

1. 训练案例1

1.1. 需求说明

1. 请描述什么是线程池。
2. 线程池的优点

2. 训练案例2

2.1 需求说明

1. 请列举Lambda语法的省略规则。
2. 请描述Lambda的使用前提。

3. 训练案例3

3.1. 需求说明

1. 使用线程池,分别开启三条线程,生成10个1到100的随机数,然后计算他们的平均数,最后计算这三个平均数的平均数.(使用Callable)

4. 训练案例4

4.1. 需求说明

1. 使用线程池：一个线程计算10!，一个线程计算5!，一个线程计算8! 打印在那个线程执行的和执行结果.(使用Runnable)

5. 训练案例5

5.1. 需求说明

1. 给出以下代码，请问该程序的运行结果是什么? 如有问题，请说明原因。

```
public class Test07 implements Runnable {  
    public static void main(String[] args) {  
        Thread t = new Thread(new Test07());  
        t.start();  
    }  
    public void run(int num) {  
        for (int i = 0; i < num; i++) {  
            System.out.println(i);  
        }  
    }  
}
```

6. 训练案例6

6.1. 需求说明

1. 给定一个导演 Director接口，内含唯一的抽象方法 makeMovie，且无参数、无返回值。

```
public interface Director {  
    void makeMovie();  
}
```

2. 在下面的代码中，请分别使用Lambda的标准格式及省略格式调用 invokeDirect 方法，打印输出：导演拍电影啦！

```
public class TestInvokeDirect {  
    public static void main(String[] args) {  
        // TODO 请分别使用Lambda【标准格式】及【省略格式】调用invokeDirect方法•  
    }  
    private static void invokeDirect(Director director) {  
        director.makeMovie();  
    }  
}
```

7. 训练案例7

7.1. 需求说明

1. 给定一个计算器 Calculator 接口，内含抽象方法 calc(减法)，其功能是可以将两个数字进行相减，并返回差值。

```
public interface Calculator {  
    int calc(int a, int b);  
}
```

2. 在下面的代码中，请分别使用Lambda的标准格式及省略格式调用 invokeCalc 方法，完成130和120的相减计算：

```
public class TestInvokeCalc {  
    public static void main(String[] args) {  
        // TODO 请分别使用Lambda【标准格式】及【省略格式】调用invokeCalc方法来计算130-120的结果  
    }  
    private static void invokeCalc(int a, int b, Calculator calculator) {  
        int result = calculator.calc(a, b);  
        System.out.println("结果是: " + result);  
    }  
}
```

8. 训练案例8

8.1. 需求说明

1. 对Java基础班学生的考试成绩进行排序
 - 定义一个学生类，有两个属性。姓名，考试成绩
- 在测试类中定义一个学生数组，存储学生信息。
- 使用Lambda省略格式写法对学生数组按照考试成绩由高到低进行排序。

9. 训练案例9

9.1. 需求说明

1. 使用lambda表达式对List，Set，Map集合进行遍历。

10. 训练案例10

10.1. 需求说明

1. 使用等待唤醒,实现一个线程给共享对象Person的属性(姓名,性别)赋值,一个线程打印共享对象的属性要求赋值->打印->赋值->打印 ... 不允许出现人妖.

11. 训练案例11

11.1. 训练描述

某公司组织年会，会议入场时有两个入口，在入场时每位员工都能获取一张双色球彩票，假设公司有100个员工，利用多线程模拟年会入场过程，并分别统计每个入口入场的人数,以及每个员工拿到的彩票的号码。线程运行后打印格式如下：

```
编号为： 2 的员工 从后门 入场！ 拿到的双色球彩票号码是:[17, 24, 29, 30, 31, 32, 07]
编号为： 1 的员工 从后门 入场！ 拿到的双色球彩票号码是:[06, 11, 14, 22, 29, 32, 15]
//.....
从后门入场的员工总共： 13 位员工
从前门入场的员工总共： 87 位员工
```

题目中用到的产生双色球的工具类已经写好,可以直接使用

```
public class DoubleColorBallUtil {
    // 产生双色球的代码
    public static String create() {
        String[] red = {"01", "02", "03", "04", "05", "06", "07", "08", "09", "10",
            "11", "12", "13", "14", "15", "16", "17", "18", "19", "20", "21", "22", "23",
            "24", "25", "26", "27", "28", "29", "30", "31", "32", "33"};
        //创建蓝球
        String[] blue = "01,02,03,04,05,06,07,08,09,10,11,12,13,14,15,16".split(",");
        boolean[] used = new boolean[red.length];
```

```
Random r = new Random();
String[] all = new String[7];
for(int i = 0; i < 6; i++) {
    int idx;
    do {
        idx = r.nextInt(red.length); // 0-32
    } while (used[idx]); // 如果使用了继续找下一个
    used[idx] = true; // 标记使用了
    all[i] = red[idx]; // 取出一个未使用的红球
}
all[all.length-1] = blue[r.nextInt(blue.length)];
Arrays.sort(all);
return Arrays.toString(all);
}
```