

## 华东师范大学数据科学与工程学院实验报告

课程名称：计算机网络与编程

年级：2021

上机实践成绩：

指导教师：张召

姓名：温兆和

学号：10205501432

上机实践名称： **JAVA 多线程编程**

上机实践日期：2022.04.01

上机实践编号：06 07

组号：001-432

上机实践时间：13: 00

### 一、实验目的

熟悉 JAVA 多线程编程；  
熟悉并掌握线程创建、线程控制；  
熟悉并掌握线程同步、线程交互。

### 二、实验任务

使用常用的两种方式创建线程；  
使用 `join()`, `yield()` 等方法对线程进行控制；  
学习使用 `synchronized` 关键字进行线程同步；  
学习使用 `wait()` 和 `notify()` 方法进行线程交互。

### 三、使用环境

IntelliJ IDEA 2020.3.2;  
JDK 11.0.6.

### 四、实验过程

#### 4.1 创建线程

##### 4.1.1 继承 `Thread` 类创建线程

Task1. 改写类中的 `run()` 方法，将每个线程的 ID 也打印出来，代码及运行结果写到实验报告中。

代码：

```
package WEIZHI;
import java.util.Arrays;
import java.util.Scanner;
import java.io.PrintWriter;
import java.util.Calendar;
import java.util.Date;
import java.lang.Thread;

class Threadtest01 extends Thread
{
    public void run()
    {
        System.out.println(Thread.currentThread().getId());
    }
    public static void main(String[] args)
    {
        Threadtest01 mythread1 = new Threadtest01();
        Threadtest01 mythread2 = new Threadtest01();
        Threadtest01 mythread3 = new Threadtest01();
        mythread1.start();
        mythread2.start();
    }
}
```

```

        mythread3.start();
        mythread1.run();
        mythread2.run();
        mythread3.run();
    }
}

```

结果:

"C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2021.3.2\jbr\bin\java.exe"  
 "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA 2021.3.2\help\IntelliJ IDEA Community  
 Edition 2021.3.2\lib\idea\_rt.jar=53722:C:\Program Files\JetBrains\IntelliJ IDEA  
 2021.3.2\help\IntelliJ IDEA Community Edition 2021.3.2\bin" -Dfile.encoding=UTF-8 -  
 classpath C:\Users\wzh\IdeaProjects\Demo\out\production\Demo WEIZHI.Threadtest01

1  
 1  
 1  
 15  
 14  
 16

Process finished with exit code 0

#### 4.1.2 实现 Runnable 接口创建线程

Task2. 通过实现 Runnable 接口的方式编写两个线程，一个线程负责打印字母，另一个线程负责打印数字，两个线程同时进行打印，要求打印出来的结果形式为 a1b23c456d7891.....z.....（数字 1-9 循环 2 次）。将关键代码和总结的内容写到实验报告中。

```

package WEIZHI;
import java.util.Arrays;
import java.util.Scanner;
import java.io.PrintWriter;
import java.util.Calendar;
import java.util.Date;
import java.lang.Thread;

class Number implements Runnable
{
    public static int i=0,j=0;
    private char[] ZIMU = new
char[]{'a','b','c','d','e','f','g','h','i','j','k','l','m','n','o','p','q','r','s',
't','u','v','w','x','y','z'};
    @Override
    public void run() {
        while (i<26&&j<26)
        {
            if (Thread.currentThread().getName().equals("p1"))
            {
                System.out.print(ZIMU[i]);
                i++;
            }
            else
            {
                System.out.print(j);
                j++;
            }
        }
    }
}

```

```

public class Abc{
    public static void main(String[] args) throws InterruptedException
    {
        Number nr = new Number();
        Thread thread01 = new Thread(nr, "p1");
        Thread thread02 = new Thread(nr, "p2");
        thread01.start();
        thread02.start();
        thread01.join();
    }
}

```

结果:

"C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2021.3.2\jbr\bin\java.exe"  
 "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA 2021.3.2\help\IntelliJ IDEA Community  
 Edition 2021.3.2\lib\idea\_rt.jar=57274:C:\Program Files\JetBrains\IntelliJ IDEA  
 2021.3.2\help\IntelliJ IDEA Community Edition 2021.3.2\bin" -Dfile.encoding=UTF-8 -  
 classpath C:\Users\wzh\IdeaProjects\Demo\out\production\Demo WEIZHI.Abc  
 012a3b456789cdefg10h11ijk12mnopqrst131415161718uvwxyz19  
 Process finished with exit code 0

## 4.2 线程控制

### 4.2.1 线程 join()

Task3. 完善代码，用 join 方法实现正常的逻辑，并将关键代码和结果写到实验报告中。

代码:

```

package WEIZHI;
import java.util.Arrays;
import java.util.Scanner;
import java.io.PrintWriter;
import java.util.Calendar;
import java.util.Date;
import java.lang.Thread;

public class Abc implements Runnable{
    @Override
    public void run(){
        System.out.println(Thread.currentThread().getName());
    }
    public static void main(String[] args) throws InterruptedException {
        Abc join = new Abc();
        Thread thread1 = new Thread(join, "上课铃响");
        Thread thread2 = new Thread(join, "老师上课");
        Thread thread3 = new Thread(join, "下课铃响");
        Thread thread4 = new Thread(join, "老师下课");
        thread1.start();
        thread2.start();
        thread1.join();
        thread3.start();
        thread4.start();
    }
}

```

结果:

"C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2021.3.2\jbr\bin\java.exe"  
 "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA 2021.3.2\help\IntelliJ IDEA Community  
 Edition 2021.3.2\lib\idea\_rt.jar=56603:C:\Program Files\JetBrains\IntelliJ IDEA  
 2021.3.2\help\IntelliJ IDEA Community Edition 2021.3.2\bin" -Dfile.encoding=UTF-8 -

classpath C:\Users\wzh\IdeaProjects\Demo\out\production\Demo WEIZHI.Abc

上课铃响

老师上课

下课铃响

老师下课

Process finished with exit code 0

#### 4.2.2 守护线程

代码:

```
package WEIZHI;
import java.util.Arrays;
import java.util.Scanner;
import java.io.PrintWriter;
import java.util.Calendar;
import java.util.Date;
import java.lang.Thread;

public class Abc implements Runnable{
    @Override
    public void run(){
        int worktime = 0;
        while(true){
            System.out.println("助教在教室的第"+ worktime + "秒");
            try{
                Thread.currentThread().sleep(1000);
            }catch (InterruptedException e){
                e.printStackTrace();
            }
            worktime ++;
        }
    }
    public static void main(String[] args) throws InterruptedException{
        Abc inClassroom = new Abc();
        Thread thread = new Thread(inClassroom,"助教");
        thread.setDaemon(true);
        thread.start();
        for(int i = 0; i < 10; i++){
            thread.sleep(1000);
            System.out.println("同学们正在上课");
            if(i == 9){
                System.out.println("同学们下课了");
            }
        }
    }
}
```

结果:

"C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2021.3.2\jbr\bin\java.exe"  
-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA 2021.3.2\help\IntelliJ IDEA Community  
Edition 2021.3.2\lib\idea\_rt.jar=61871:C:\Program Files\JetBrains\IntelliJ IDEA  
2021.3.2\help\IntelliJ IDEA Community Edition 2021.3.2\bin" -Dfile.encoding=UTF-8 -  
classpath C:\Users\wzh\IdeaProjects\Demo\out\production\Demo WEIZHI.Abc

助教在教室的第 0 秒

同学们正在上课

助教在教室的第 1 秒

同学们正在上课

助教在教室的第 2 秒  
同学们正在上课  
助教在教室的第 3 秒  
同学们正在上课  
助教在教室的第 4 秒  
同学们正在上课  
助教在教室的第 5 秒  
同学们正在上课  
助教在教室的第 6 秒  
同学们正在上课  
助教在教室的第 7 秒  
同学们正在上课  
助教在教室的第 8 秒  
同学们正在上课  
助教在教室的第 9 秒  
同学们正在上课  
同学们下课了

Process finished with exit code 0

#### 4.2.3 线程优先级

#### 4.2.4 线程让步

Task5. 完善代码，将两个线程设置为不同的优先级，并将第一个线程设置为让步状态。  
将关键的代码和总结的内容写到实验报告中，并总结线程让步的特点。

代码 1:

```
package WEIZHI;
import java.util.Arrays;
import java.util.Scanner;
import java.io.PrintWriter;
import java.util.Calendar;
import java.util.Date;
import java.lang.Thread;

class MyRunnable01 implements Runnable{
    @Override
    public void run(){
        for(int i = 1; i < 100; i++){
            System.out.println("线程"+ Thread.currentThread().getName() + "第" + i +
"次执行! ");
            Thread.yield();
        }
    }
}
class MyRunnable02 implements Runnable{
    @Override
    public void run(){
        for(int i = 1; i < 100; i++){
            System.out.println("线程"+ Thread.currentThread().getName() + "第" + i +
"次执行! ");
        }
    }
}
public class Abc {
    public static void main(String[] args) throws InterruptedException{
        Thread t1 = new Thread(new MyRunnable01(), "p1");
```

```
Thread t2 = new Thread(new MyRunnable02(), "p2");  
t1.setPriority(5);  
t2.setPriority(5);  
t1.start();  
t2.start();  
}  
}
```

结果 1:

"C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2021.3.2\jbr\bin\java.exe"  
"-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA 2021.3.2\help\IntelliJ IDEA Community  
Edition 2021.3.2\lib\idea\_rt.jar=62645:C:\Program Files\JetBrains\IntelliJ IDEA  
2021.3.2\help\IntelliJ IDEA Community Edition 2021.3.2\bin" -Dfile.encoding=UTF-8 -  
classpath C:\Users\wzh\IdeaProjects\Demo\out\production\Demo WEIZHI.Abc

线程 p2 第 1 次执行!  
线程 p1 第 1 次执行!  
线程 p2 第 2 次执行!  
线程 p2 第 3 次执行!  
线程 p2 第 4 次执行!  
线程 p2 第 5 次执行!  
线程 p2 第 6 次执行!  
线程 p2 第 7 次执行!  
线程 p2 第 8 次执行!  
线程 p2 第 9 次执行!  
线程 p2 第 10 次执行!  
线程 p2 第 11 次执行!  
线程 p2 第 12 次执行!  
线程 p2 第 13 次执行!  
线程 p2 第 14 次执行!  
线程 p2 第 15 次执行!  
线程 p2 第 16 次执行!  
线程 p1 第 2 次执行!  
线程 p2 第 17 次执行!  
线程 p2 第 18 次执行!  
线程 p2 第 19 次执行!  
线程 p2 第 20 次执行!  
线程 p2 第 21 次执行!  
线程 p2 第 22 次执行!  
线程 p2 第 23 次执行!  
线程 p2 第 24 次执行!  
线程 p2 第 25 次执行!  
线程 p2 第 26 次执行!  
线程 p2 第 27 次执行!  
线程 p2 第 28 次执行!  
线程 p2 第 29 次执行!  
线程 p2 第 30 次执行!  
线程 p2 第 31 次执行!  
线程 p2 第 32 次执行!

线程 p2 第 33 次执行!  
线程 p2 第 34 次执行!  
线程 p2 第 35 次执行!  
线程 p2 第 36 次执行!  
线程 p2 第 37 次执行!  
线程 p2 第 38 次执行!  
线程 p2 第 39 次执行!  
线程 p2 第 40 次执行!  
线程 p2 第 41 次执行!  
线程 p2 第 42 次执行!  
线程 p2 第 43 次执行!  
线程 p2 第 44 次执行!  
线程 p2 第 45 次执行!  
线程 p2 第 46 次执行!  
线程 p2 第 47 次执行!  
线程 p2 第 48 次执行!  
线程 p2 第 49 次执行!  
线程 p2 第 50 次执行!  
线程 p2 第 51 次执行!  
线程 p1 第 3 次执行!  
线程 p2 第 52 次执行!  
线程 p1 第 4 次执行!  
线程 p2 第 53 次执行!  
线程 p2 第 54 次执行!  
线程 p2 第 55 次执行!  
线程 p2 第 56 次执行!  
线程 p2 第 57 次执行!  
线程 p2 第 58 次执行!  
线程 p2 第 59 次执行!  
线程 p2 第 60 次执行!  
线程 p2 第 61 次执行!  
线程 p2 第 62 次执行!  
线程 p2 第 63 次执行!  
线程 p2 第 64 次执行!  
线程 p2 第 65 次执行!  
线程 p1 第 5 次执行!  
线程 p2 第 66 次执行!  
线程 p2 第 67 次执行!  
线程 p2 第 68 次执行!  
线程 p2 第 69 次执行!  
线程 p2 第 70 次执行!  
线程 p2 第 71 次执行!  
线程 p2 第 72 次执行!  
线程 p2 第 73 次执行!

线程 p2 第 74 次执行!  
线程 p2 第 75 次执行!  
线程 p1 第 6 次执行!  
线程 p2 第 76 次执行!  
线程 p2 第 77 次执行!  
线程 p2 第 78 次执行!  
线程 p2 第 79 次执行!  
线程 p2 第 80 次执行!  
线程 p2 第 81 次执行!  
线程 p2 第 82 次执行!  
线程 p2 第 83 次执行!  
线程 p2 第 84 次执行!  
线程 p2 第 85 次执行!  
线程 p1 第 7 次执行!  
线程 p2 第 86 次执行!  
线程 p1 第 8 次执行!  
线程 p2 第 87 次执行!  
线程 p2 第 88 次执行!  
线程 p1 第 9 次执行!  
线程 p2 第 89 次执行!  
线程 p1 第 10 次执行!  
线程 p2 第 90 次执行!  
线程 p1 第 11 次执行!  
线程 p2 第 91 次执行!  
线程 p1 第 12 次执行!  
线程 p2 第 92 次执行!  
线程 p1 第 13 次执行!  
线程 p2 第 93 次执行!  
线程 p1 第 14 次执行!  
线程 p2 第 94 次执行!  
线程 p1 第 15 次执行!  
线程 p2 第 95 次执行!  
线程 p1 第 16 次执行!  
线程 p2 第 96 次执行!  
线程 p1 第 17 次执行!  
线程 p2 第 97 次执行!  
线程 p1 第 18 次执行!  
线程 p2 第 98 次执行!  
线程 p1 第 19 次执行!  
线程 p2 第 99 次执行!  
线程 p1 第 20 次执行!  
线程 p1 第 21 次执行!  
线程 p1 第 22 次执行!  
线程 p1 第 23 次执行!



线程 p1 第 24 次执行!  
线程 p1 第 25 次执行!  
线程 p1 第 26 次执行!  
线程 p1 第 27 次执行!  
线程 p1 第 28 次执行!  
线程 p1 第 29 次执行!  
线程 p1 第 30 次执行!  
线程 p1 第 31 次执行!  
线程 p1 第 32 次执行!  
线程 p1 第 33 次执行!  
线程 p1 第 34 次执行!  
线程 p1 第 35 次执行!  
线程 p1 第 36 次执行!  
线程 p1 第 37 次执行!  
线程 p1 第 38 次执行!  
线程 p1 第 39 次执行!  
线程 p1 第 40 次执行!  
线程 p1 第 41 次执行!  
线程 p1 第 42 次执行!  
线程 p1 第 43 次执行!  
线程 p1 第 44 次执行!  
线程 p1 第 45 次执行!  
线程 p1 第 46 次执行!  
线程 p1 第 47 次执行!  
线程 p1 第 48 次执行!  
线程 p1 第 49 次执行!  
线程 p1 第 50 次执行!  
线程 p1 第 51 次执行!  
线程 p1 第 52 次执行!  
线程 p1 第 53 次执行!  
线程 p1 第 54 次执行!  
线程 p1 第 55 次执行!  
线程 p1 第 56 次执行!  
线程 p1 第 57 次执行!  
线程 p1 第 58 次执行!  
线程 p1 第 59 次执行!  
线程 p1 第 60 次执行!  
线程 p1 第 61 次执行!  
线程 p1 第 62 次执行!  
线程 p1 第 63 次执行!  
线程 p1 第 64 次执行!  
线程 p1 第 65 次执行!  
线程 p1 第 66 次执行!  
线程 p1 第 67 次执行!

线程 p1 第 68 次执行!  
线程 p1 第 69 次执行!  
线程 p1 第 70 次执行!  
线程 p1 第 71 次执行!  
线程 p1 第 72 次执行!  
线程 p1 第 73 次执行!  
线程 p1 第 74 次执行!  
线程 p1 第 75 次执行!  
线程 p1 第 76 次执行!  
线程 p1 第 77 次执行!  
线程 p1 第 78 次执行!  
线程 p1 第 79 次执行!  
线程 p1 第 80 次执行!  
线程 p1 第 81 次执行!  
线程 p1 第 82 次执行!  
线程 p1 第 83 次执行!  
线程 p1 第 84 次执行!  
线程 p1 第 85 次执行!  
线程 p1 第 86 次执行!  
线程 p1 第 87 次执行!  
线程 p1 第 88 次执行!  
线程 p1 第 89 次执行!  
线程 p1 第 90 次执行!  
线程 p1 第 91 次执行!  
线程 p1 第 92 次执行!  
线程 p1 第 93 次执行!  
线程 p1 第 94 次执行!  
线程 p1 第 95 次执行!  
线程 p1 第 96 次执行!  
线程 p1 第 97 次执行!  
线程 p1 第 98 次执行!  
线程 p1 第 99 次执行!

Process finished with exit code 0

代码 2:

```
package WEIZHI;
import java.util.Arrays;
import java.util.Scanner;
import java.io.PrintWriter;
import java.util.Calendar;
import java.util.Date;
import java.lang.Thread;

class MyRunnable01 implements Runnable{
    @Override
    public void run(){
        for(int i = 1; i < 100; i++){
            System.out.println("线程"+ Thread.currentThread().getName() + "第" + i +
```

```
"次执行! ");
    Thread.yield();
}
}
class MyRunnable02 implements Runnable{
    @Override
    public void run(){
        for(int i = 1; i < 100; i++){
            System.out.println("线程"+ Thread.currentThread().getName() + "第" + i +
"次执行! ");
        }
    }
}
public class Abc {
    public static void main(String[] args) throws InterruptedException{
        Thread t1 = new Thread(new MyRunnable01(), "p1");
        Thread t2 = new Thread(new MyRunnable02(), "p2");
        t1.setPriority(10);
        t2.setPriority(1);
        t1.start();
        t2.start();
    }
}
```

结果 2:

"C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2021.3.2\jbr\bin\java.exe"  
"-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA 2021.3.2\help\IntelliJ IDEA Community  
Edition 2021.3.2\lib\idea\_rt.jar=62696:C:\Program Files\JetBrains\IntelliJ IDEA  
2021.3.2\help\IntelliJ IDEA Community Edition 2021.3.2\bin" -Dfile.encoding=UTF-8 -  
classpath C:\Users\wzh\IdeaProjects\Demo\out\production\Demo WEIZHI.Abc

线程 p1 第 1 次执行!  
线程 p2 第 1 次执行!  
线程 p1 第 2 次执行!  
线程 p2 第 2 次执行!  
线程 p1 第 3 次执行!  
线程 p2 第 3 次执行!  
线程 p1 第 4 次执行!  
线程 p2 第 4 次执行!  
线程 p1 第 5 次执行!  
线程 p2 第 5 次执行!  
线程 p1 第 6 次执行!  
线程 p2 第 6 次执行!  
线程 p1 第 7 次执行!  
线程 p2 第 7 次执行!  
线程 p1 第 8 次执行!  
线程 p2 第 8 次执行!  
线程 p1 第 9 次执行!  
线程 p2 第 9 次执行!  
线程 p1 第 10 次执行!  
线程 p2 第 10 次执行!  
线程 p1 第 11 次执行!  
线程 p2 第 11 次执行!  
线程 p1 第 12 次执行!

线程 p2 第 12 次执行!  
线程 p1 第 13 次执行!  
线程 p2 第 13 次执行!  
线程 p1 第 14 次执行!  
线程 p2 第 14 次执行!  
线程 p1 第 15 次执行!  
线程 p2 第 15 次执行!  
线程 p1 第 16 次执行!  
线程 p2 第 16 次执行!  
线程 p1 第 17 次执行!  
线程 p2 第 17 次执行!  
线程 p1 第 18 次执行!  
线程 p2 第 18 次执行!  
线程 p1 第 19 次执行!  
线程 p2 第 19 次执行!  
线程 p1 第 20 次执行!  
线程 p2 第 20 次执行!  
线程 p1 第 21 次执行!  
线程 p2 第 21 次执行!  
线程 p1 第 22 次执行!  
线程 p2 第 22 次执行!  
线程 p1 第 23 次执行!  
线程 p2 第 23 次执行!  
线程 p1 第 24 次执行!  
线程 p2 第 24 次执行!  
线程 p1 第 25 次执行!  
线程 p2 第 25 次执行!  
线程 p2 第 26 次执行!  
线程 p2 第 27 次执行!  
线程 p2 第 28 次执行!  
线程 p2 第 29 次执行!  
线程 p1 第 26 次执行!  
线程 p1 第 27 次执行!  
线程 p1 第 28 次执行!  
线程 p1 第 29 次执行!  
线程 p1 第 30 次执行!  
线程 p1 第 31 次执行!  
线程 p1 第 32 次执行!  
线程 p1 第 33 次执行!  
线程 p1 第 34 次执行!  
线程 p1 第 35 次执行!  
线程 p2 第 30 次执行!  
线程 p2 第 31 次执行!  
线程 p2 第 32 次执行!

线程 p2 第 33 次执行!  
线程 p2 第 34 次执行!  
线程 p1 第 36 次执行!  
线程 p2 第 35 次执行!  
线程 p1 第 37 次执行!  
线程 p2 第 36 次执行!  
线程 p1 第 38 次执行!  
线程 p2 第 37 次执行!  
线程 p1 第 39 次执行!  
线程 p2 第 38 次执行!  
线程 p1 第 40 次执行!  
线程 p2 第 39 次执行!  
线程 p1 第 41 次执行!  
线程 p2 第 40 次执行!  
线程 p1 第 42 次执行!  
线程 p2 第 41 次执行!  
线程 p1 第 43 次执行!  
线程 p2 第 42 次执行!  
线程 p1 第 44 次执行!  
线程 p2 第 43 次执行!  
线程 p1 第 45 次执行!  
线程 p2 第 44 次执行!  
线程 p1 第 46 次执行!  
线程 p2 第 45 次执行!  
线程 p1 第 47 次执行!  
线程 p2 第 46 次执行!  
线程 p1 第 48 次执行!  
线程 p2 第 47 次执行!  
线程 p1 第 49 次执行!  
线程 p2 第 48 次执行!  
线程 p1 第 50 次执行!  
线程 p2 第 49 次执行!  
线程 p2 第 50 次执行!  
线程 p1 第 51 次执行!  
线程 p2 第 51 次执行!  
线程 p1 第 52 次执行!  
线程 p2 第 52 次执行!  
线程 p1 第 53 次执行!  
线程 p2 第 53 次执行!  
线程 p1 第 54 次执行!  
线程 p2 第 54 次执行!  
线程 p1 第 55 次执行!  
线程 p2 第 55 次执行!  
线程 p1 第 56 次执行!

线程 p2 第 56 次执行!  
线程 p1 第 57 次执行!  
线程 p2 第 57 次执行!  
线程 p1 第 58 次执行!  
线程 p2 第 58 次执行!  
线程 p1 第 59 次执行!  
线程 p2 第 59 次执行!  
线程 p1 第 60 次执行!  
线程 p2 第 60 次执行!  
线程 p1 第 61 次执行!  
线程 p2 第 61 次执行!  
线程 p1 第 62 次执行!  
线程 p2 第 62 次执行!  
线程 p1 第 63 次执行!  
线程 p2 第 63 次执行!  
线程 p1 第 64 次执行!  
线程 p2 第 64 次执行!  
线程 p1 第 65 次执行!  
线程 p2 第 65 次执行!  
线程 p1 第 66 次执行!  
线程 p2 第 66 次执行!  
线程 p1 第 67 次执行!  
线程 p2 第 67 次执行!  
线程 p1 第 68 次执行!  
线程 p1 第 69 次执行!  
线程 p1 第 70 次执行!  
线程 p1 第 71 次执行!  
线程 p1 第 72 次执行!  
线程 p1 第 73 次执行!  
线程 p1 第 74 次执行!  
线程 p2 第 68 次执行!  
线程 p1 第 75 次执行!  
线程 p2 第 69 次执行!  
线程 p1 第 76 次执行!  
线程 p2 第 70 次执行!  
线程 p1 第 77 次执行!  
线程 p2 第 71 次执行!  
线程 p1 第 78 次执行!  
线程 p2 第 72 次执行!  
线程 p1 第 79 次执行!  
线程 p2 第 73 次执行!  
线程 p1 第 80 次执行!  
线程 p2 第 74 次执行!  
线程 p1 第 81 次执行!

线程 p2 第 75 次执行!  
线程 p1 第 82 次执行!  
线程 p2 第 76 次执行!  
线程 p2 第 77 次执行!  
线程 p2 第 78 次执行!  
线程 p2 第 79 次执行!  
线程 p2 第 80 次执行!  
线程 p1 第 83 次执行!  
线程 p2 第 81 次执行!  
线程 p1 第 84 次执行!  
线程 p2 第 82 次执行!  
线程 p1 第 85 次执行!  
线程 p2 第 83 次执行!  
线程 p1 第 86 次执行!  
线程 p2 第 84 次执行!  
线程 p1 第 87 次执行!  
线程 p2 第 85 次执行!  
线程 p1 第 88 次执行!  
线程 p2 第 86 次执行!  
线程 p1 第 89 次执行!  
线程 p2 第 87 次执行!  
线程 p1 第 90 次执行!  
线程 p2 第 88 次执行!  
线程 p1 第 91 次执行!  
线程 p2 第 89 次执行!  
线程 p1 第 92 次执行!  
线程 p2 第 90 次执行!  
线程 p1 第 93 次执行!  
线程 p2 第 91 次执行!  
线程 p1 第 94 次执行!  
线程 p2 第 92 次执行!  
线程 p1 第 95 次执行!  
线程 p2 第 93 次执行!  
线程 p1 第 96 次执行!  
线程 p2 第 94 次执行!  
线程 p1 第 97 次执行!  
线程 p2 第 95 次执行!  
线程 p1 第 98 次执行!  
线程 p2 第 96 次执行!  
线程 p2 第 97 次执行!  
线程 p2 第 98 次执行!  
线程 p2 第 99 次执行!  
线程 p1 第 99 次执行!

Process finished with exit code 0

对比和分析：

由这两段代码及其运行结果可以看出：当一个线程的优先级显著大于另一个时，即使它被设置为让步状态，优先级比它小的线程仍然会和它同时运行，而不是先于它；当两个线程优先级相近或该线程优先级小于另一个，另一个线程才会在它被设为让步状态时优先被运行。

#### 4.3 线程同步

Task6. 运行测试类中的 main 函数，将 num 可能得到的两种不同的值截图。

结果 1:

```
"C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2021.3.2\jbr\bin\java.exe"
"-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA 2021.3.2\license\third-party\IntelliJ
IDEA Community Edition 2021.3.2\lib\idea_rt.jar=60871:C:\Program Files\JetBrains\IntelliJ
IDEA 2021.3.2\license\third-party\IntelliJ IDEA Community Edition 2021.3.2\bin" -
Dfile.encoding=UTF-8 -classpath C:\Users\wzh\IdeaProjects\WEIZHI\out\production\WEIZHI
WEIZHI.TestPlusMinus
    num equals998
```

Process finished with exit code 0

结果 2:

```
"C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2021.3.2\jbr\bin\java.exe"
"-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA 2021.3.2\license\third-party\IntelliJ
IDEA Community Edition 2021.3.2\lib\idea_rt.jar=60946:C:\Program Files\JetBrains\IntelliJ
IDEA 2021.3.2\license\third-party\IntelliJ IDEA Community Edition 2021.3.2\bin" -
Dfile.encoding=UTF-8 -classpath C:\Users\wzh\IdeaProjects\WEIZHI\out\production\WEIZHI
WEIZHI.TestPlusMinus
    num equals1005
```

Process finished with exit code 0

Task7. 换一种方法创建线程，将修改后的代码附在实验报告中。

代码:

```
package WEIZHI;
import java.util.Arrays;
import java.util.Scanner;
import java.io.PrintWriter;
import java.util.Calendar;
import java.util.Date;
import java.lang.Thread;

public class weilegezhi
{
    public volatile int num;
    public int cishu1;
    public int cishu2;
    public void plusOne() {
        num = num + 1;
        cishu1++;
    }
    public void minusOne() {
        num = num - 1;
        cishu2++;
    }
}

class JIAFA implements Runnable
```



```
{
    weilegezhi jf = new weilegezhi();
    @Override
    public void run()
    {
        if(Thread.currentThread().getName().equals("p1"))
        {
            for(int i=0;i<1000;i++)
            {
                jf.plusOne();
            }
        }
        else
        {
            for(int i=0;i<1000;i++)
            {
                jf.minusOne();
            }
        }
    }
}

class JIANFA implements Runnable
{
    JIAFA fj = new JIAFA();
    @Override
    public void run() {
        for(int i=0;i<1000;i++)
        {
            fj.jf.minusOne();
        }
    }
}

class ZUIZHONG
{
    public static void main(String[] args)
    {
        JIANFA bb = new JIANFA();
        Thread t1 = new Thread(bb.fj,"p1");
        Thread t2 = new Thread(bb,"p2");
        bb.fj.jf.num=1000;
        bb.fj.jf.cishu1=0;
        bb.fj.jf.cishu2=0;
        t1.start();
        t2.start();
        System.out.println(bb.fj.jf.num);
        System.out.println(bb.fj.jf.cishu1);
        System.out.println(bb.fj.jf.cishu2);
    }
}
```

结果:

"C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2021.3.2\jbr\bin\java.exe"  
"-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA 2021.3.2\license\third-party\IntelliJ  
IDEA Community Edition 2021.3.2\lib\idea\_rt.jar=61706:C:\Program Files\JetBrains\IntelliJ  
IDEA 2021.3.2\license\third-party\IntelliJ IDEA Community Edition 2021.3.2\bin" -  
Dfile.encoding=UTF-8 -classpath C:\Users\wzh\IdeaProjects\WEIZHI\out\production\WEIZHI  
WEIZHI.ZUIZHONG

1000  
1000  
1000

Process finished with exit code 0

Task8. 使用 `synchronized` 关键字修改 `TestPlusMinus` 测试类，使得 `num` 值不会出现不同步的问题，将修改后的代码以及运行结果写在实验报告内。

```
package WEIZHI;
import java.util.Arrays;
import java.util.Scanner;
import java.io.PrintWriter;
import java.util.Calendar;
import java.util.Date;
import java.lang.Thread;

public class weilegezhi {
    public volatile int num;
    public void plusOne(){
        num = num + 1;
    }
    public void minusOne(){
        num = num - 1;
    }
    public int printNum(){
        return num;
    }
}

class TestPlusMinus {
    public static void main(String[] args){
        weilegezhi plusMinus = new weilegezhi();
        plusMinus.num = 1000; // ??????1000
        int threadNum = 100; // ??????1000
        Thread[] plusThreads = new Thread[threadNum];
        Thread[] minusThreads = new Thread[threadNum];
        for(int i = 0; i < threadNum; i++){
            /* Java???????????????????????????????? */
            Thread thread1 = new Thread(){
                @Override
                public void run(){
                    synchronized (plusMinus)
                    {
                        try {
                            Thread.sleep(10);
                        } catch (InterruptedException e) {
                            e.printStackTrace();
                        }
                        plusMinus.plusOne();
                    }
                }
            };
            Thread thread2 = new Thread(){
                @Override
                public void run(){
                    synchronized (plusMinus)
                    {
                        try {
                            Thread.sleep(10);
                        } catch (InterruptedException e) {
                            e.printStackTrace();
                        }
                    }
                }
            };
            plusThreads[i] = thread1;
            minusThreads[i] = thread2;
        }
        Arrays.parallelStream(plusThreads).forEach(Thread::start);
        Arrays.parallelStream(minusThreads).forEach(Thread::start);
    }
}
```

```

        plusMinus.minusOne();
    }
    });
    thread1.start();
    thread2.start();
    plusThreads[i]=thread1;
    minusThreads[i] = thread2;
}
for(Thread thread:plusThreads){ // ??????????
    try {
        thread.join();
    } catch (InterruptedException e) { // ??????????
        e.printStackTrace();
    }
}
for(Thread thread:minusThreads){
    try {
        thread.join();
    } catch (InterruptedException e) {
        e.printStackTrace();
    }
}
System.out.println("num equals " + plusMinus.printNum());
}
}

```

结果:

```

"C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2021.3.2\jbr\bin\java.exe"
"-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA 2021.3.2\license\third-party\IntelliJ
IDEA Community Edition 2021.3.2\lib\idea_rt.jar=62397:C:\Program Files\JetBrains\IntelliJ
IDEA 2021.3.2\license\third-party\IntelliJ IDEA Community Edition 2021.3.2\bin" -
Dfile.encoding=UTF-8 -classpath C:\Users\wzh\IdeaProjects\WEIZHI\out\production\WEIZHI
WEIZHI.TestPlusMinus
    num equals 1000

```

Process finished with exit code 0

Task9. 不修改 TestPlusMinus 测试类, 使用 synchronized 关键字修改 PlusMinus 基础类, 使得 num 值不会出现不同步的问题, 将修改后的 PlusMinus 基础类代码附在实验报告中。

代码:

```

package WEIZHI;
import java.util.Arrays;
import java.util.Scanner;
import java.io.PrintWriter;
import java.util.Calendar;
import java.util.Date;
import java.lang.Thread;

public class weilegezhi {
    public volatile int num;
    public synchronized void plusOne() {
        num = num + 1;
    }
    public synchronized void minusOne() {
        num = num - 1;
    }
    public int printNum() {
        return num;
    }
}

class TestPlusMinus {

```

结果:

```
"C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2021.3.2\jbr\bin\java.exe"
"-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA 2021.3.2\license\third-party\IntelliJ
IDEA Community Edition 2021.3.2\lib\idea_rt.jar=49162:C:\Program Files\JetBrains\IntelliJ
IDEA 2021.3.2\license\third-party\IntelliJ IDEA Community Edition 2021.3.2\bin" -
Dfile.encoding=UTF-8 -classpath C:\Users\wzh\IdeaProjects\WEIZHI\out\production\WEIZHI
WEIZHI.TestPlusMinus
    num equals 1000
```

Process finished with exit code 0

Task10. 设计 3 个线程发生死锁的场景并编写代码，将关键代码和结果附在实验报告中。

代码：

```
package WEIZHI;
import java.util.Arrays;
import java.util.Scanner;
import java.io.PrintWriter;
import java.util.Calendar;
import java.util.Date;
import java.lang.Thread;

class PlusMinus {
    public volatile int num;
    public void plusOne(){
        num = num + 1;
    }
    public void minusOne(){
        num = num - 1;
    }
    public int printNum(){
        return num;
    }
}

public class weilegezhi {
    public static void main(String[] args){
        PlusMinus plusMinus1 = new PlusMinus();
        plusMinus1.num = 1000;
        PlusMinus plusMinus2 = new PlusMinus();
        plusMinus2.num = 1000;
        PlusMinus plusMinus3 = new PlusMinus();
        plusMinus3.num = 1000;
        Thread thread1 = new Thread(){
            @Override
            public void run(){
                synchronized (plusMinus1){
                    System.out.println("thread1 正在占用 plusMinus1");
                    try {
                        sleep(1000);
                    } catch (InterruptedException e) {
                        e.printStackTrace();
                    }
                    System.out.println("thread1 试图继续占用 plusMinus2");
                    System.out.println("thread1 等待中...");
                    synchronized (plusMinus2){
                        System.out.println("thread1 成功占用了 plusMinus2 ");
                    }
                }
            }
        };
        thread1.start();
        Thread thread2 = new Thread(){
            @Override
            public void run(){
                synchronized (plusMinus2){
                    System.out.println("thread2 正在占用 plusMinus2");
                    try {
                        sleep(1000);
                    } catch (InterruptedException e) {
                        e.printStackTrace();
                    }
                }
                System.out.println("thread2 试图继续占用 plusMinus3");
            }
        };
        thread2.start();
    }
}
```

```

        synchronized (plusMinus3) {
            System.out.println("thread2 成功占用了 plusMinus3 ");
        }
    }
}
thread2.start();
Thread thread3 = new Thread()
{
    @Override
    public void run()
    {
        synchronized (plusMinus3)
        {
            System.out.println("thread3 正在占用 plusMinus3");
            try {
                sleep(1000);
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
            System.out.println("thread3 试图继续占用 plusMinus1");
            synchronized (plusMinus1)
            {
                System.out.println("thread3 成功占用了 plusMinus1 ");
            }
        }
    }
};
thread3.start();
}
}

```

结果:

"C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2021.3.2\jbr\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA 2021.3.2\license\third-party\IntelliJ IDEA Community Edition 2021.3.2\lib\idea\_rt.jar=49517:C:\Program Files\JetBrains\IntelliJ IDEA 2021.3.2\license\third-party\IntelliJ IDEA Community Edition 2021.3.2\bin" -Dfile.encoding=UTF-8 -classpath C:\Users\wzh\IdeaProjects\WEIZHI\out\production\WEIZHI WEIZHI.weilegezhi

```

thread1 正在占用 plusMinus1
thread3 正在占用 plusMinus3
thread2 正在占用 plusMinus2
thread1 试图继续占用 plusMinus2
thread1 等待中...
thread3 试图继续占用 plusMinus1
thread2 试图继续占用 plusMinus3

```

#### 4.4 线程交互

Task11. 阐述 synchronized 关键字在多线程方法手段作用, 然后运行本代码, 观察 CPU 的使用情况, 将实验结果和 CPU 使用情况截图附在实验报告中。

分析:

当我们不使用 synchronized 关键字, 结果中出现了以下情况:

```

num = 1000
num = 999
num = 999
num = 998
num = 997

```

.....

可见：999 连续出现了两次，而正常来说，num 的值要么加 1，要么减 1。

以上现象说明：synchronized 关键字的作用是防止一个线程运行时另一个线程也在访问该线程正在访问的变量。

部分 CPU 使用情况：（Help->Diagnostic Tools->Activity Monitor...）

%CPU Subsystem

6.2 <Process total CPU usage>

3.8 <infrastructure: AWT-Windows>

1.2 <JIT compiler>

0.7 ui.tabs.impl (in com.intellij)

0.7 <unidentified: JobScheduler FJ pool>

Task12. 在 Task11 的基础上增加若干减 1 操作线程，运行久一点，观察有没有发生错误。若有，请分析错误原因，并给出解决方法。

实验中发生了错误。

在测试过程中，出现了如下现象：

.....

num = 2

num = 1

num = 0

num = -1

num = -2

.....

照理来说，当 num=1 时减法进程不应该运行，num 也不应该从 1 变成 0。

可能原因：运行过程中发生了线程不同步。如：线程 3 和线程 5 同为减法线程。由于程序中 num 并没有被 synchronized 且减法线程运行较快，线程 3 开始运行后还未结束前线程 5 也开始运行并访问了 num，此时 num 还是 2。当线程 3 把 num 从 2 减为 1 时，线程 5 中的循环节

```
if(plusMinus.num == 1){
    continue;
}
```

已经执行完毕。线程 5 遂将 num 的值从 1 减为 0。

解决方法：在每一个减法线程中使用 synchronized 等关键字，让线程同步不可能发生。

Task13. 将 PlusMinus02 类中的 volatile 关键字删除，观察实验结果变化，将实验结果附在实验报告中并分析 volatile 关键字的作用。

删除 volatile 关键字前，在 num 第一次被减到 1 之后，发生了如下现象：

.....

num = 2

num = 1

num = 2

num = 1

.....

可见，此时线程 1 和线程 2 交替运行。

而删除 volatile 关键字前，在 num 第一次被减到 1 之后，发生了如下现象：

.....

num = 2

num = 1

num = 2

```
num = 3
num = 4
num = 5
.....
```

可见，在 num 第一次被减到 1 之后，就几乎只有线程 1 在运行。

以上现象说明，volatile 关键字的作用是让一个线程在另一个线程对变量进行修改后立即看到修改的值，如本实验中，当线程 2 把 num 从 1 加到 2 时线程 1 立即注意到这种变化。否则，线程 1 就不会再次被运行。此外，volatile 关键字还保证程序按照代码的顺序执行。

Task14. 使用 wait 和 notify 修改下面的代码段，使之也达到 4.4 开头中描述的功能，并将修改后的完整段和实验结果以及 CPU 占用情况截图附在实验报告中。

代码：

```
package WEIZHI;
import java.util.Arrays;
import java.util.Scanner;
import java.io.PrintWriter;
import java.util.Calendar;
import java.util.Date;
import java.lang.Thread;
import java.lang.Object;

class PlusMinus02 {
    public volatile int num;
    public volatile int s;
    public void plusOne() {
        synchronized (this) {
            if (this.s == 1)
            {
                try {
                    wait();
                } catch (InterruptedException e) {
                    e.printStackTrace();
                }
            }
            else
            {
                notify();
            }
            num = num + 1;
            s = 1;
            printNum();
        }
    }
    public void minusOne() {
        synchronized (this) {
            if (this.num == 1)
            {
                try {
                    wait();
                } catch (InterruptedException e) {
                    e.printStackTrace();
                }
            }
            else
            {
                notify();
            }
            num = num - 1;
            s = 0;
            printNum();
        }
    }
}
```



```
    }}  
    public void printNum(){  
        System.out.println("num = " + num);  
    }  
}  
public class weilegezhi {  
    public static void main(String[] args){  
        PlusMinus02 plusMinus = new PlusMinus02();  
        plusMinus.num = 1000;  
        Thread thread1 = new Thread(){  
            @Override  
            public void run(){  
                while (true){  
                    plusMinus.minusOne();  
                    try {  
                        sleep(10);  
                    } catch (InterruptedException e) {  
                        e.printStackTrace();  
                    }  
                }  
            }  
        };  
        thread1.start();  
        Thread thread2 = new Thread(){  
            @Override  
            public void run(){  
                while (true){  
                    plusMinus.plusOne();  
                    try {  
                        sleep(100);  
                    } catch (InterruptedException e) {  
                        e.printStackTrace();  
                    }  
                }  
            }  
        };  
        thread2.start();  
    }  
}
```

结果:

"C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2021.3.2\jbr\bin\java.exe" "-  
javaagent:C:\Program Files\JetBrains\IntelliJ IDEA 2021.3.2\license\third-party\IntelliJ IDEA  
Community Edition 2021.3.2\lib\idea\_rt.jar=57366:C:\Program Files\JetBrains\IntelliJ IDEA  
2021.3.2\license\third-party\IntelliJ IDEA Community Edition 2021.3.2\bin" -Dfile.encoding=UTF-  
8 -classpath C:\Users\wzh\IdeaProjects\WEIZHI\out\production\WEIZHI WEIZHI.weilegezhi

num = 999  
num = 1000  
num = 999  
num = 998  
num = 997  
num = 996  
num = 995  
num = 994  
num = 995  
num = 994  
num = 993  
num = 992  
num = 991  
num = 990  
num = 989

num = 988  
num = 989  
num = 988  
num = 987  
num = 986  
num = 985  
num = 984  
num = 983  
num = 984  
num = 983  
num = 982  
num = 981  
num = 980  
num = 979  
num = 978  
num = 977  
num = 976  
num = 977  
num = 976  
num = 975  
num = 974  
num = 973  
num = 972  
num = 971  
num = 970  
num = 971  
num = 970  
num = 969  
num = 968  
num = 967  
num = 966  
num = 965  
num = 964  
num = 965  
num = 964  
num = 963  
num = 962  
num = 961  
num = 960  
num = 959  
num = 958  
num = 959  
num = 958  
num = 957  
num = 956  
num = 955  
num = 954  
num = 953  
num = 952  
num = 953

num = 952  
num = 951  
num = 950  
num = 949  
num = 948  
num = 947  
num = 946  
num = 947  
num = 946  
num = 945  
num = 944  
num = 943  
num = 942  
num = 941  
num = 940  
num = 941  
num = 940  
num = 939  
num = 938  
num = 937  
num = 936  
num = 935  
num = 934  
num = 935  
num = 934  
num = 933  
num = 932  
num = 931  
num = 930  
num = 929  
num = 928  
num = 929  
num = 928  
num = 927  
num = 926  
num = 925  
num = 924  
num = 923  
num = 924  
num = 923  
num = 922  
num = 921  
num = 920  
num = 919  
num = 918  
num = 917  
num = 916  
num = 917  
num = 916  
num = 915

num = 914  
num = 913  
num = 912  
num = 911  
num = 912  
num = 911  
num = 910  
num = 909  
num = 908  
num = 907  
num = 906  
num = 905  
num = 904  
num = 905  
num = 904  
num = 903  
num = 902  
num = 901  
num = 900  
num = 899  
num = 898  
num = 899  
num = 898  
num = 897  
num = 896  
num = 895  
num = 894  
num = 893  
num = 894  
num = 893  
num = 892  
num = 891  
num = 890  
num = 889  
num = 888  
num = 887  
num = 886  
num = 887  
num = 886  
num = 885  
num = 884  
num = 883  
num = 882  
num = 881  
num = 882  
num = 881  
num = 880  
num = 879  
num = 878  
num = 877

num = 876  
num = 875  
num = 876  
num = 875  
num = 874  
num = 873  
num = 872  
num = 871  
num = 870  
num = 869  
num = 870  
num = 869  
num = 868  
num = 867  
num = 866  
num = 865  
num = 864  
num = 863  
num = 864  
num = 863  
num = 862  
num = 861  
num = 860  
num = 859  
num = 858  
num = 857  
num = 858  
num = 857  
num = 856  
num = 855  
num = 854  
num = 853  
num = 852  
num = 851  
num = 850  
num = 851  
num = 850  
num = 849  
num = 848  
num = 847  
num = 846  
num = 845  
num = 844  
num = 845  
num = 844  
num = 843  
num = 842  
num = 841  
num = 840  
num = 839

num = 838  
num = 839  
num = 838  
num = 837  
num = 836  
num = 835  
num = 834  
num = 833  
num = 834  
num = 833  
num = 832  
num = 831  
num = 830  
num = 829  
num = 828  
num = 827  
num = 828  
num = 827  
num = 826  
num = 825  
num = 824  
num = 823  
num = 822  
num = 821  
num = 822  
num = 821  
num = 820  
num = 819  
num = 818  
num = 817  
num = 816  
num = 815  
num = 814  
num = 815  
num = 814  
num = 813  
num = 812  
num = 811  
num = 810  
num = 809  
num = 808  
num = 809  
num = 808  
num = 807  
num = 806  
num = 805  
num = 804  
num = 803  
num = 802  
num = 803

num = 802  
num = 801  
num = 800  
num = 799  
num = 798  
num = 797  
num = 798  
num = 797  
num = 796  
num = 795  
num = 794  
num = 793  
num = 792  
num = 791  
num = 792  
num = 791  
num = 790  
num = 789  
num = 788  
num = 787  
num = 786  
num = 785  
num = 786  
num = 785  
num = 784  
num = 783  
num = 782  
num = 781  
num = 780  
num = 779  
num = 780  
num = 779  
num = 778  
num = 777  
num = 776  
num = 775  
num = 774  
num = 773  
num = 774  
num = 773  
num = 772  
num = 771  
num = 770  
num = 769  
num = 768  
num = 767  
num = 766  
num = 767  
num = 766  
num = 765

num = 764  
num = 763  
num = 762  
num = 761  
num = 762  
num = 761  
num = 760  
num = 759  
num = 758  
num = 757  
num = 756  
num = 755  
num = 756  
num = 755  
num = 754  
num = 753  
num = 752  
num = 751  
num = 750  
num = 749  
num = 748  
num = 749  
num = 748  
num = 747  
num = 746  
num = 745  
num = 744  
num = 743  
num = 742  
num = 743  
num = 742  
num = 741  
num = 740  
num = 739  
num = 738  
num = 737  
num = 738  
num = 737  
num = 736  
num = 735  
num = 734  
num = 733  
num = 732  
num = 731  
num = 732  
num = 731  
num = 730  
num = 729  
num = 728  
num = 727



num = 726  
num = 725  
num = 726  
num = 725  
num = 724  
num = 723  
num = 722  
num = 721  
num = 720  
num = 719  
num = 718  
num = 719  
num = 718  
num = 717  
num = 716  
num = 715  
num = 714  
num = 713  
num = 714  
num = 713  
num = 712  
num = 711  
num = 710  
num = 709  
num = 708  
num = 707  
num = 708  
num = 707  
num = 706  
num = 705  
num = 704  
num = 703  
num = 702  
num = 701  
num = 700  
num = 701  
num = 700  
num = 699  
num = 698  
num = 697  
num = 696  
num = 695  
num = 694  
num = 695  
num = 694  
num = 693  
num = 692  
num = 691  
num = 690  
num = 689

num = 688  
num = 689  
num = 688  
num = 687  
num = 686  
num = 685  
num = 684  
num = 683  
num = 684  
num = 683  
num = 682  
num = 681  
num = 680  
num = 679  
num = 678  
num = 677  
num = 676  
num = 677  
num = 676  
num = 675  
num = 674  
num = 673  
num = 672  
num = 671  
num = 672  
num = 671  
num = 670  
num = 669  
num = 668  
num = 667  
num = 666  
num = 665  
num = 664  
num = 665  
num = 664  
num = 663  
num = 662  
num = 661  
num = 660  
num = 659  
num = 658  
num = 659  
num = 658  
num = 657  
num = 656  
num = 655  
num = 654  
num = 653  
num = 652  
num = 653

num = 652  
num = 651  
num = 650  
num = 649  
num = 648  
num = 647  
num = 646  
num = 647  
num = 646  
num = 645  
num = 644  
num = 643  
num = 642  
num = 641  
num = 640  
num = 641  
num = 640  
num = 639  
num = 638  
num = 637  
num = 636  
num = 635  
num = 634  
num = 635  
num = 634  
num = 633  
num = 632  
num = 631  
num = 630  
num = 629  
num = 628  
num = 629  
num = 628  
num = 627  
num = 626  
num = 625  
num = 624  
num = 623  
num = 622  
num = 623  
num = 622  
num = 621  
num = 620  
num = 619  
num = 618  
num = 617  
num = 616  
num = 617  
num = 616  
num = 615

num = 614  
num = 613  
num = 612  
num = 611  
num = 610  
num = 611  
num = 610  
num = 609  
num = 608  
num = 607  
num = 606  
num = 605  
num = 604  
num = 605  
num = 604  
num = 603  
num = 602  
num = 601  
num = 600  
num = 599  
num = 600  
num = 599  
num = 598  
num = 597  
num = 596  
num = 595  
num = 594  
num = 593  
num = 594  
num = 593  
num = 592  
num = 591  
num = 590  
num = 589  
num = 588  
num = 587  
num = 586  
num = 587  
num = 586  
num = 585  
num = 584  
num = 583  
num = 582  
num = 581  
num = 580  
num = 581  
num = 580  
num = 579  
num = 578  
num = 577

num = 576  
num = 575  
num = 574  
num = 575  
num = 574  
num = 573  
num = 572  
num = 571  
num = 570  
num = 569  
num = 568  
num = 569  
num = 568  
num = 567  
num = 566  
num = 565  
num = 564  
num = 563  
num = 562  
num = 563  
num = 562  
num = 561  
num = 560  
num = 559  
num = 558  
num = 557  
num = 556  
num = 557  
num = 556  
num = 555  
num = 554  
num = 553  
num = 552  
num = 551  
num = 550  
num = 551  
num = 550  
num = 549  
num = 548  
num = 547  
num = 546  
num = 545  
num = 544  
num = 545  
num = 544  
num = 543  
num = 542  
num = 541  
num = 540  
num = 539

num = 538  
num = 539  
num = 538  
num = 537  
num = 536  
num = 535  
num = 534  
num = 533  
num = 532  
num = 533  
num = 532  
num = 531  
num = 530  
num = 529  
num = 528  
num = 527  
num = 526  
num = 527  
num = 526  
num = 525  
num = 524  
num = 523  
num = 522  
num = 521  
num = 520  
num = 521  
num = 520  
num = 519  
num = 518  
num = 517  
num = 516  
num = 515  
num = 514  
num = 515  
num = 514  
num = 513  
num = 512  
num = 511  
num = 510  
num = 509  
num = 508  
num = 509  
num = 508  
num = 507  
num = 506  
num = 505  
num = 504  
num = 503  
num = 502  
num = 503

num = 502  
num = 501  
num = 500  
num = 499  
num = 498  
num = 497  
num = 498  
num = 497  
num = 496  
num = 495  
num = 494  
num = 493  
num = 492  
num = 491  
num = 492  
num = 491  
num = 490  
num = 489  
num = 488  
num = 487  
num = 486  
num = 485  
num = 484  
num = 485  
num = 484  
num = 483  
num = 482  
num = 481  
num = 480  
num = 479  
num = 480  
num = 479  
num = 478  
num = 477  
num = 476  
num = 475  
num = 474  
num = 473  
num = 474  
num = 473  
num = 472  
num = 471  
num = 470  
num = 469  
num = 468  
num = 467  
num = 468  
num = 467  
num = 466  
num = 465

num = 464  
num = 463  
num = 462  
num = 461  
num = 460  
num = 461  
num = 460  
num = 459  
num = 458  
num = 457  
num = 456  
num = 455  
num = 454  
num = 455  
num = 454  
num = 453  
num = 452  
num = 451  
num = 450  
num = 449  
num = 448  
num = 449  
num = 448  
num = 447  
num = 446  
num = 445  
num = 444  
num = 443  
num = 444  
num = 443  
num = 442  
num = 441  
num = 440  
num = 439  
num = 438  
num = 437  
num = 436  
num = 437  
num = 436  
num = 435  
num = 434  
num = 433  
num = 432  
num = 431  
num = 432  
num = 431  
num = 430  
num = 429  
num = 428  
num = 427



num = 426  
num = 425  
num = 424  
num = 425  
num = 424  
num = 423  
num = 422  
num = 421  
num = 420  
num = 419  
num = 420  
num = 419  
num = 418  
num = 417  
num = 416  
num = 415  
num = 414  
num = 413  
num = 414  
num = 413  
num = 412  
num = 411  
num = 410  
num = 409  
num = 408  
num = 407  
num = 408  
num = 407  
num = 406  
num = 405  
num = 404  
num = 403  
num = 402  
num = 401  
num = 402  
num = 401  
num = 400  
num = 399  
num = 398  
num = 397  
num = 396  
num = 395  
num = 394  
num = 395  
num = 394  
num = 393  
num = 392  
num = 391  
num = 390  
num = 389

num = 388  
num = 389  
num = 388  
num = 387  
num = 386  
num = 385  
num = 384  
num = 383  
num = 382  
num = 383  
num = 382  
num = 381  
num = 380  
num = 379  
num = 378  
num = 377  
num = 376  
num = 377  
num = 376  
num = 375  
num = 374  
num = 373  
num = 372  
num = 371  
num = 370  
num = 371  
num = 370  
num = 369  
num = 368  
num = 367  
num = 366  
num = 365  
num = 364  
num = 365  
num = 364  
num = 363  
num = 362  
num = 361  
num = 360  
num = 359  
num = 358  
num = 359  
num = 358  
num = 357  
num = 356  
num = 355  
num = 354  
num = 353  
num = 352  
num = 353

num = 352  
num = 351  
num = 350  
num = 349  
num = 348  
num = 347  
num = 348  
num = 347  
num = 346  
num = 345  
num = 344  
num = 343  
num = 342  
num = 341  
num = 340  
num = 341  
num = 340  
num = 339  
num = 338  
num = 337  
num = 336  
num = 335  
num = 334  
num = 335  
num = 334  
num = 333  
num = 332  
num = 331  
num = 330  
num = 329  
num = 328  
num = 329  
num = 328  
num = 327  
num = 326  
num = 325  
num = 324  
num = 323  
num = 324  
num = 323  
num = 322  
num = 321  
num = 320  
num = 319  
num = 318  
num = 317  
num = 318  
num = 317  
num = 316  
num = 315

num = 314  
num = 313  
num = 312  
num = 311  
num = 312  
num = 311  
num = 310  
num = 309  
num = 308  
num = 307  
num = 306  
num = 305  
num = 304  
num = 305  
num = 304  
num = 303  
num = 302  
num = 301  
num = 300  
num = 299  
num = 298  
num = 299  
num = 298  
num = 297  
num = 296  
num = 295  
num = 294  
num = 293  
num = 292  
num = 293  
num = 292  
num = 291  
num = 290  
num = 289  
num = 288  
num = 287  
num = 288  
num = 287  
num = 286  
num = 285  
num = 284  
num = 283  
num = 282  
num = 281  
num = 282  
num = 281  
num = 280  
num = 279  
num = 278  
num = 277

num = 276  
num = 275  
num = 274  
num = 275  
num = 274  
num = 273  
num = 272  
num = 271  
num = 270  
num = 269  
num = 268  
num = 269  
num = 268  
num = 267  
num = 266  
num = 265  
num = 264  
num = 263  
num = 264  
num = 263  
num = 262  
num = 261  
num = 260  
num = 259  
num = 258  
num = 257  
num = 256  
num = 257  
num = 256  
num = 255  
num = 254  
num = 253  
num = 252  
num = 251  
num = 252  
num = 251  
num = 250  
num = 249  
num = 248  
num = 247  
num = 246  
num = 245  
num = 244  
num = 245  
num = 244  
num = 243  
num = 242  
num = 241  
num = 240  
num = 239

num = 238  
num = 239  
num = 238  
num = 237  
num = 236  
num = 235  
num = 234  
num = 233  
num = 234  
num = 233  
num = 232  
num = 231  
num = 230  
num = 229  
num = 228  
num = 227  
num = 226  
num = 227  
num = 226  
num = 225  
num = 224  
num = 223  
num = 222  
num = 221  
num = 220  
num = 221  
num = 220  
num = 219  
num = 218  
num = 217  
num = 216  
num = 215  
num = 216  
num = 215  
num = 214  
num = 213  
num = 212  
num = 211  
num = 210  
num = 209  
num = 210  
num = 209  
num = 208  
num = 207  
num = 206  
num = 205  
num = 204  
num = 203  
num = 204  
num = 203

num = 202  
num = 201  
num = 200  
num = 199  
num = 198  
num = 197  
num = 198  
num = 197  
num = 196  
num = 195  
num = 194  
num = 193  
num = 192  
num = 191  
num = 190  
num = 191  
num = 190  
num = 189  
num = 188  
num = 187  
num = 186  
num = 185  
num = 184  
num = 185  
num = 184  
num = 183  
num = 182  
num = 181  
num = 180  
num = 179  
num = 178  
num = 179  
num = 178  
num = 177  
num = 176  
num = 175  
num = 174  
num = 173  
num = 172  
num = 173  
num = 172  
num = 171  
num = 170  
num = 169  
num = 168  
num = 167  
num = 166  
num = 167  
num = 166  
num = 165

num = 164  
num = 163  
num = 162  
num = 161  
num = 162  
num = 161  
num = 160  
num = 159  
num = 158  
num = 157  
num = 156  
num = 155  
num = 154  
num = 155  
num = 154  
num = 153  
num = 152  
num = 151  
num = 150  
num = 149  
num = 148  
num = 149  
num = 148  
num = 147  
num = 146  
num = 145  
num = 144  
num = 143  
num = 144  
num = 143  
num = 142  
num = 141  
num = 140  
num = 139  
num = 138  
num = 137  
num = 138  
num = 137  
num = 136  
num = 135  
num = 134  
num = 133  
num = 132  
num = 131  
num = 132  
num = 131  
num = 130  
num = 129  
num = 128  
num = 127



num = 126  
num = 125  
num = 126  
num = 125  
num = 124  
num = 123  
num = 122  
num = 121  
num = 120  
num = 119  
num = 118  
num = 119  
num = 118  
num = 117  
num = 116  
num = 115  
num = 114  
num = 113  
num = 112  
num = 113  
num = 112  
num = 111  
num = 110  
num = 109  
num = 108  
num = 107  
num = 106  
num = 107  
num = 106  
num = 105  
num = 104  
num = 103  
num = 102  
num = 101  
num = 102  
num = 101  
num = 100  
num = 99  
num = 98  
num = 97  
num = 96  
num = 95  
num = 94  
num = 95  
num = 94  
num = 93  
num = 92  
num = 91  
num = 90  
num = 89

num = 90  
num = 89  
num = 88  
num = 87  
num = 86  
num = 85  
num = 84  
num = 83  
num = 82  
num = 83  
num = 82  
num = 81  
num = 80  
num = 79  
num = 78  
num = 77  
num = 76  
num = 77  
num = 76  
num = 75  
num = 74  
num = 73  
num = 72  
num = 71  
num = 72  
num = 71  
num = 70  
num = 69  
num = 68  
num = 67  
num = 66  
num = 65  
num = 66  
num = 65  
num = 64  
num = 63  
num = 62  
num = 61  
num = 60  
num = 59  
num = 58  
num = 59  
num = 58  
num = 57  
num = 56  
num = 55  
num = 54  
num = 53  
num = 52  
num = 53

num = 52  
num = 51  
num = 50  
num = 49  
num = 48  
num = 47  
num = 48  
num = 47  
num = 46  
num = 45  
num = 44  
num = 43  
num = 42  
num = 41  
num = 40  
num = 41  
num = 40  
num = 39  
num = 38  
num = 37  
num = 36  
num = 35  
num = 34  
num = 35  
num = 34  
num = 33  
num = 32  
num = 31  
num = 30  
num = 29  
num = 28  
num = 29  
num = 28  
num = 27  
num = 26  
num = 25  
num = 24  
num = 23  
num = 22  
num = 23  
num = 22  
num = 21  
num = 20  
num = 19  
num = 18  
num = 17  
num = 18  
num = 17  
num = 16  
num = 15

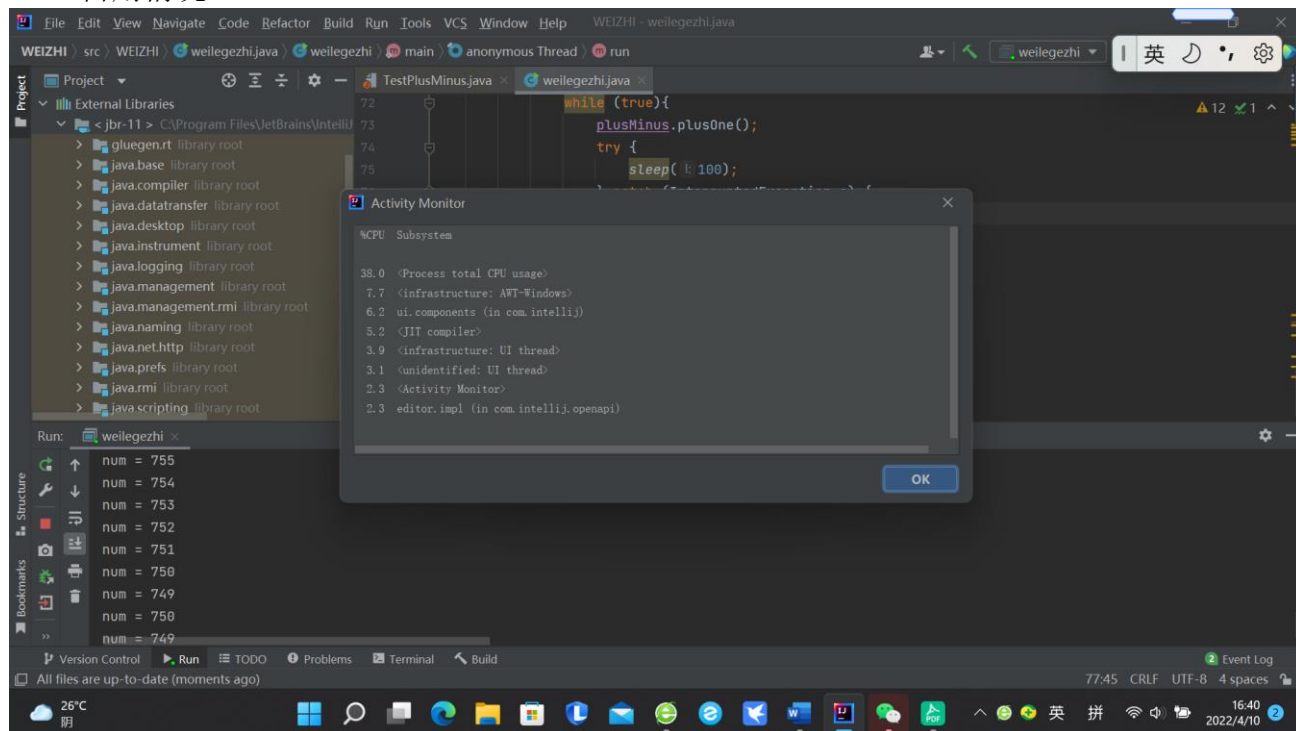
```

num = 14
num = 13
num = 12
num = 11
num = 12
num = 11
num = 10
num = 9
num = 8
num = 7
num = 6
num = 5
num = 6
num = 5
num = 4
num = 3
num = 2
num = 1
num = 2
num = 1
num = 2
num = 1
num = 2
num = 1
num = 1
num = 2
num = 1
.....

```

Process finished with exit code 130

CPU 占用情况:



## 五、总结

通过这两周的实验，我大致了解了多线程编程的基本知识，初步掌握了多线程编程的基本技能，为今后的实验、课程项目打下了基础。