

多线程

一、概述

- **线程(Thread)**: 是一个程序内部的一条执行流程。
- **单线程程序**: 如果一个程序内部只有一条执行流程，则称这个程序是一个单线程程序。
- **多线程**: 是指从软硬件上实现的多条执行流程的技术（多条线程由CPU负责调度执行）。

二、创建多线程

Important

`java.lang.Thread` 类代表线程。

2.1 创建方式1：继承 `Thread` 类

P1 步骤

1. 创建一个子类并继承 `Thread` 类；
2. 重写 `run()` 方法；
3. 创建子类的实例对象；
4. 调用线程对象的 `start()` 方法启动线程，`start()` 方法内部会去调用子类重写的 `run()` 方法。

P2 代码演示

- `ThreadTest1` 类：

```
1 package create_thread;
2
```

```
3 public class ThreadTest1 {  
4     // main方法是由主线程负责执行的  
5     public static void main(String[] args) {  
6  
7         Thread t = new MyThread();  
8         // 启动后，实现了多线程：主线程 + t线程  
9         t.start(); // start会去调用子类重写的run方法  
10  
11        for (int i = 0; i < 10; i++) {  
12            System.out.println("Main Thread output: " + i);  
13        }  
14    }  
15 }
```

- **MyThread 类：**

```
1 package create_thread;  
2  
3 /**  
4  * 1.让子类继承Thread类  
5  * 2.重写Thread类的run方法  
6  */  
7 public class MyThread extends Thread {  
8     @Override  
9     public void run() {  
10         // 描述线程的执行任务  
11         for (int i = 0; i < 10; i++) {  
12             System.out.println("child--My Thread output: " + i);  
13         }  
14     }  
15 }
```

- 控制台输出（每次输出结果都不同，仅截取其中某一次）：

```
D:\CS-Softwares\JDK\JDK21\bin\java.exe "-javaagent:D:\CS-Softwares\IntelliJ IDEA 2021.3.1\lib\idea_rt.jar" -Dfile.encoding=UTF-8 -classpath "D:\CS-Softwares\Java\MyThread\MyThread.jar" com.imooc.threads.MyThread
child--My Thread output: 0
child--My Thread output: 1
Main Thread output: 0
Main Thread output: 1
Main Thread output: 2
Main Thread output: 3
Main Thread output: 4
Main Thread output: 5
Main Thread output: 6
Main Thread output: 7
Main Thread output: 8
child--My Thread output: 2
Main Thread output: 9
child--My Thread output: 3
child--My Thread output: 4
child--My Thread output: 5
child--My Thread output: 6
child--My Thread output: 7
child--My Thread output: 8
child--My Thread output: 9

Process finished with exit code 0
```

P3 优缺点

- **优点：**编码简单。
- **缺点：**子线程类已经继承了 `Thread` 类，无法再继承其他类，不利于后续功能扩展。

2.2 创建方式2：