**集美大学计算机工程学院实验报告**

|  |  |  |
| --- | --- | --- |
| **课程名称**：Java程序设计 | **班级**：软件1912 | **实验成绩**： |
| **指导教师**：李　旺 | **姓名**：游嘉升 |  |
| **实验项目名称**：**类与对象** | **学号**：201921122050 | **上机实践日期**：10月11日 |
| **实验项目编号**： | **组号**： | **上机实践时间**： 2 学时 |

**一、目的**

掌握类的定义，构造方法，初始化；理解类与对象，并能够加以区分。

**二、实验内容与设计思想**

1.Person类的定义与使用

(1)定义一个有关人的Person类，内含姓名name、性别sex、年龄age属性，所有的变量必须是私有的(private)。

(2)设计一无参构造函数初始化Person的age属性为0、sex属性为0(或者其他，视定义的数据类型而定)、name属性为空字符串("")。

(3)使用这些构造方法对Person类进行初始化。

(4)其他类只能通过该类的setter、getter方法修改与获取属性。

(5)改写toString()方法，显示其所有属性。另外定义测试类TestPerson，对Person类进行测试。

**思考：private修饰的属性，访问起来到底有何限制？如果Person p = new Person();可以直接使用p.name访问p的name属性吗？**

2. 类的简单使用

定义一个类LeapYear，其中有一个**非静态方法**public boolean isLeapYear(int year)，用于判断年份year是否为闰年。

编写测试类，在程序主入口main()方法中创建一个LeapYear对象，调用isLeapYear方法并输出（测试数据自定，也可以由键盘输入）。

**思考：静态方法与非静态方法使用起来有什么区别？**

3. 三角形与圆形

(1)编写长方形类Rectangle、三角形类Triangle、圆形类Circle，各有各的属性。

(2)在三个类中分别定义求周长的方法public double getPerimeter()、求面积的方法getArea()。

(3)新建测试类TestShape，针对每个类建立两个对象(共六个对象，成员值自定或随机)，并将这些对象的所有面积与周长相加统计输出。

**思考：对面积、周长进行相加统计的代码是否过于复杂？有无改进空间？**

4. 初始化块

(1)为题1中的Person类添加两个属性：身份证号int id(不要求与真正的身份证类似，这里的id只是一个简单的整型值，每个Person对象均不同即可。如，可以从1开始编号，第一个Person对象的id为1，第二个为2，以此类推)，静态变量(类变量)static int count(用于记录身份证号自动增长的历史)。

(2)在构造方法中对id进行初始化，并使得每个生成对象的id均不同。

(3)为Person类创建两个构造方法：第一个为包含两个参数的构造方法，可根据提供的“name”、“age”参数，对姓名、年龄属性进行初始化；第二个构造函数包含三个参数name,age,sex，使用构造方法重载技术，调用第一个构造方法，并额外对sex属性进行初始化。

(4)覆盖Person类中的toString方法，输出Person的基本信息。选择适当的测试语句，测试本题中的改动。

5. 二次方程式求解

为二次方程ax2+bx+c=0设计一个名为QuadraticEquation的类。这个类包括：

(1)代表三个系数的私有数据成员a,b和c；

(2)一个参数为a,b,c的构造方法；

(3)一个名为getDiscriminant()的方法，返回判别式：b2-4ac；

(4)名为getRoot1()和getRoot2()的方法，返回二次方程的两个根：r1=(-b+(b2-4ac)1/2)/2a，r2=(-b-(b2-4ac)1/2)/2a。这两个方法只有在判别式为非负数时才有用。如果判别式为负，则返回0。

实现这个类。编写一个测试程序，提示用户输入a,b和c的值，然后显示判别式的值。如果判别式为正数，显示两个根；如果判别式为0，则显示一个根，否则显示"The equation has no roots."字样。

6. 矩阵四则运算

定义矩阵类Matrix，包括：

(1)代表矩阵的行数rows(或m)、列数cols(或n)，以及二维数组data；

(2)一个参数为rows,cols的构造方法，实现初始化操作，并将矩阵元素全部置为0；

(3)public void setElement(int row, int col, double value);方法，用于设置第row行，第col列元素的值；

(4)public Matrix add(Matrix m);方法，实现当前矩阵与m矩阵相加，并返回新的矩阵；若无法相加，则返回null；

(5)public Matrix minus(Matrix m);方法，实现当前矩阵减去m矩阵，并返回新的矩阵；若无法相减，则返回null；

(6)public Matrix multiple(Matrix m);方法，实现当前矩阵乘以m矩阵，并返回新的矩阵；若无法相乘，则返回null；

(7)public Matrix transposition();方法，实现矩阵转置，并返回新的矩阵；

(8)public void display();方法，打印当前矩阵。

实现该类。编写一个测试程序，随机生成矩阵元素或者由程序中用常量设置（可不必由键盘输入），测试上述四则运算，打印运算结果。

7. 学生成绩管理

实验2中WriteReadFileTest.java中实现了对学生的信息存储与读取，但不是采用面向对象的编程方法。现在要求通过采用面向对象的方法改造该程序。

(1)创建Student类，包含name,math,java,ds,avg,total等属性(其中，avg和total成员可以写，也可以省略，具体看程序中如何处理)，并编写器setter/getter方法（注意相应的avg与total的getter/setter方法的编写）。并编写两个构造函数，一个构造函数包含除avg,total外的所有属性，另一个构造函数包含所有属性。编写相应的toString()方法。

(2)创建一Students.txt文件，文件格式如下：

张三 90 95 98 0 0

李四 80 90 100 0 0

(3)使用上一次的编写的public static String[] readStudentsFromFile(String fileName)，把文件中的信息读取到String数组中。

(4)编写public static Student[] makeStudentFromString(String[] students)，实现将字符串转化成学生对象。

(5)编写一个方法对返回的Student数组中的所有学生的平均成绩进行排序，并输出。

提示：(3)(4)可以合并，方法签名改为：public static Student[] readStudentsFromFile(String fileName)。

**三、实验使用环境**

Java version "13.0.2"

IntelliJ IDEA 2020.2.2 x64

**四、实验步骤和调试过程**

1.Person类的定义与使用

4. 初始化块

具体代码见附件person

关键代码：



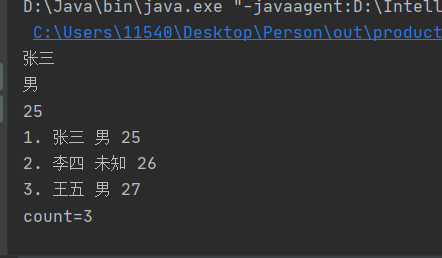
测试数据：

张三 1 25

李四 null 26

王五 27 1

测试结果：



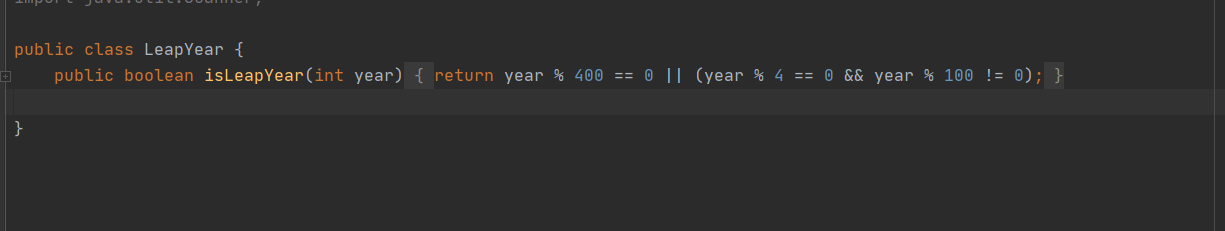
思考：private修饰的属性，访问起来到底有何限制？如果Person p = new Person();可以直接使用p.name访问p的name属性吗？

修饰词private修饰的某个属性，限制了不同类之间的访问，但是并没有限制同类对象之间的访问。在main方法或者其他不同类方法中不可以直接使用p.name访问p的name属性，在同类的Person对象里就可以访问。

2. 类的简单使用

代码见附件leapyear

关键代码：



测试数据：

2020

测试结果：



思考：静态方法与非静态方法使用起来有什么区别？

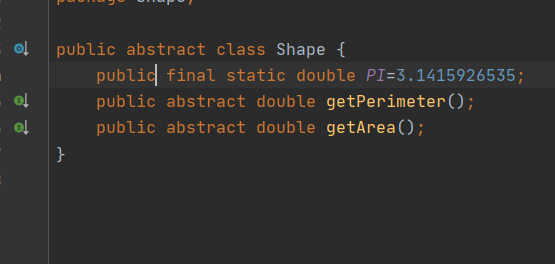
静态方法在此类装载的时候就已经生成且生存周期一直到程序结束，非静态方法在对象生成时生成，对象回收时伴随对象回收一起销毁。

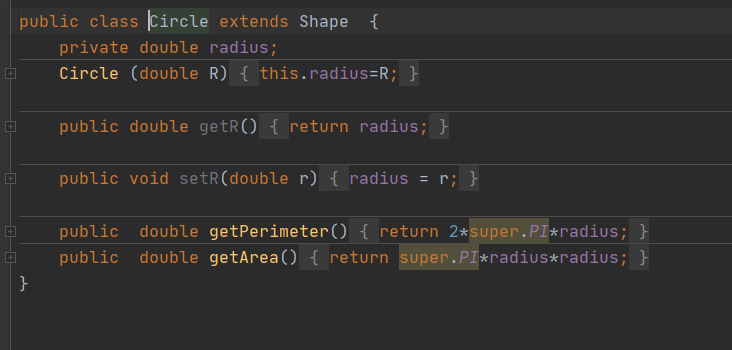
3. 三角形与圆形

代码见附件shape

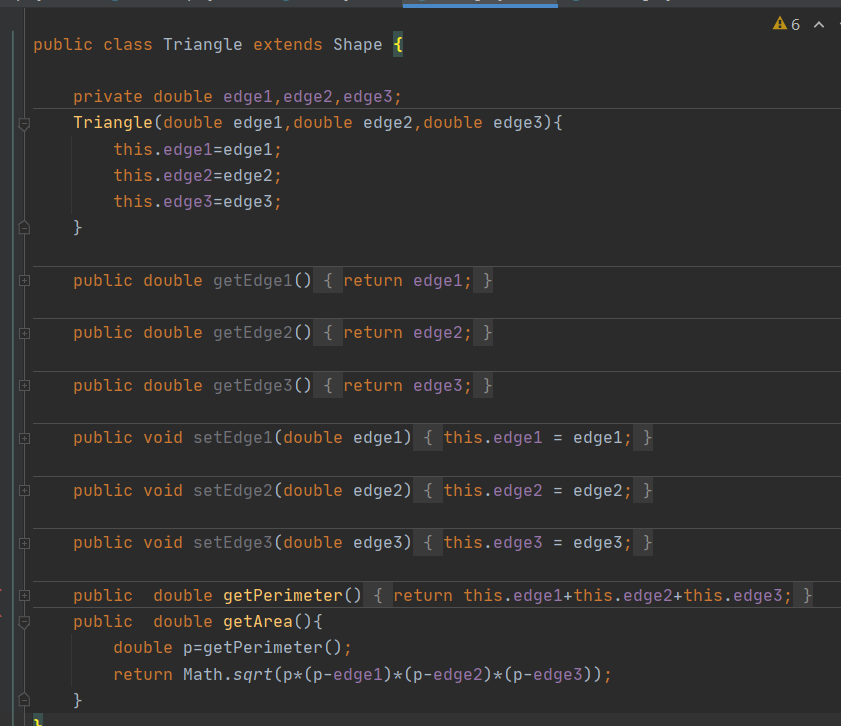
设计思想：定义一个超类shape，再定义三个子类图形继承和多态它的定义，达到可以统一接口的目的，进而简化代码。

关键代码：



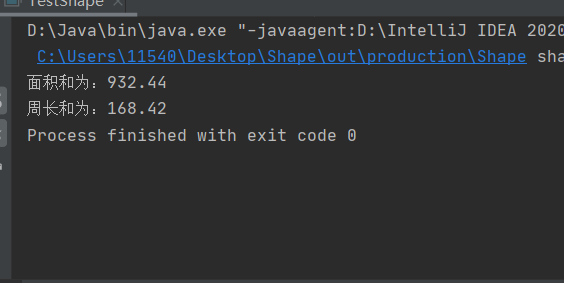






测试数据：各数据由随机函数随机生成

测试结果：



思考：对面积、周长进行相加统计的代码是否过于复杂？有无改进空间？

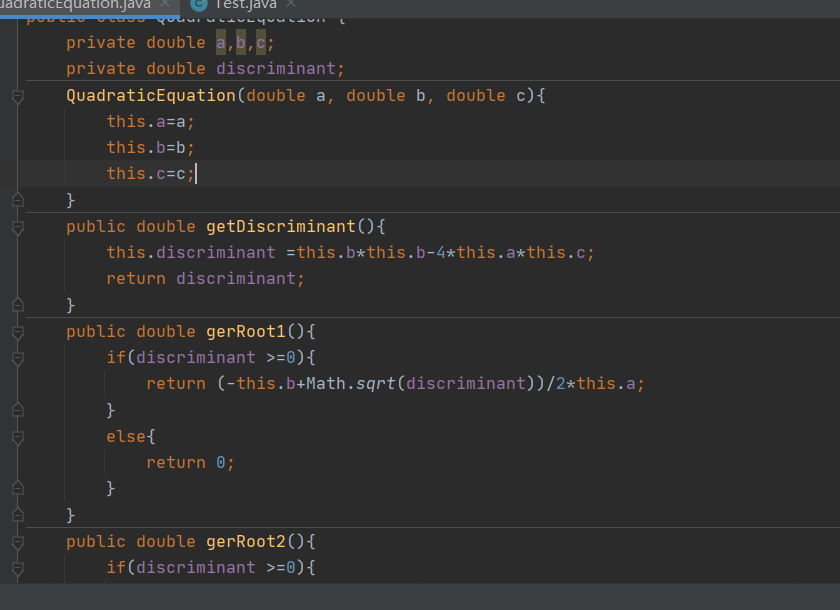
定义一个Shape类包含其他三个形状的类，定义了两个个方法可调用三个形状类各自的周长面积方法，统一了接口，就可以在一个循环里直接调用Shape类的方法进行值的加减。

5. 二次方程式求解

代码见附件quadraticequation

设计思想：设计一个类，包含多种方法计算方程的判别式与根

关键代码：



测试数据：

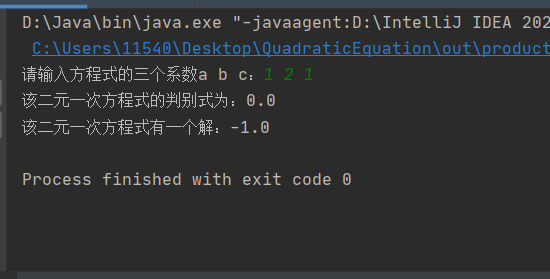
1 2 1（方程一个根

-1 -2 -3（方程无根

1 4 3（方程有两个根

测试结果：

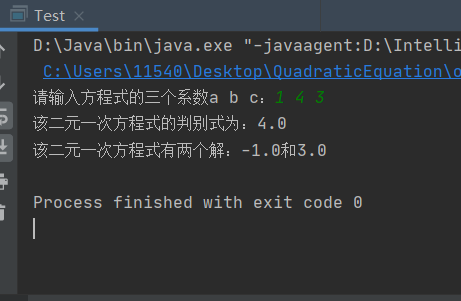
方程一个根：



方程无根：



方程有两个根：



6. 矩阵四则运算

代码见附件Matrix

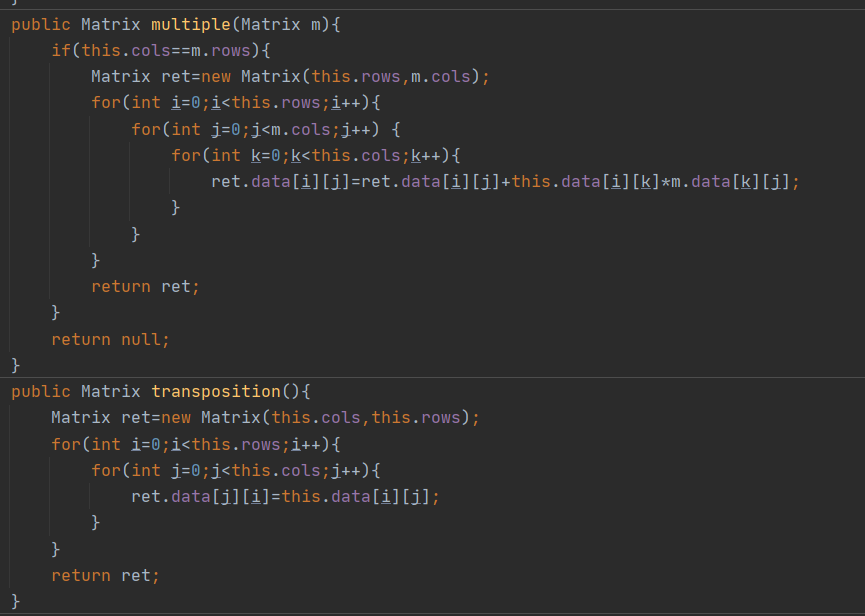
设计思想：设计一个类，包含多种计算加减，乘法，转置，以及输出矩阵的方法。

关键代码：

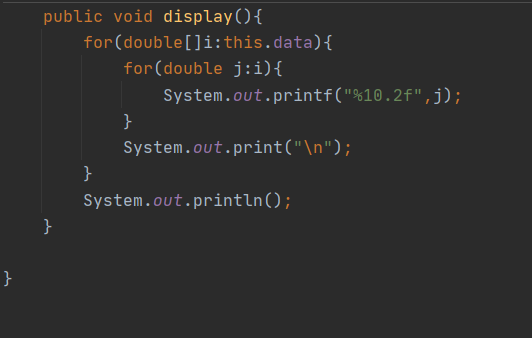
矩阵加减法：



矩阵乘法与转置：

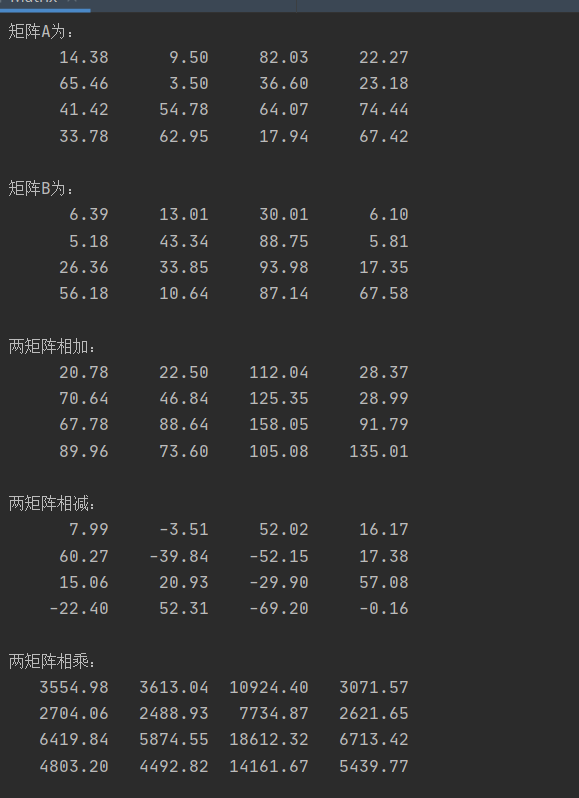


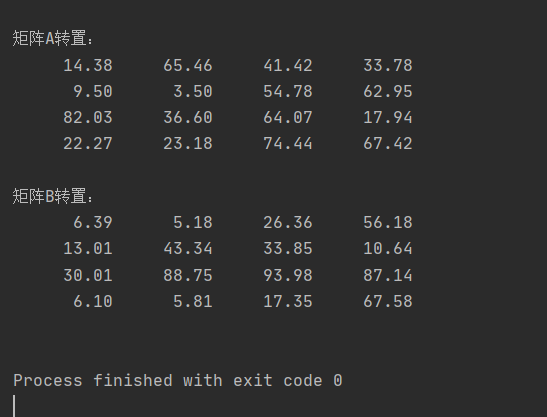
矩阵输出：



测试数据：由随机函数随机生成

测试结果：



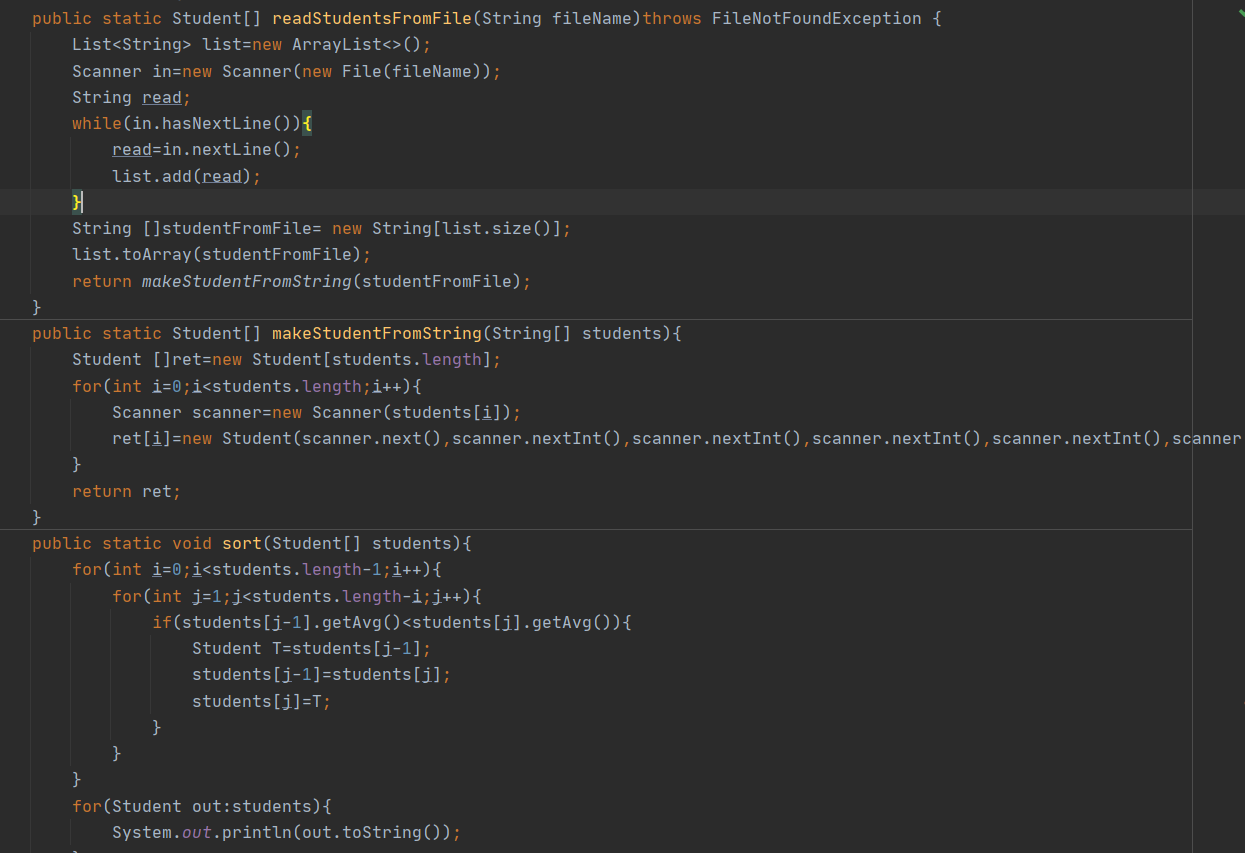


7. 学生成绩管理

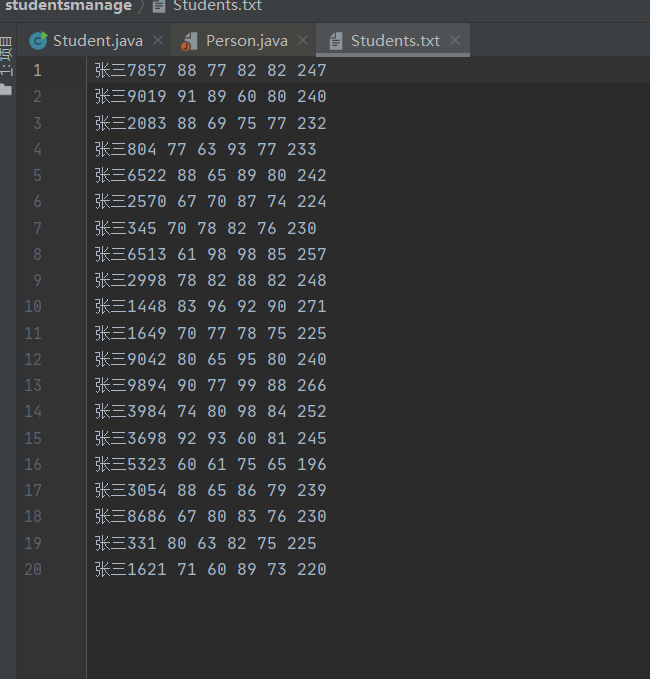
代码见附件studentsmanage

设计思想：定义一个Student类，用于存放每个学生数据，定义两个外置方法读入并创建Student列表，定义一个sort方法根据Student对象的具体数据具体排序（采用普普通通朴实无华冒泡排序）。

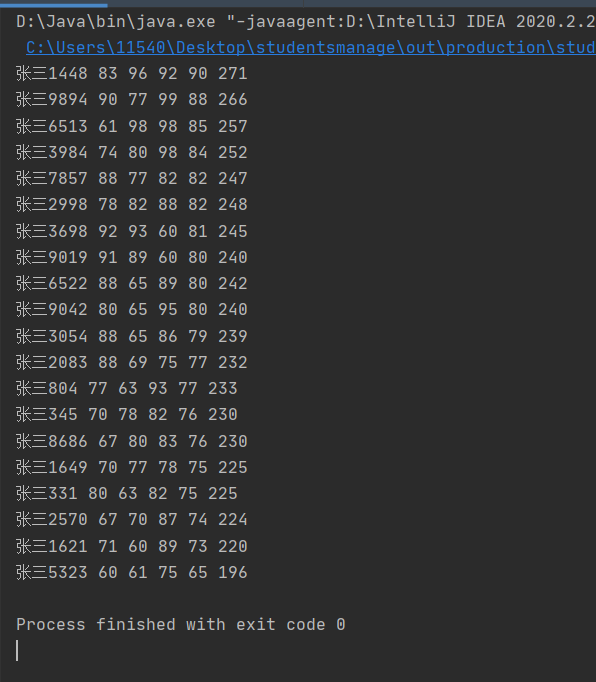
关键代码：



测试数据：



测试结果：



**五、实验小结**

1．实验中遇到的问题及解决过程

①静态方法和非静态方法傻傻分不清楚，在该用非静态的时候用了静态方法，导致静态方法无法使用非静态对象的变量报错。

解决过程：发现问题，把静态方法改为非静态方法，静态方法速度比较快但是太占内存，非静态方法的好处就是节省内存，动态规划。要分清楚。

②考虑如何让第三题的循环更加简略。

解决过程：在本来有的类上又套了一层壳，使不同图形的周长面积计算有一个统一的接口，后来学的继承才知道应该用继承的思想来写这个题目。不过套个壳勉强能实现，下次再改一下。

2．实验中产生的错误及原因分析

实验中产生的错误还是对概念不熟悉的问题，比如静态和非静态混用，private作用域搞不清楚诸如此类的一些问题。尽量去了解不同东西在特定时候的作用吧，分不清楚混用往往会造成更大的麻烦。

3．实验体会和收获。

大概是能体会到一些面向对象的意思了吧，将所有的数据方法封装在一定程度上确实可以让程序思路更加清晰，降低了写多代码时候容易错乱麻烦的出现。还有就是封装为具体的类能更好的解决实际问题，当一个问题能抽象出数据与方法时，用面向对象的思路就可以很方便地解决这个问题。但是具体要用什么想法来抽象还需要再做考虑，感觉有的时候抽象出来的东西反而没什么用倒是像为了抽象而抽象的东西，还需要累积一点经验才能来写好一个类。

**六、附录**

[1][美] 凯.S.霍斯特曼（Cay S. Horstmann）.Java核心技术·卷 I（原书第10版）[M].机械工业出版社