****

****

**信息学院软件工程系**

**《JAVA程序设计》实验报告**

实验七

**姓名：陈澄**

**学号：32420212202930**

**学院：信息学院**

**专业：软件工程专业**

**完成时间：2023.04.12**

**一、实验目的及要求**

* 熟悉异常处理
* 熟悉泛型方法和泛型类

**二、实验题目及实现过程**

题目2：（构造方法失败）编写一个程序，给出一个构造方法，它将关于构造方法失败的信息传递给一个异常处理器。定义一个 SomeClass 类，它在构造方法中抛出异常。程序应创建一个 SomeClass 型的对象，并捕获由这个构造方法抛出的异常。

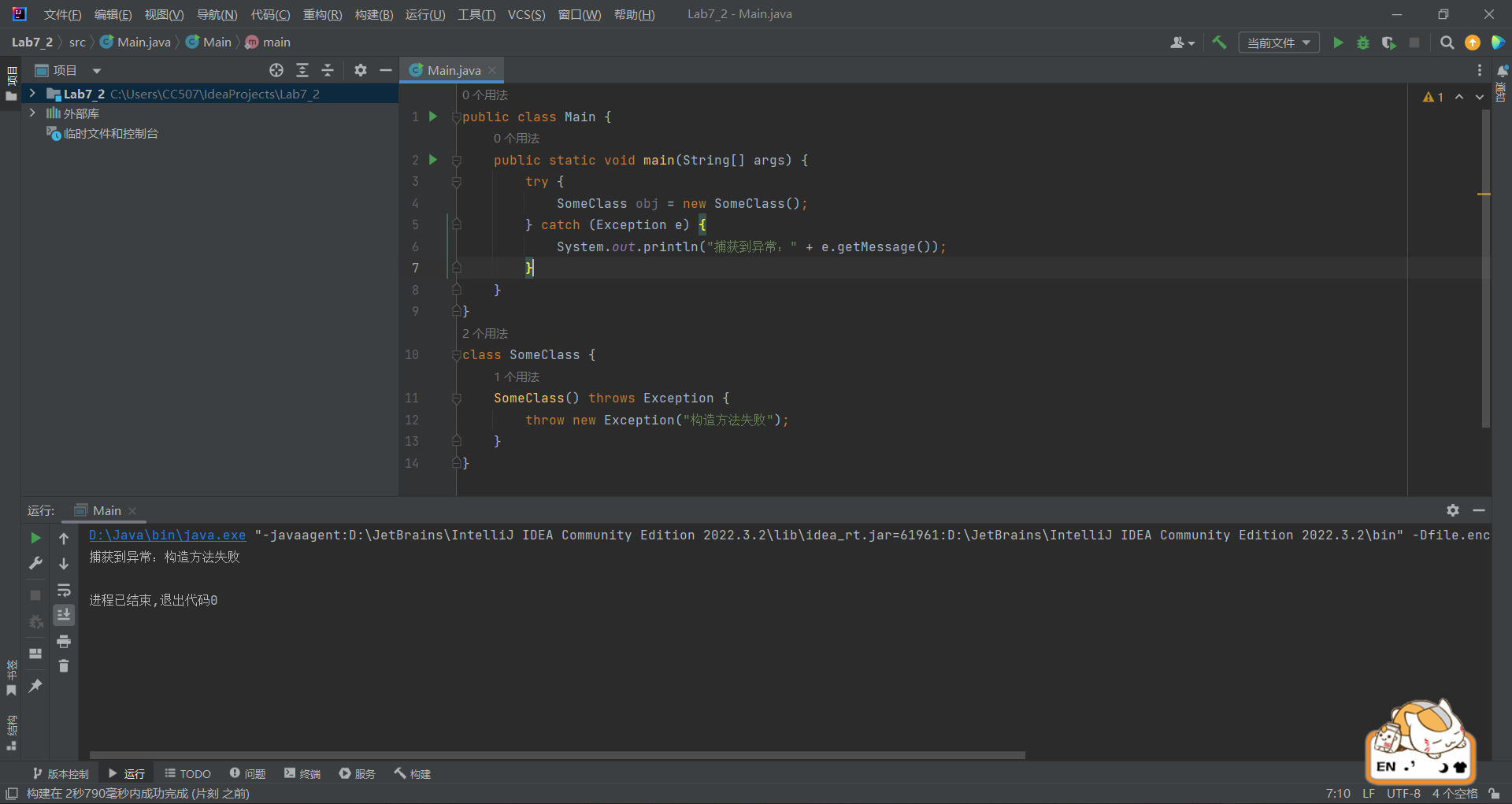
1. 实验环境（集成开发环境、jdk版本、字符编码等）

集成开发环境：Intellij IDEA

jdk版本：17.0.5

字符编码：ASCII

1. 实现过程（本部分为主要评分依据，请描述解题思路，比如总共设计几个类，各个类的用途、成员、主要方法等及其之间调用关系等）
2. 新建一个类SomeClass，在构建该类的方法中抛出一个Exception，报错内容是“构造方法失败”。
3. 在主函数中利用try...catch语句：try中创建一个SomeClass对象catch中捕获异常并输出该异常的信息。
4. 过程截图（本部分为主要评分依据，一张全屏截图（必须）、若干运行结果展示图（可选），主要代码（可选））



题目3：（重抛异常）编写一个演示重抛异常的程序。定义两个方法 someMethod 和 someMethod2, someMethod2 方法的功能就是抛出一个异常。someMethod 方法调用 someMethod2，捕获一个异常并重抛它。用 main 方法调用 someMethod 方法，并捕获被重抛的异常。输出这个异常的栈踪迹。

（一）实验环境（集成开发环境、jdk版本、字符编码等）

集成开发环境：Intellij IDEA

jdk版本：17.0.5

字符编码：ASCII

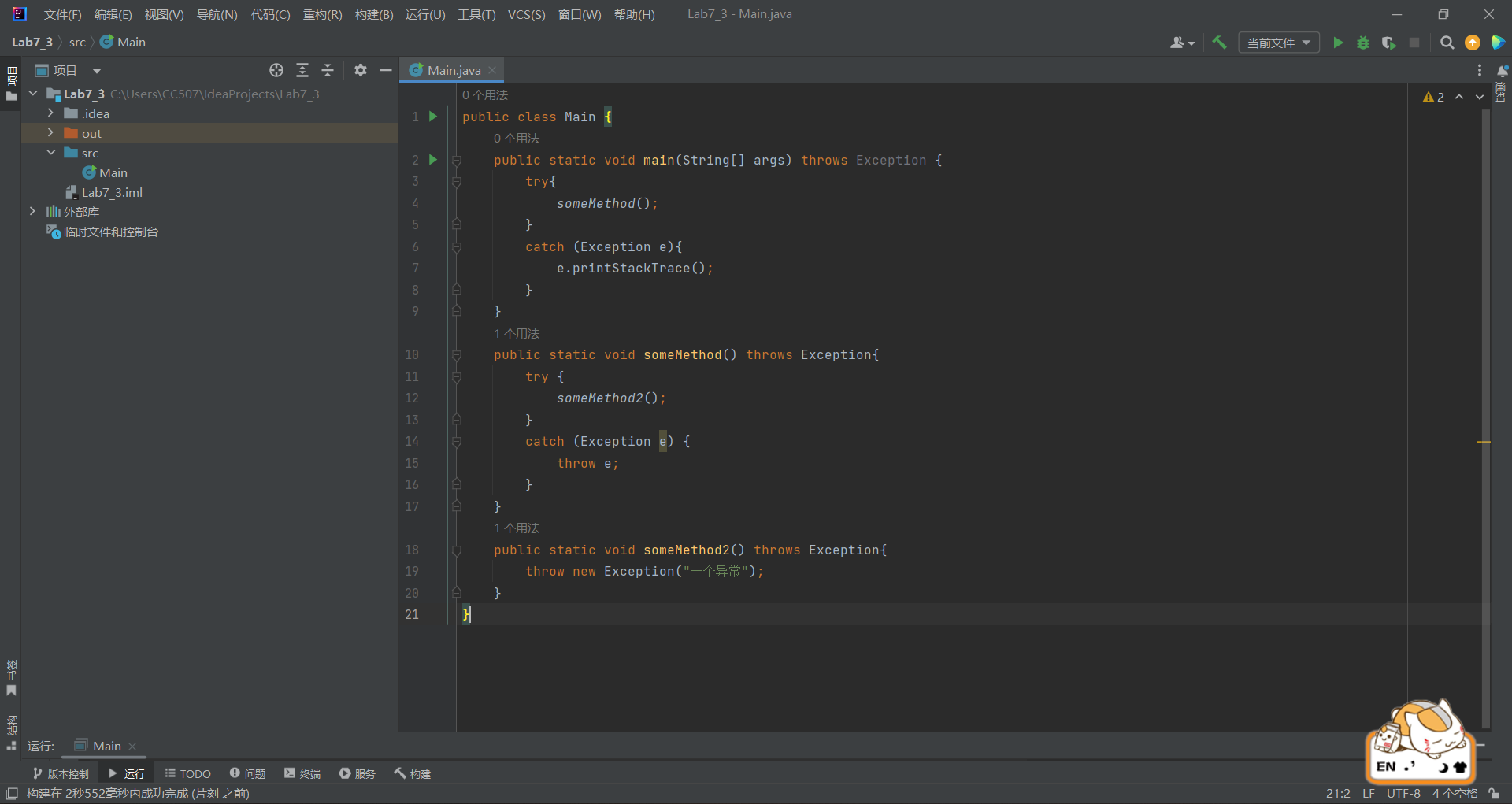
（二）实现过程（本部分为主要评分依据，请描述解题思路，比如总共设计几个类，各个类的用途、成员、主要方法等及其之间调用关系等）

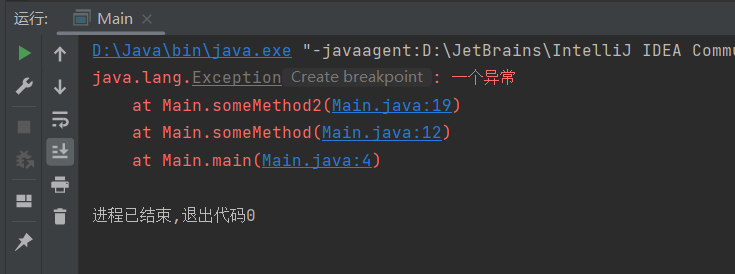
1.新建一个方法someMethod2在其中抛出一个异常，其报错信息为“一个异常”。

2.新建一个方法someMethod在其中利用try...catch语句，调用someMethod2如果出现异常则在catch中捕获并重新抛出这个异常。

3.在主函数中利用try...catch语句，调用someMethod方法，若出现异常则在catch中捕获并调用printStackTrace方法输出该异常的栈踪迹。

（三）过程截图（本部分为主要评分依据，一张全屏截图（必须）、若干运行结果展示图（可选），主要代码（可选））





题目4：自定义异常的定义、拋出和捕获：

(1) 自定义两个异常类：非法姓名异常IllegaNameException和非法地址异常IllegalAddressException。

(2) 定义Student类包含name和address属性，和setName、setAddress方法，当姓名长度小于1或者大于5时抛出IllegaNameException，当地址中不含有“省”或者“市”关键字时抛出IllegalAddressException。

(3) 编写程序抛出这两种异常，在main方法中进行捕获并合理地处理。

（一）实验环境（集成开发环境、jdk版本、字符编码等）

集成开发环境：Intellij IDEA

jdk版本：17.0.5

字符编码：ASCII

（二）实现过程（本部分为主要评分依据，请描述解题思路，比如总共设计几个类，各个类的用途、成员、主要方法等及其之间调用关系等）

1.新建IllegaNameException类继承自Exception并在其中重写getMessage函数，将报错信息改为“姓名长度必须在1~5之间”。

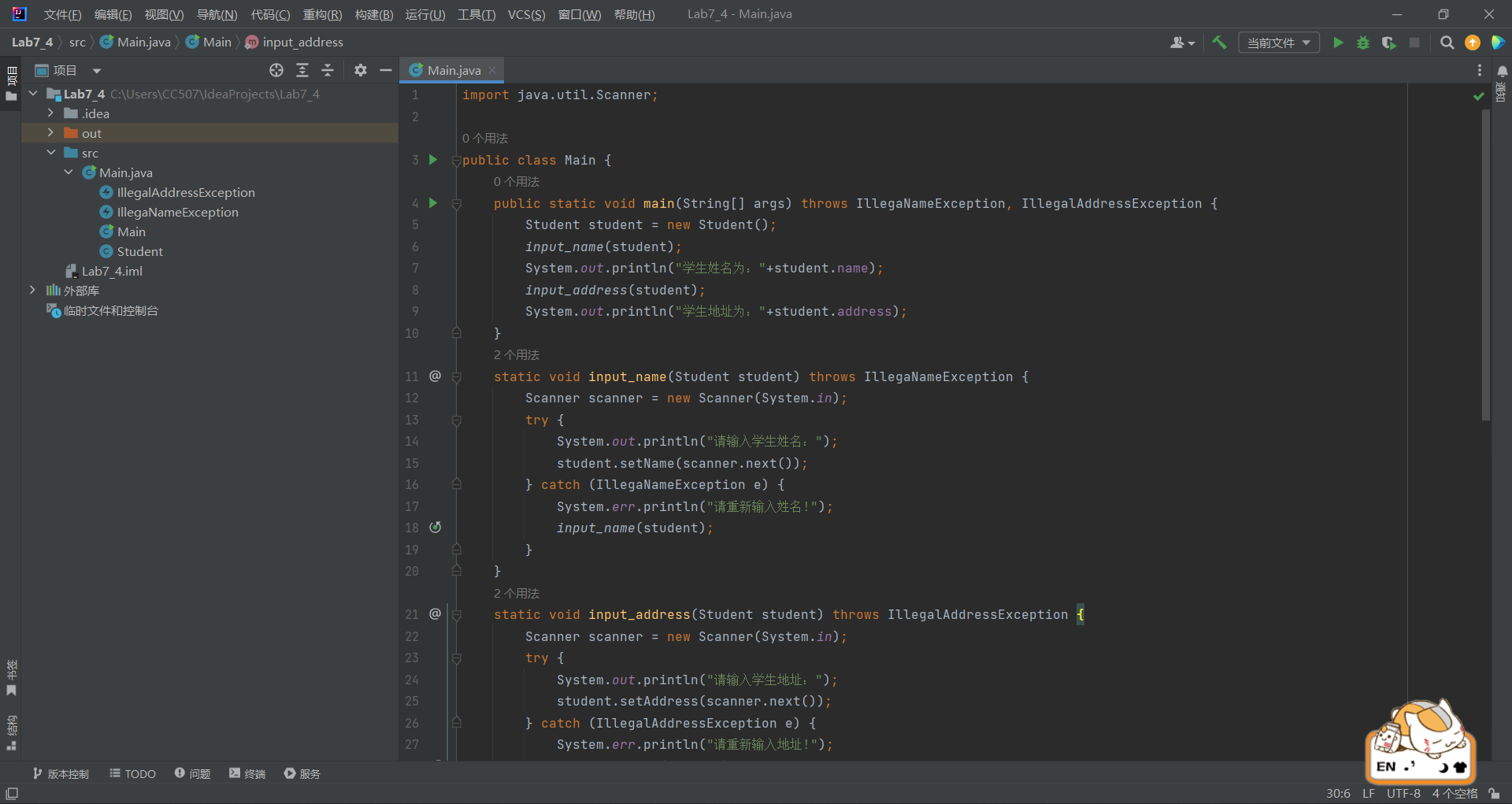
2.新建IllegalAddressException类继承自Exception并在其中重写getMessage函数，将报错信息改为“地址必须包含省或市”。

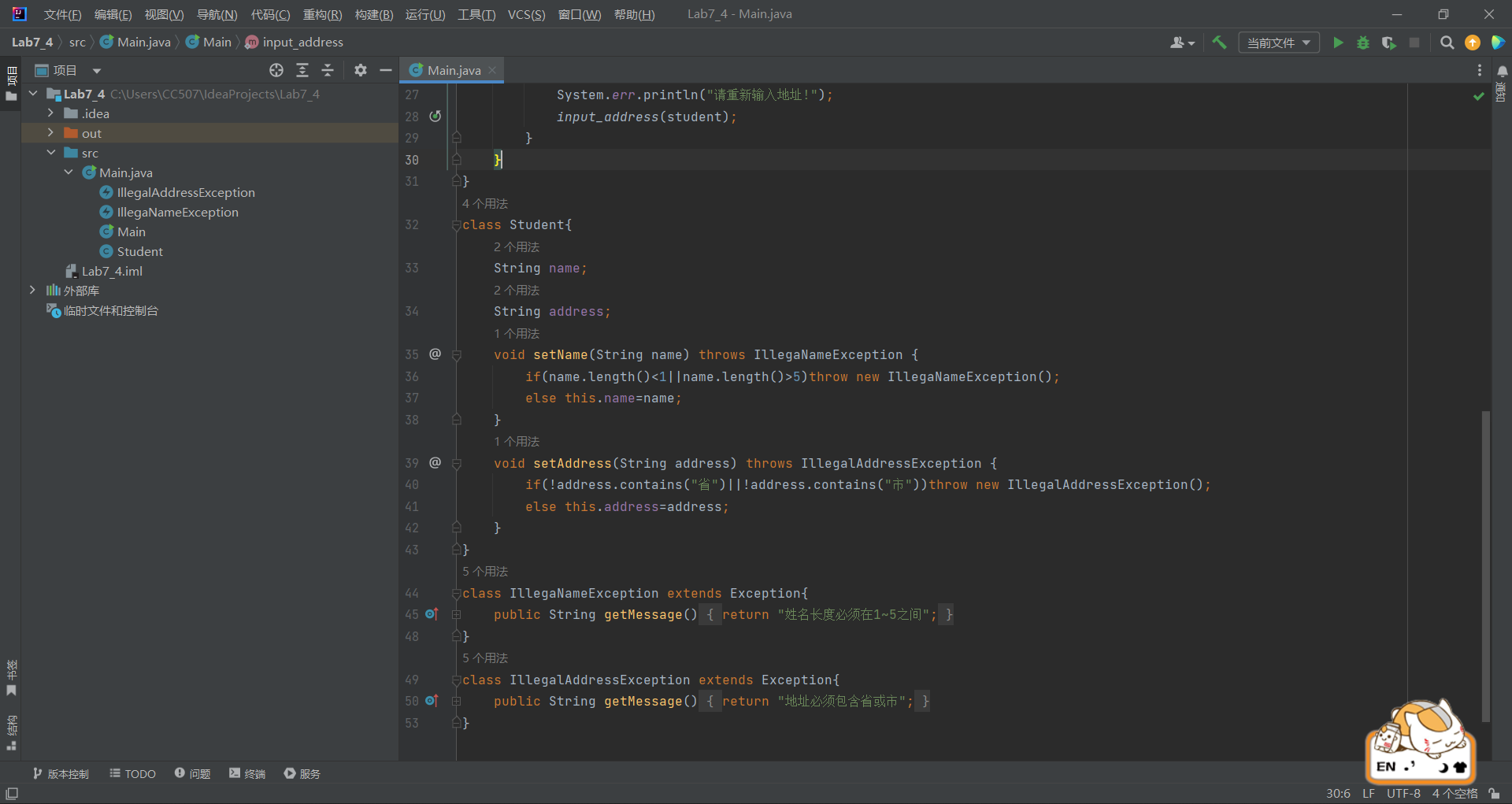
3.新建Student类，包含两个String型变量name和address，在其中建立setName方法，输入参数name，当name的长度小于1或者大于5的时候抛IllegaNameException异常。在其中建立setAddress方法，输入参数address，用contains方法检测address中是否包含“省”或者“市”关键词，若都没有则抛出IllegalAddressException异常。

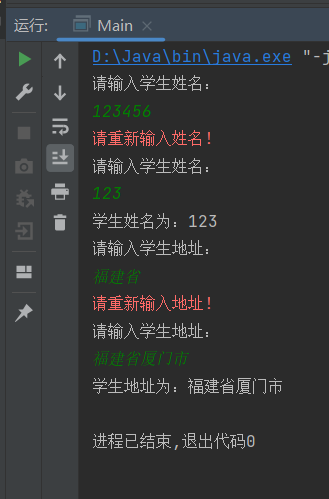
4.在主函数所在的类中创建两个方法input\_name和input\_address，其中用try...catch语句，try中输入学生的姓名（地址），若不合法则在catch中捕获相应的错误，并让用户重新输入并再次调用该方法。

5.主函数中创建一个Student变量，一次调用上述两个方法，其后输出获得的地址和姓名。

（三）过程截图（本部分为主要评分依据，一张全屏截图（必须）、若干运行结果展示图（可选），主要代码（可选））







题目5：（泛型方法 isEqualTo）编写 isEqualTo 方法的一个简单泛型版本。它用 equals 方法比较两个实参相等时返回 true，否则返回 false。利用这个泛型方法，在程序中调用 isEqualTo 处理各种内置的类型，例如 Object 或 Integer。运行程序时，传递给 isEqualTo 方法的对象会根据它们的内容或者所引用的对象进行比较吗？

（一）实验环境（集成开发环境、jdk版本、字符编码等）

集成开发环境：Intellij IDEA

jdk版本：17.0.5

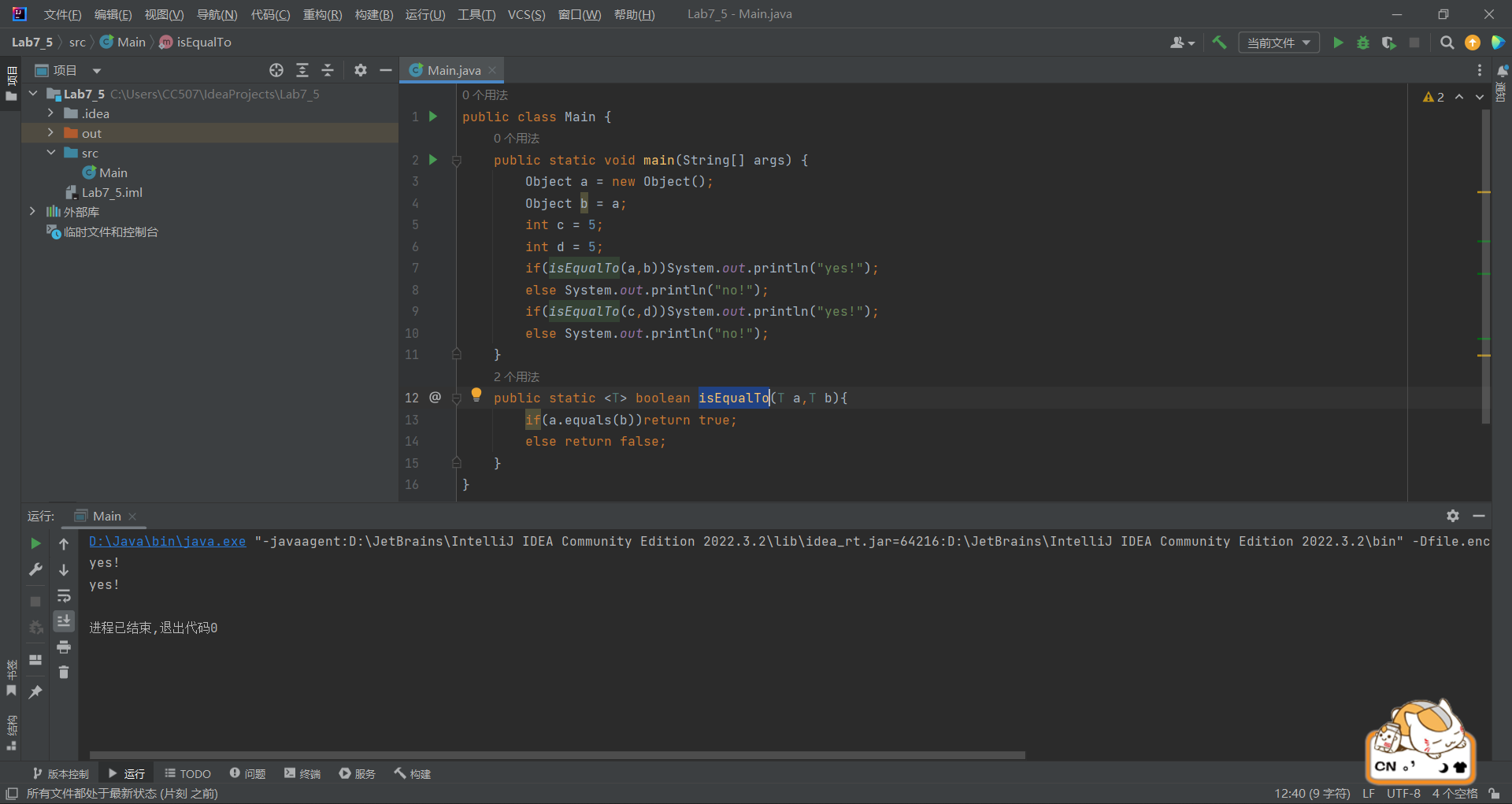
字符编码：ASCII

（二）实现过程（本部分为主要评分依据，请描述解题思路，比如总共设计几个类，各个类的用途、成员、主要方法等及其之间调用关系等）

1.新建一个方法isEqualTo，选用泛型T，输入参数为T类型的a,b，在其中调用equals方法，比较参数a,b是否相同，若相同则返回true，否则返回false。

2.主函数中新建一个Object型变量a,b，并令b=a，调用isEqualTo方法比较，若相同输出yes，否则输出no。新建两个int型变量c,d，并令他们都为5，一样调用isEqualTo方法比较。结果都为yes。

（三）过程截图（本部分为主要评分依据，一张全屏截图（必须）、若干运行结果展示图（可选），主要代码（可选））



题目6：（泛型类 Pair）编写一个泛型类 Pair，它有两个类型参数 F 和 S，分别代表一对值中第一个元素第二个元素的类型。为第一个元素和第二个元素添加 get 方法和 set 方法。（提示：类首部应当是public class Pair <F, S>.)

（一）实验环境（集成开发环境、jdk版本、字符编码等）

集成开发环境：Intellij IDEA

jdk版本：17.0.5

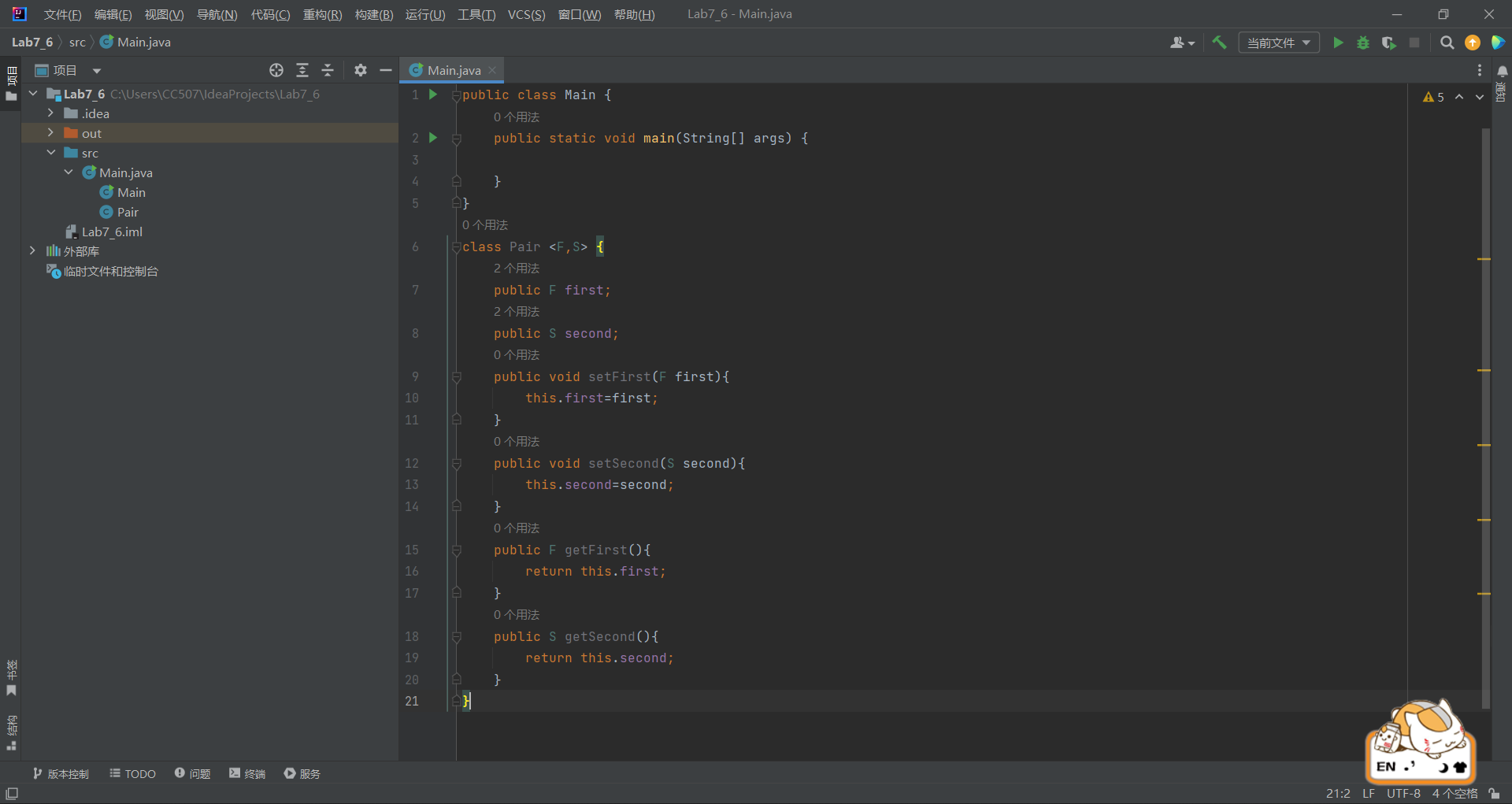
字符编码：ASCII

（二）实现过程（本部分为主要评分依据，请描述解题思路，比如总共设计几个类，各个类的用途、成员、主要方法等及其之间调用关系等）

1.创建一个含两个泛型F,S的类Pair，包含一个F型变量first，一个S型变量second

2.建立函数setFirst，setSecond分别用于输入两个变量，建立函数getFirst，getSecond用于返回两个变量。

（三）过程截图（本部分为主要评分依据，一张全屏截图（必须）、若干运行结果展示图（可选），主要代码（可选））



题目7：（混用组合和継承）将CommissionEmploye—BaseCommissionEmployee 继承层次重新建模成一个 Employee 层次，使得每一种员工都具有不同 CompensationModel 对象。本练习中，要求重新实现 CompensationModel 类成为一个接口，提供的公共抽象方法 earnings 不包含参数，且返回一个 double 值。然后，实现了 CompensationModel 接口的如下几个类：

a) SalariedCompensationModel 类——对于周薪固定的员工，这个类需包含一个 weeklySalary 实例变量，且需要实现 earnings 方法，返回 weeklySalary 值。

b) HourlyCompensationModel 类——对于按时薪计酬（包括每周工作超过 40 小时的加班工资）的员工，这个类需包含 wage 和 hours 实例变量，且需根据工作的小时数实现 earnings 方法。

c) CommissionCompensationModel 类——对于按佣金付酬的员工，这个类需包含 grossSales 和 commissionRate 实例变量，还需要实现 earnings 方法，它返回 grossSales x commissionRate 的结果。

d) BasePlusCommissionCompensationMode 类——对按加金付酬的员工，这个类需包含 grossSales commissionRate 和 baseSalary 实例变量，还需要实现 earnings 方法，它返回 baseSalary + grossSales x commissionRate。

在测试程序中，需为上面描述的每一种 CompensationModel 创建一个 Employee 对象，然后显示每一类员工的收入。接着，动态地改变员工的 CompensationModel，重新显示他的收入。

（一）实验环境（集成开发环境、jdk版本、字符编码等）

集成开发环境：Intellij IDEA

jdk版本：17.0.5

字符编码：ASCII

（二）实现过程（本部分为主要评分依据，请描述解题思路，比如总共设计几个类，各个类的用途、成员、主要方法等及其之间调用关系等）

1.新建一个接口CompensationModel，内含一个方法earnings直接返回double型的0

2.建立类SalariedCompensationModel实现CompensationModel接口，包含一个double型变量weeklySalary，在其中实现构建方法，并在其中重写earnings方法，直接返回weeklySalary。

3.建立类HourlyCompensationModel实现CompensationModel接口，包含两个double型变量wage和hours，在其中实现构建方法，并在其中重写earnings方法，若hours>40，返回wage\*40+wage\*3\*(hours-40)，否则返回wage\*hours。

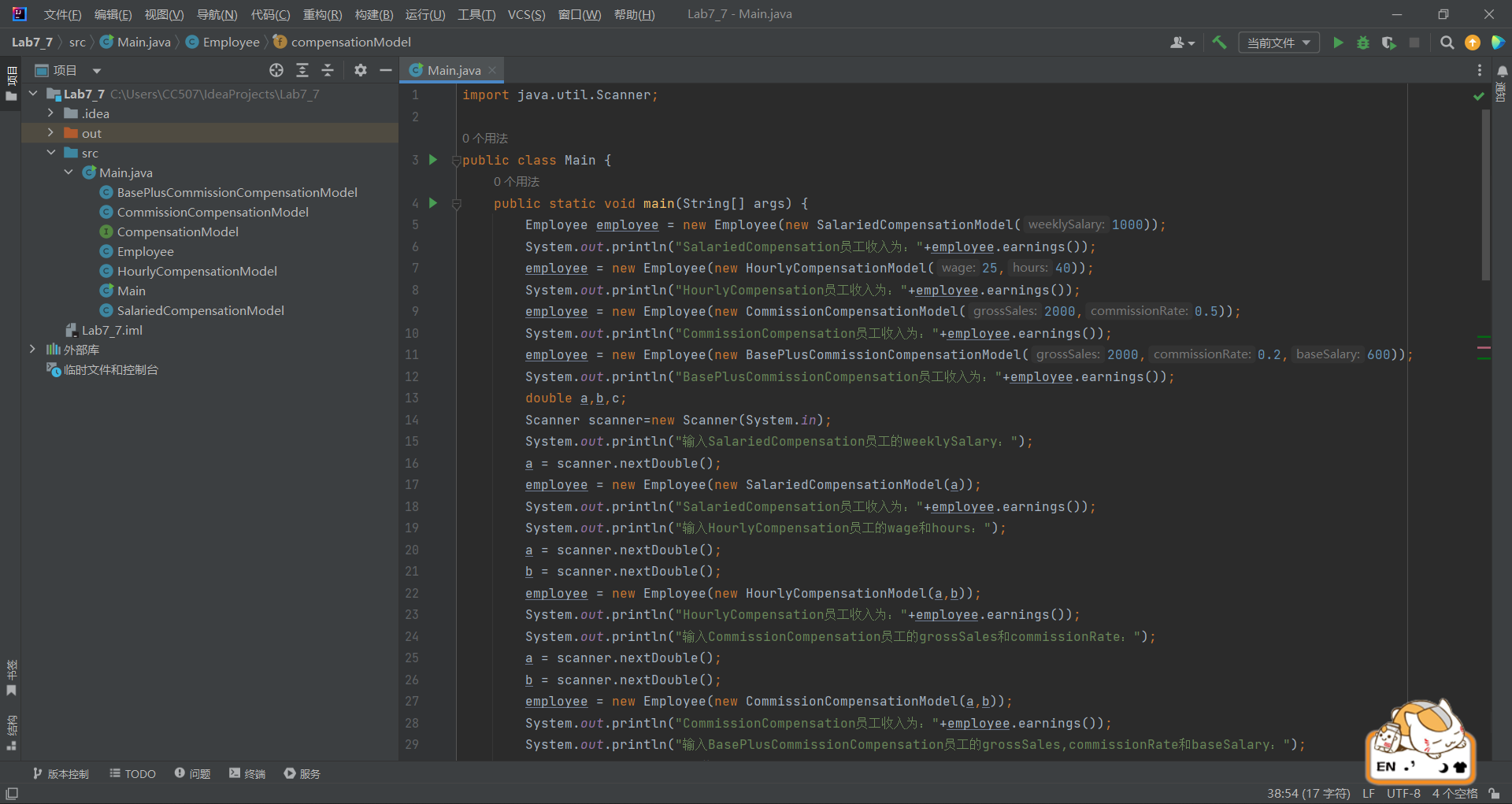
4.建立类CommissionCompensationModel实现CompensationModel接口，包含两个double型变量grossSales和commissionRate，在其中实现构建方法，并在其中重写earnings方法，直接返回grossSales\*commissionRate。

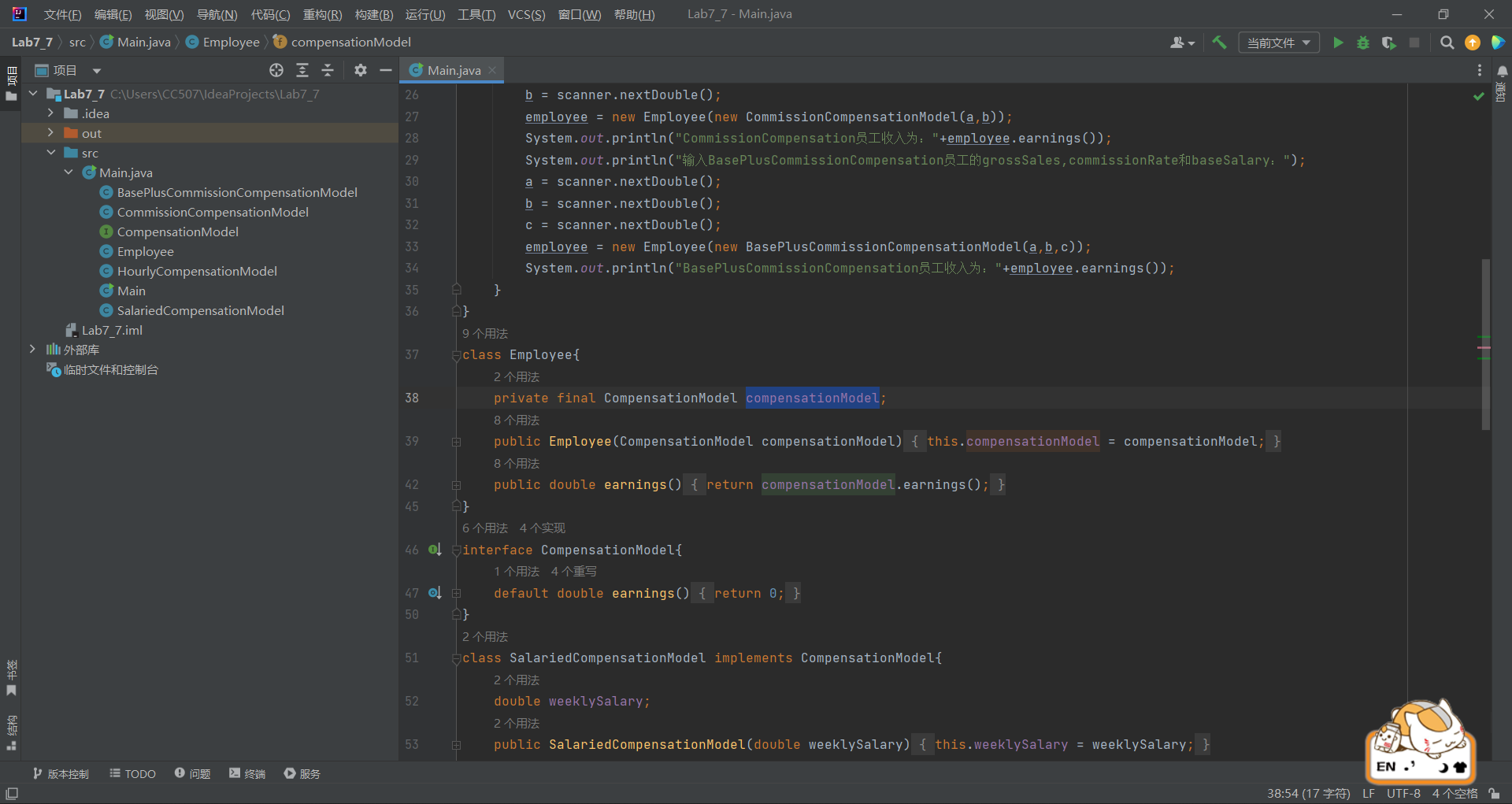
5.建立类BasePlusCommissionCompensationModel实现CompensationModel接口，包含三个double型变量grossSales，commissionRate和baseSalary，在其中实现构建方法，并在其中重写earnings方法，直接返回baseSalary + grossSales\*commissionRate。

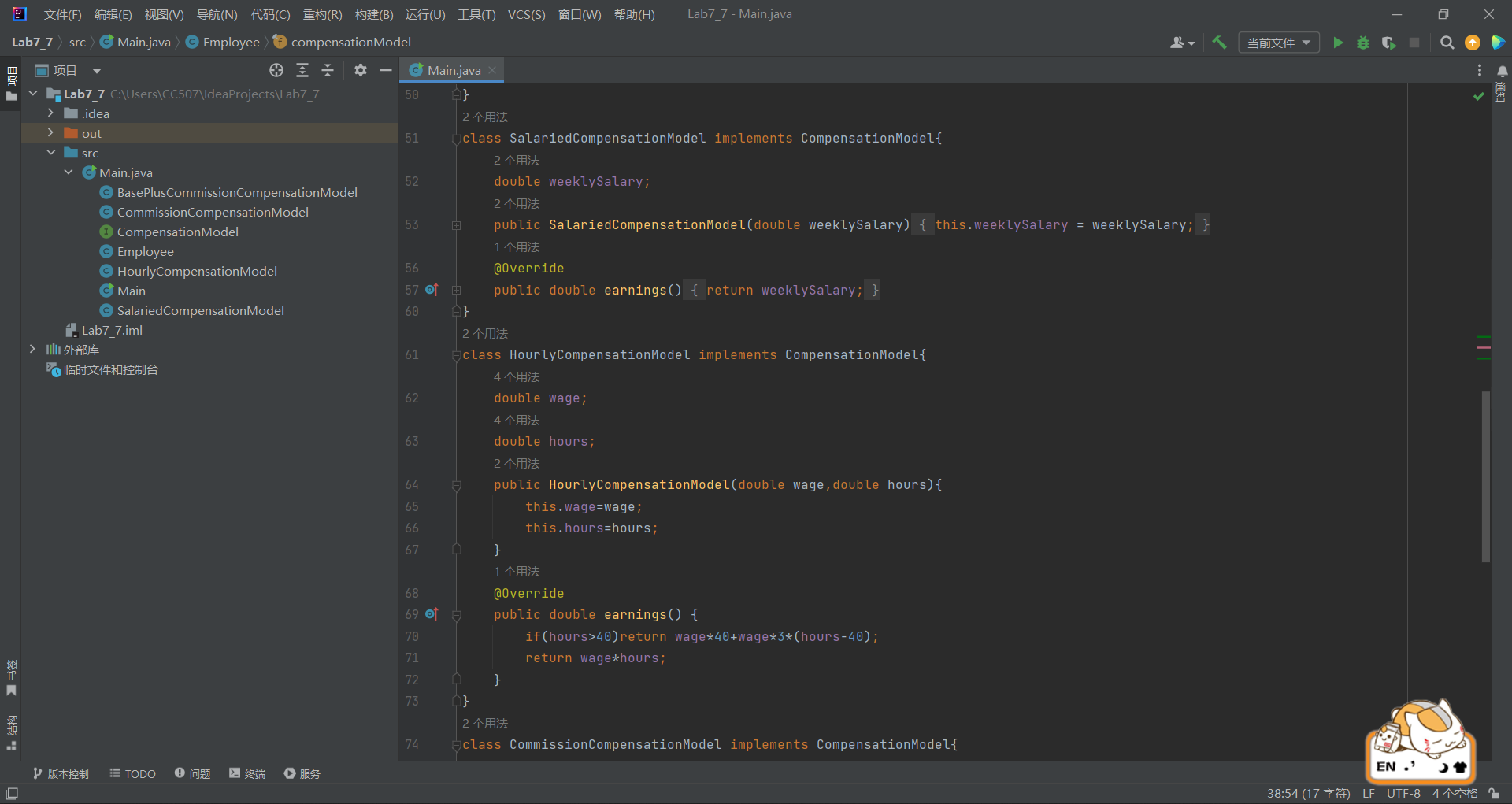
6.建立类Employee，内含一个CompensationModel变量compensationModel，使得员工的类型唯一。包含一个构建方法，还有一个earnings方法调用compensationModel下的earnings方法返回一个double值。

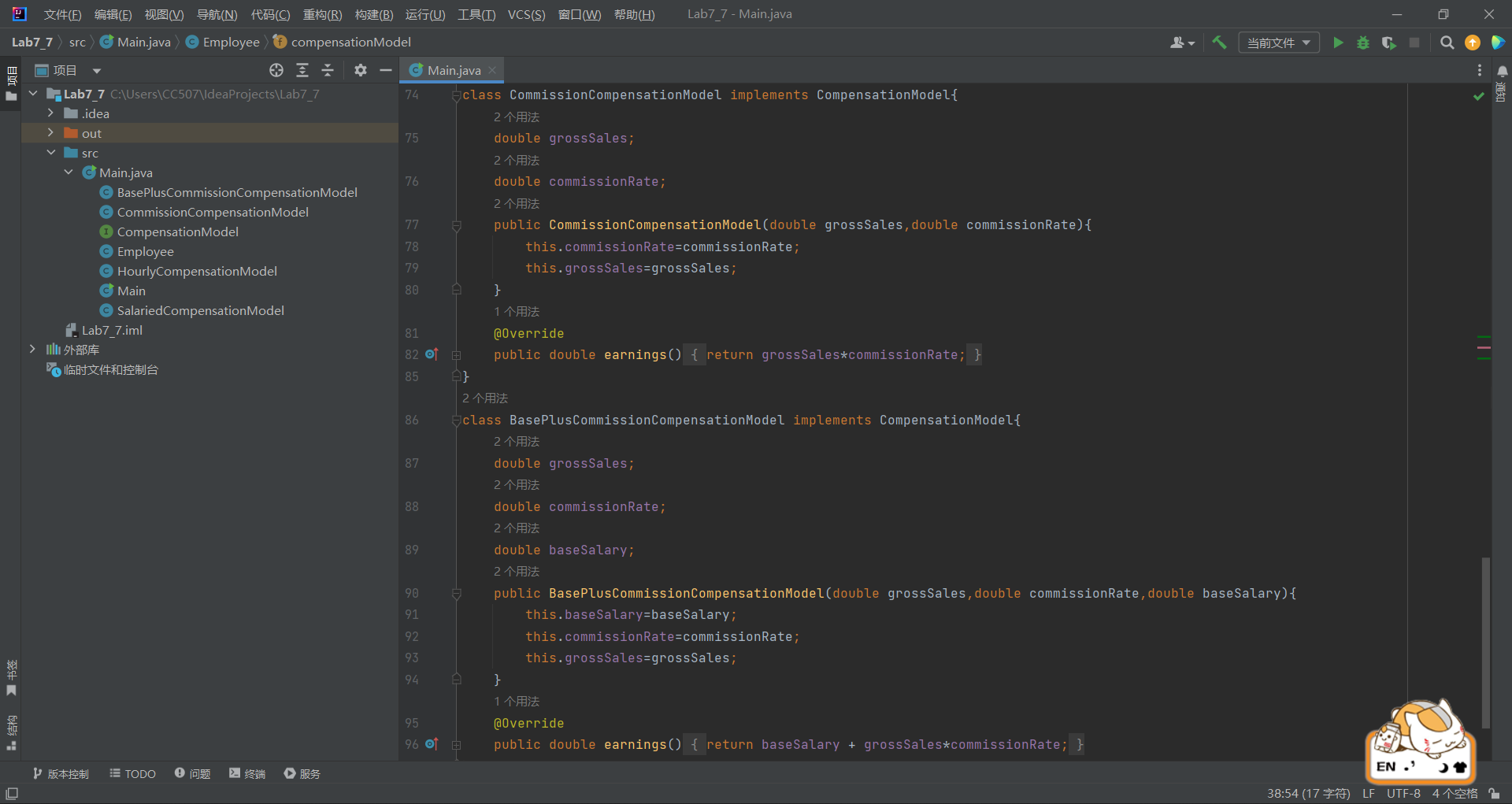
7.在主函数中创建一个Employee型employee，分别将其初始化为四个不同类型员工并给定对应变量的值，调用earnings方法输出其薪资。其后创建三个double型变量a,b,c用于获得输入，根据输入初始化四种不同类型员工，输出其薪资。

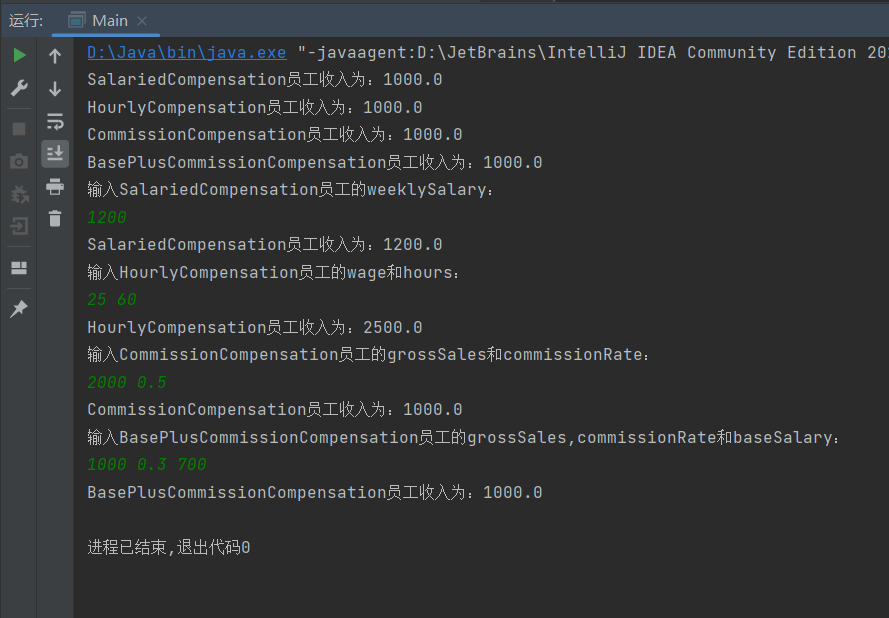
（三）过程截图（本部分为主要评分依据，一张全屏截图（必须）、若干运行结果展示图（可选），主要代码（可选））











**三、实验总结与心得记录**

本部分根据实验过程的所得所想描述，记录可供以后复习回看 {可以记录调试过程遇到的问题，自己哪些知识点话掌握不够，设计是否有缺陷（比如耗时？耗内存？）是否有亮点，是否有精妙的算法，或者设计模式的应用，可吐槽，也可与其他语言作适当对比。}（本部分不作为平时评分依据）

备注：

建议附带代码提交的方式：导出工程压缩包。

平时实验成绩以考查参与度为主，所有实验要求自己完成，一旦发现抄袭或者其他投机取巧，取消所有平时成绩