

6.JDK并发包2 .....	1
1. 线程池的基本使用 .....	2
1.1. 为什么需要线程池.....	2
1.2. JDK为我们提供了哪些支持.....	2
1.3. 线程池的使用.....	2
1.3.1. 线程池的种类 .....	2
1.3.2. 不同线程池的共同性 .....	2
1.4. 线程池使用的小例子.....	2
1.4.1. 简单线程池 .....	3
1.4.2. ScheduledThreadPool .....	3
2. 扩展和增强线程池 .....	3
2.1. 回调接口 .....	3
2.2. 拒绝策略.....	3
2.3. 自定义ThreadFactory .....	3
3. 线程池及其核心代码分析 .....	3
4. ForkJoin.....	3
4.1. 思想.....	3
4.2. 使用接口 .....	4
4.2.1. RecursiveAction .....	4
4.2.2. RecursiveTask .....	4
4.3. 简单例子.....	4
4.4. 实现要素.....	4
4.4.1. 工作窃取 .....	4

## 1. 线程池的基本使用

### 1.1. 为什么需要线程池

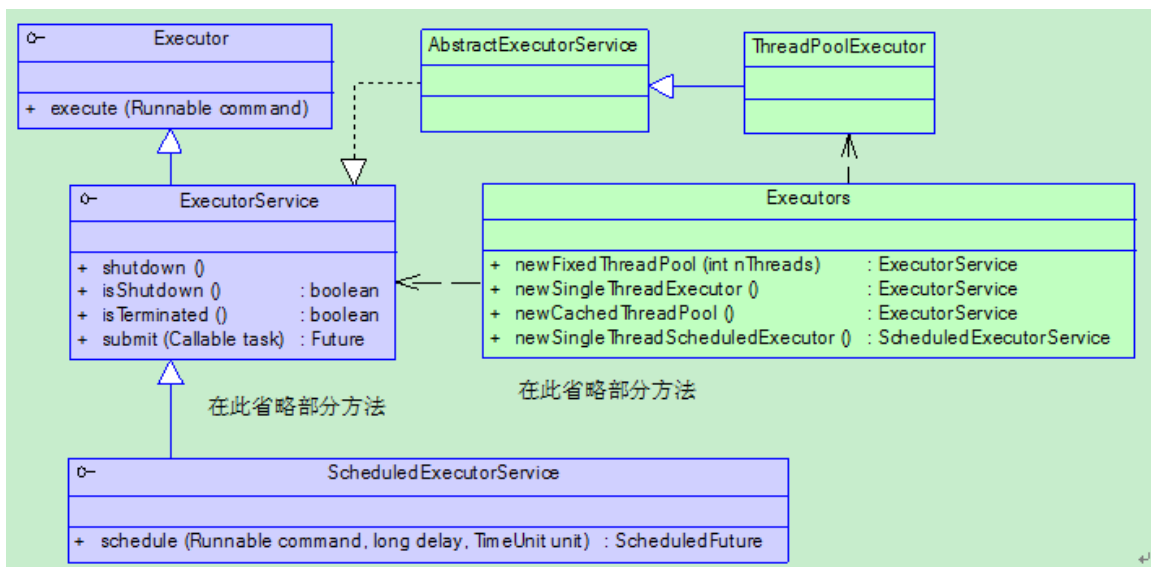
原因

简单的线程池实现

### 1.2. JDK为我们提供了哪些支持

内置线程池

#### 1.2.1.



### 1.3. 线程池的使用

#### 1.3.1. 线程池的种类

`newFixedThreadPool`

`newSingleThreadExecutor`

`newCachedThreadPool`

`newScheduledThreadPool`

#### 1.3.2. 不同线程池的共同性

线程池构造函数详解

### 1.4. 线程池使用的小例子

#### **1.4.1. 简单线程池**

#### **1.4.2. ScheduledThreadPool**

### **2. 扩展和增强线程池**

#### **2.1. 回调接口**

beforeExecute

afterExecute

terminated

#### **2.2. 拒绝策略**

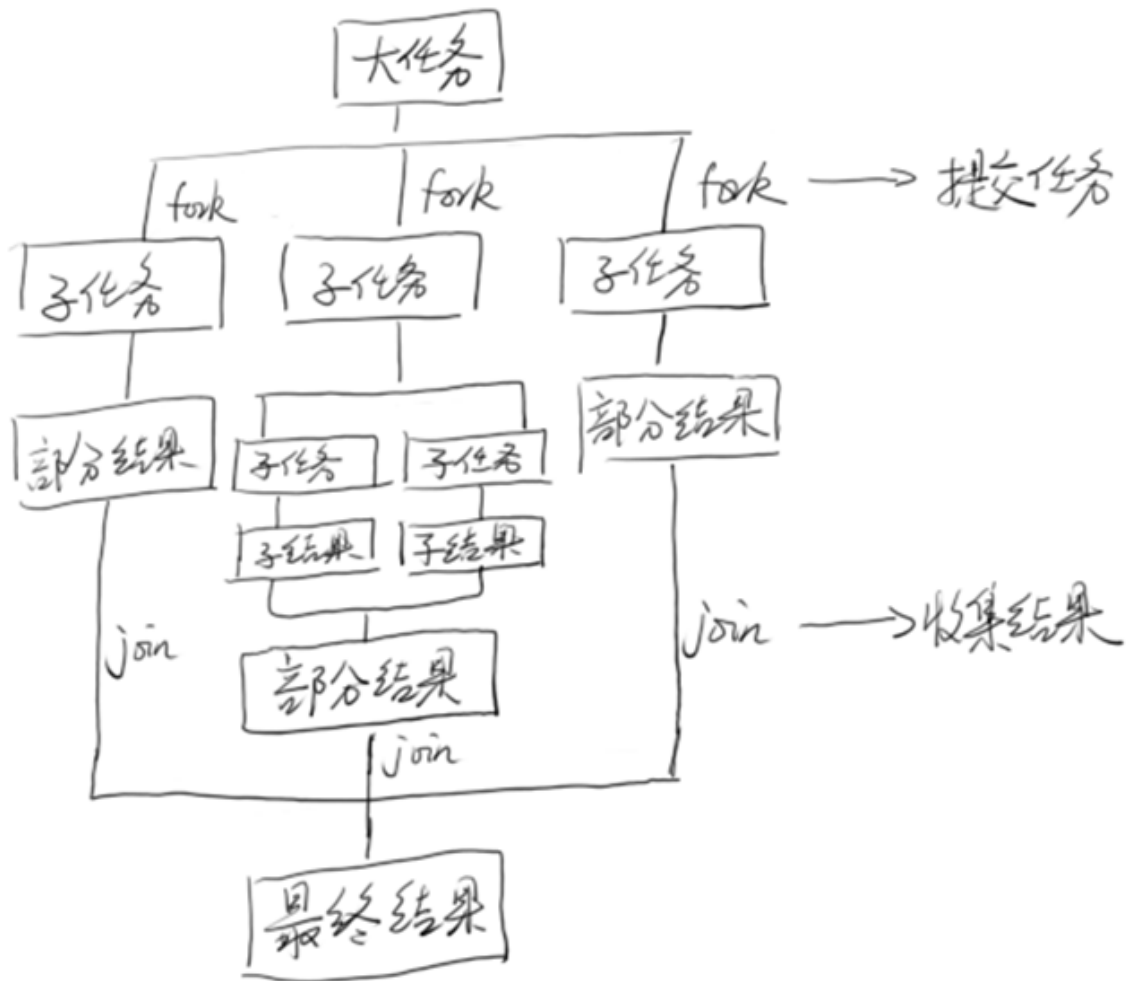
#### **2.3. 自定义ThreadFactory**

### **3. 线程池及其核心代码分析**

### **4. ForkJoin**

#### **4.1. 思想**

##### **4.1.1.**



## 4.2. 使用接口

### 4.2.1. RecursiveAction

无返回值

### 4.2.2. RecursiveTask

有返回值

## 4.3. 简单例子

## 4.4. 实现要素

### 4.4.1. 工作窃取

