

# JAVA 预热班讲义

讲师：毕向东

## 多线程

---

- 进程
- 线程（例：FlashGet）
- 多线程存在的意义。
- 线程的创建方式
- 多线程的特性

## 创建线程方式一

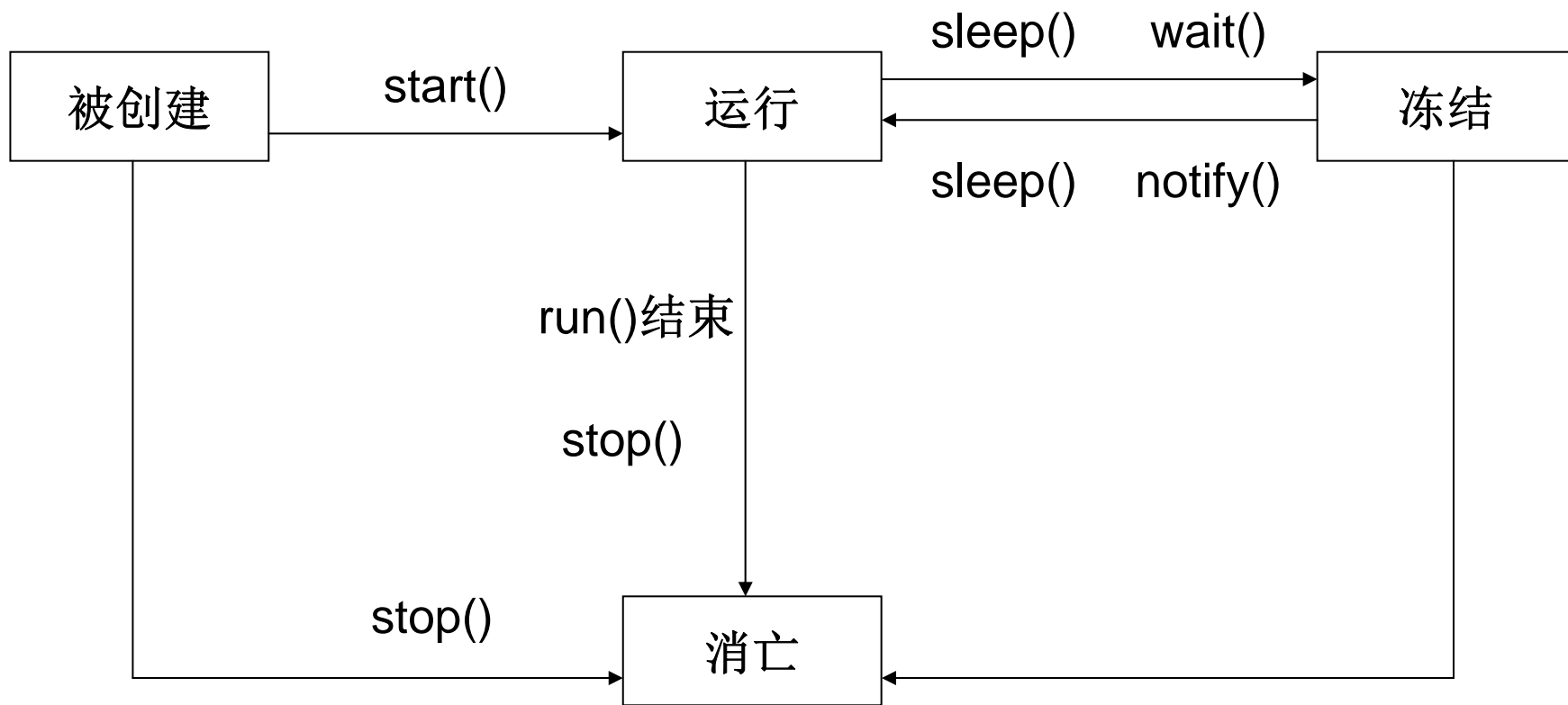
---

### 继承Thread类

1. 子类覆盖父类中的run方法，将线程运行的代码存放在run中。
2. 建立子类对象的同时线程也被创建。
3. 通过调用start方法开启线程。

### 例程

## 线程的四种状态



`sleep`方法需要指定睡眠时间，单位是毫秒。

一个特殊的状态：就绪。具备了执行资格，但是还没有获取资源。

## 创建线程方式二

---

### 实现Runnable接口

1. 子类覆盖接口中的run方法。
2. 通过Thread类创建线程，并将实现了Runnable接口的子类对象作为参数传递给Thread类的构造函数。
3. Thread类对象调用start方法开启线程。

思考：为什么要给Thread类的构造函数传递Runnable的子类对象？

## 线程安全问题

---

导致安全问题的出现的原因:

- 多个线程访问出现延迟。
- 线程随机性 。

注：线程安全问题在理想状态下，不容易出现，但一旦出现对软件的影响是非常大的。

## 同步 (synchronized)

---

格式:

synchronized (对象)

{

    需要同步的代码;

}

同步可以解决安全问题的根本原因就在那个对象上。

该对象如同锁的功能。

## 同步的特点

---

同步的前提:

- 同步需要两个或者两个以上的线程。
- 多个线程使用的是同一个锁。

未满足这两个条件，不能称其为同步。

同步的弊端:

当线程相当多时，因为每个线程都会去判断同步上的锁，这是很耗费资源的，无形中会降低程序的运行效率。



## 同步函数

---

格式:

在函数上加上**synchronized**修饰符即可。

思考：同步函数用的是哪个锁呢？

程序验证。

## 线程间通信

例程。



思考1: `wait()`, `notify()`, `notifyAll()`, 用来操作线程为什么定义在了 `Object` 类中?

- 1, 这些方法存在与同步中。
- 2, 使用这些方法时必须标识所属的同步的锁。
- 3, 锁可以是任意对象, 所以任意对象调用的方法一定定义 `Object` 类中。

思考2: `wait()`, `sleep()` 有什么区别?

`wait()`: 释放cpu执行权, 释放锁。

`sleep()`: 释放cpu执行权, 不释放锁。

## 停止线程

---

### 1. 定义循环结束标记

- 因为线程运行代码一般都是循环，只要控制了循环即可。

### 2. 使用interrupt（中断）方法。

- 该方法是结束线程的冻结状态，使线程回到运行状态中来。

注：stop方法已经过时不再使用。

## 线程类的其他方法

---

- `setPriority(int num)`
- `setDaemon(boolean b)`
- `join()`
- 自定义线程名称
- `toString()`