1. 分支合并框架
   1. 原理
      1. Fork：把一个复杂任务进行分拆，大事化小
      2. Join：把分拆任务的结果进行合并
   2. 相关类
      1. ForkJoinPool : 分支合并池    类比=>   线程池
      2. ForkJoinTask : 类比=>   FutureTask
      3. RecursiveTask: 递归任务：继承后可以实现递归(自己调自己)调用的任务

范例:

|  |
| --- |
| class Fibonacci extends RecursiveTask<Integer> {     final int n;     Fibonacci(int n) { this.n = n; }     Integer compute() {       if (n <= 1)         return n;       Fibonacci f1 = new Fibonacci(n - 1);       f1.fork();       Fibonacci f2 = new Fibonacci(n - 2);       return f2.compute() + f1.join();     }   } |

范例:分支合并案例

|  |
| --- |
| **import**java.util.concurrent.ExecutionException;  **import**java.util.concurrent.ForkJoinPool;  **import**java.util.concurrent.ForkJoinTask;  **import**java.util.concurrent.RecursiveTask;  **class**MyTask **extends**RecursiveTask<Integer>{  **private static final**Integer ***ADJUST\_VALUE***= 10;  **private int begin**;  **private int end**;  **private int result**;  **public**MyTask(**int**begin, **int**end) {  **this**.**begin**= begin;  **this**.**end**= end;      }      @Override  **protected**Integer compute() {  **if**((**end**- **begin**)<=***ADJUST\_VALUE***){  **for**(**int**i =**begin**;i <= **end**;i++){  **result**= **result**+ i;             }          }**else**{  **int**middle = (**begin**+ **end**)/2;              MyTask task01 = **new**MyTask(**begin**,middle);              MyTask task02 = **new**MyTask(middle+1,**end**);              task01.fork();              task02.fork();  **result**=  task01.join() + task02.join();          }  **return result**;      }  }  */\*\**  *\* 分支合并例子*  *\* ForkJoinPool*  *\* ForkJoinTask*  *\* RecursiveTask*  *\*/*  **public class**ForkJoinDemo {  **public static void**main(String[] args) **throws**ExecutionException, InterruptedException {          MyTask myTask = **new**MyTask(0,100);          ForkJoinPool forkJoinPool = **new**ForkJoinPool();          ForkJoinTask<Integer> forkJoinTask = forkJoinPool.submit(myTask);          System.***out***.println(forkJoinTask.get());          forkJoinPool.shutdown();      }  } |