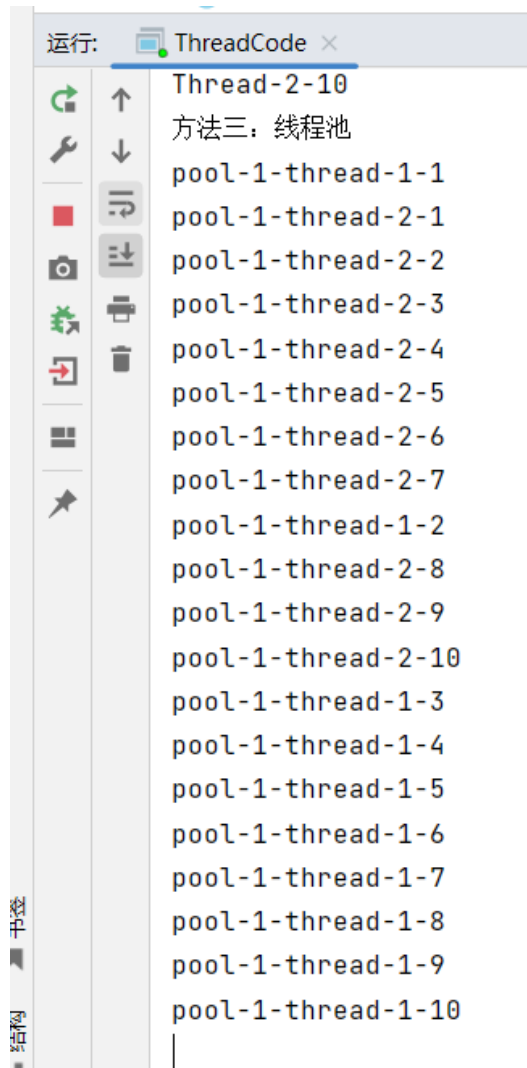
(三) 过程截图

最终结果（全屏截图）









部分代码展示

```

9  ▶ public static void main(String args[]) throws InterruptedException {
10      System.out.println("方法一: 继承Thread");
11      ThreadMethod1 T0, T1;
12      T0 = new ThreadMethod1();
13      T1 = new ThreadMethod1();
14      T0.start();
15      T1.start();
16      Thread.sleep( millis: 1000);
17      System.out.println("方法二: 实现Runnable接口");
18      Thread T2, T3;
19      ThreadMethod2 m = new ThreadMethod2();
20      T2 = new Thread(m);
21      T3 = new Thread(m);
22      T2.start();
23      T3.start();
24      Thread.sleep( millis: 1000);
25      System.out.println("方法三: 线程池");
26      ExecutorService pool = new ThreadPoolExecutor( corePoolSize: 2, maximumPoolSize: 5, keep
27          Executors.defaultThreadFactory(), new ThreadPoolExecutor.AbortPolicy());
28      pool.execute(m);
29      pool.execute(m);
30      return;
31  }

```

3 个用法

```

35  class ThreadMethod1 extends Thread {
36      /*
37       * TreadMethod1(String s){ setName(s); }
38       */
39      public void run() {
40          for (int i = 1; i <= 10; i++) {
41              System.out.println(getName() + "-" + i);
42          }
43      }
44  }

```

2 个用法

```

47  class ThreadMethod2 implements Runnable {
48
49      @Override
50      public void run() {
51          for (int i = 1; i <= 10; i++) {
52              System.out.println(Thread.currentThread().getName() + "-" + i);
53          }
54      }
55  }
56

```

三、实验总结与心得记录

通过本次实验，我对 Java 使用多线程编程有了更深入的了解。我学会了如何构建一个线程等操作，同时，我也学会了对进程的相关操作，实现了多次输出。在实验的过程中我为线程编写了对应的方法，实现了需要的功能，这对我来说收获颇丰。

ThreadPoolExecutor 方法的参数如下

参数一：指定线程池的线程数量（核心线程）：	<code>corePoolSize</code>	→	不能小于0
参数二：指定线程池可支持的最大线程数：	<code>maximumPoolSize</code>	→	最大数量 \geq 核心线程数量
参数三：指定临时线程的最大存活时间：	<code>keepAliveTime</code>	→	不能小于0
参数四：指定存活时间的单位(秒、分、时、天)：	<code>unit</code>	→	时间单位
参数五：指定任务队列：	<code>workQueue</code>	→	不能为null
参数六：指定用哪个线程工厂创建线程：	<code>threadFactory</code>	→	不能为null
参数七：指定线程忙，任务满的时候，新任务来了怎么办：	<code>handler</code>	→	不能为null

GDN @time@Friend

