****

****

**信息学院软件工程系**

**《JAVA程序设计》实验报告**

实验七

**姓名：庾晓萍**

**学号：20420192201952**

**学院：信息学院**

**专业：软件工程**

**完成时间：2022/4/8**

1. **实验目的及要求**

（一）实验目的

1、多态

2、熟悉异常处理

1. 实验要求

1、按照题目要求写代码和实验报告，并上传到FTP

1. **实验题目及实现过程**

一、基本题目：

**题目1：Payroll System Modification**

1. **实验环境**

操作系统：Windows 10；

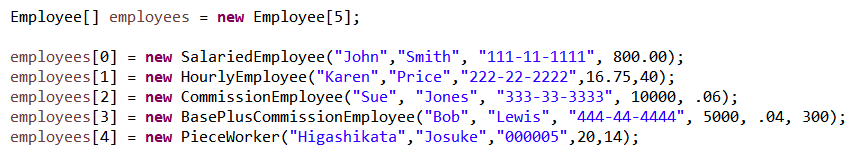
IDE：Eclipse Java 2018-12

编程语言：Java；

1. **实现过程**
2. 设计思路

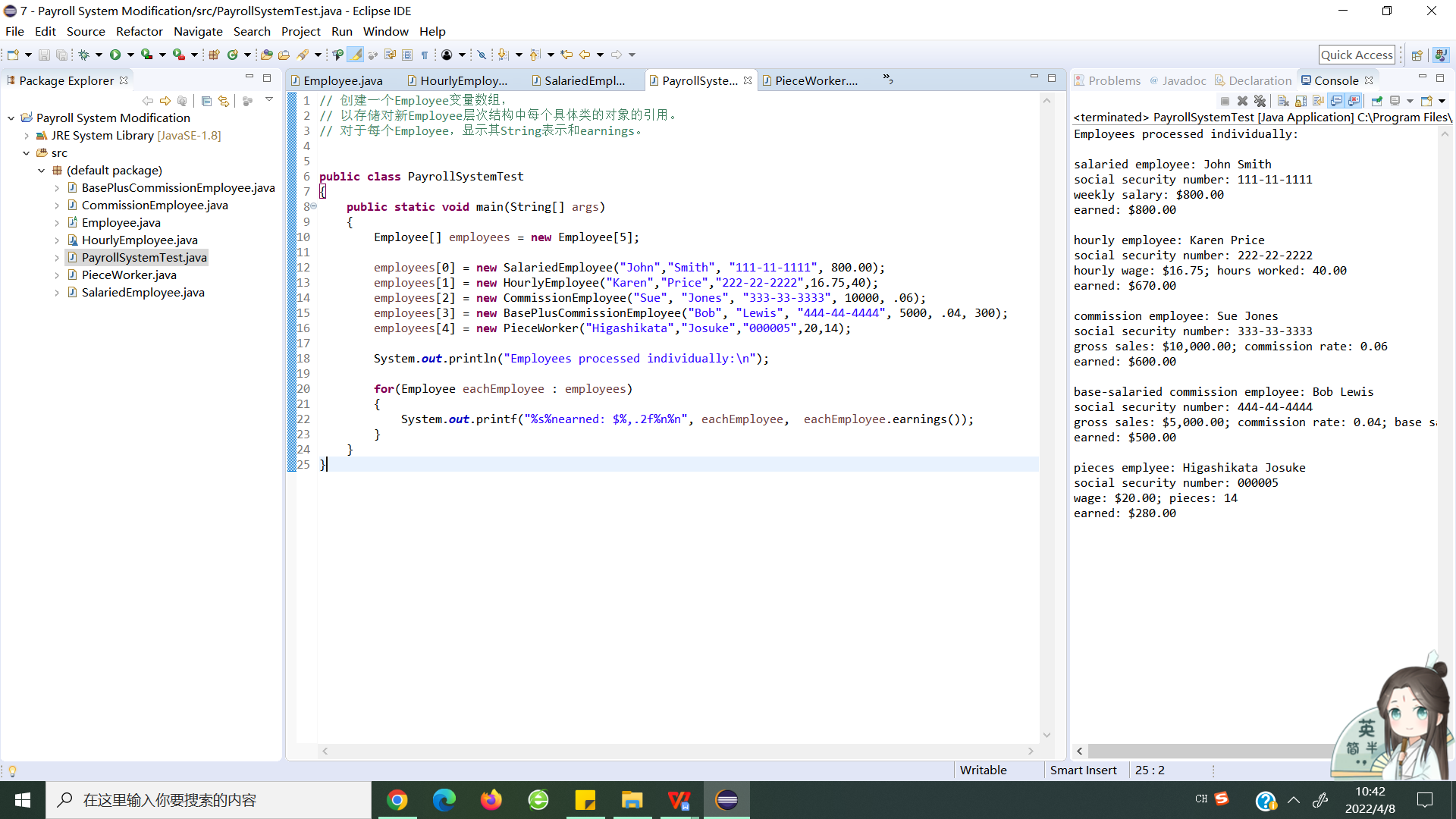
①修改工资系统，增加一个Employee子类PieceWorker，这个员工的工资是根据生产商品的数量决定的。PieceWorker类包含private实例变量wage（工资）和pieces（存储生产的件数量）。提供earnings方法的具体实现来计算工资（生产的数量乘以每件工资）。

②在PayrollSystemTest类中创建一个Employee变量数组，对Employee层次结构中每个具体类的对象引用。对于每个Employee，通过动态绑定，显示不同的String表示和earnings。



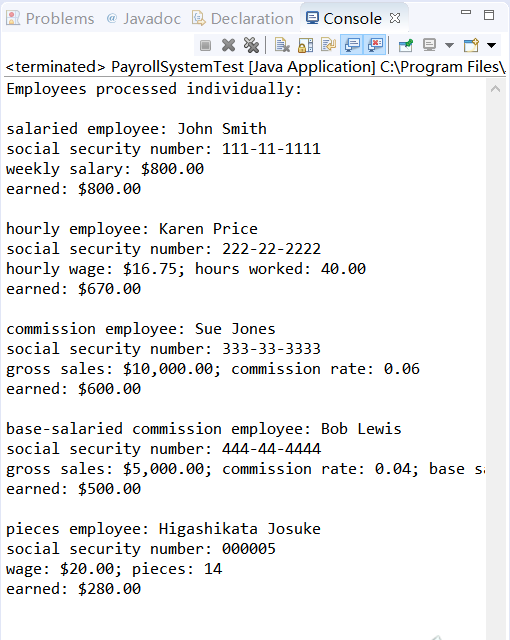
**（三） 过程截图**

（1）全屏截图



（2）运行结果

1. 创建了新类pieces employee，打印Employee变量数组，得到结果。



**题目2：Accounts Payable System Modification**

**（一） 实验环境**

操作系统：Windows 10；

IDE：Eclipse Java 2018-12

编程语言：Java；

**（二） 实现过程**

（1）设计思路

①修改accounts payable application，以包括工资单申请的完整功能。应用程序仍然应该处理两个Invoice对象，但现在应该处理四个Employee子类中每个对象中的一个对象。如果当前正在处理的对象是Base-PlusCommissionEmployee，则应用程序应该将Base-PlusCommissionEmployee的 basesalary增加10%。最后，应用程序应该输出每个对象的支付金额。

（2）设计类

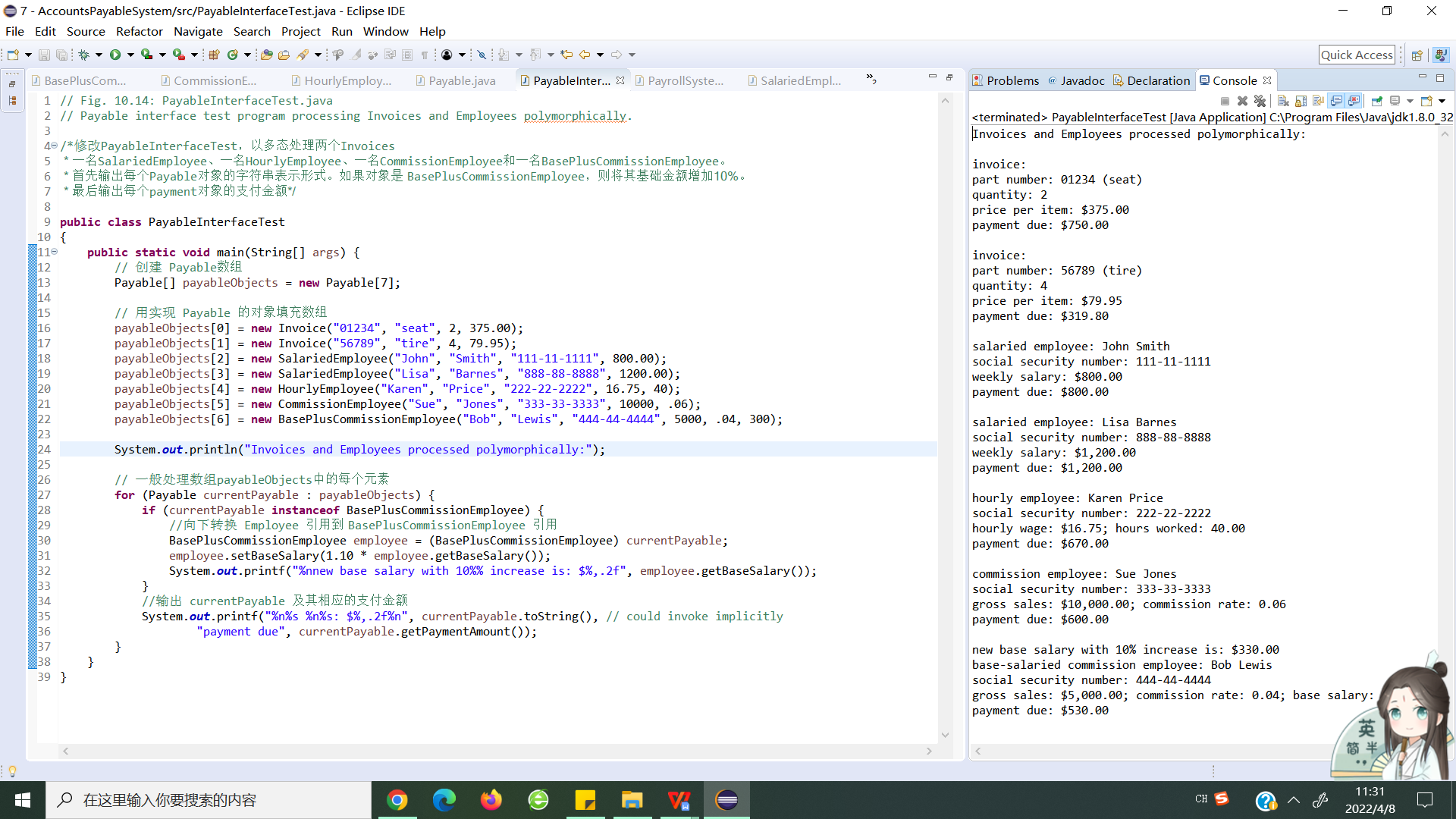
① 修改HourlyEmployee和CommissionEmployee，将它们作为实现 Payable 的 Employee版本的子类，放置在 Payable 层次结构中。将子类中的方法 earnings 的名称更改为 getPaymentAmount，以便该类满足其继承的与接口 Payable 的约定。

② 修改类BasePlusCommissionEmployee，以便扩展①部分中创建的CommissionEmployee类的版本。

③ 修改PayableInterfaceTes，以多态处理两个Invoices，一名SalariedEmployee、一名HourlyEmployee、一名CommissionEmployee和一名BasePlusCommissionEmployee。首先输出每个Payable对象的字符串表示形式。接下来，如果一个对象是 BasePlusCommissionEmployee，则将其基础金额增加10%。最后，输出每个payment对象的支付金额。

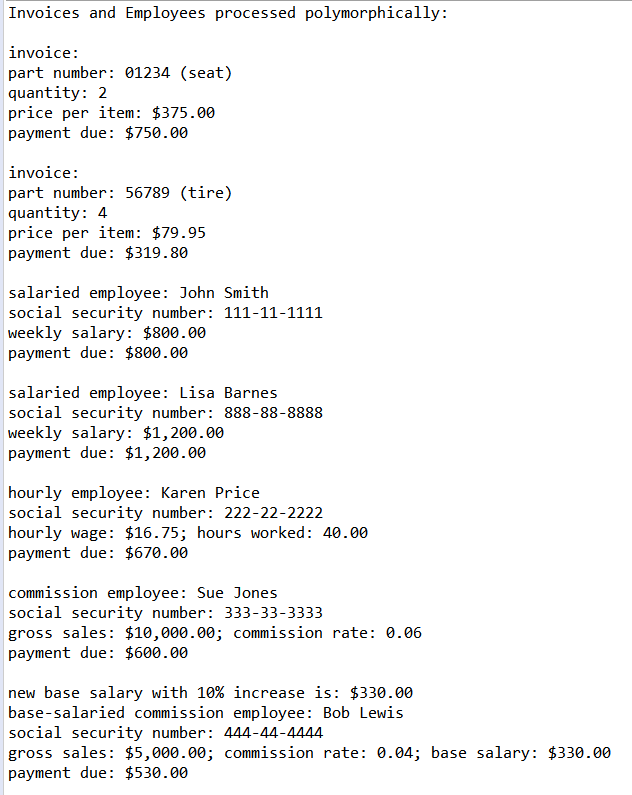
1. **过程截图**

（1）全屏截图



（2）运行结果

① 运行PayableInterfaceTest，Invoices和Employees多态处理。首先一般处理数组payableObjects中的每个元素，这个时候可以隐式调用。输出 currentPayable 及其相应的支付金额。当处理的元素类型是BasePlusCommissionEmployee 时，向下转换 Employee 引用到 BasePlusCommissionEmployee 引用，将其基础金额增加10%。



**题目3：CarbonFootprint Interface: Polymorphism**

**（一） 实验环境**

操作系统：Windows 10；

IDE：Eclipse Java 2018-12

编程语言：Java；

**（二） 实现过程**

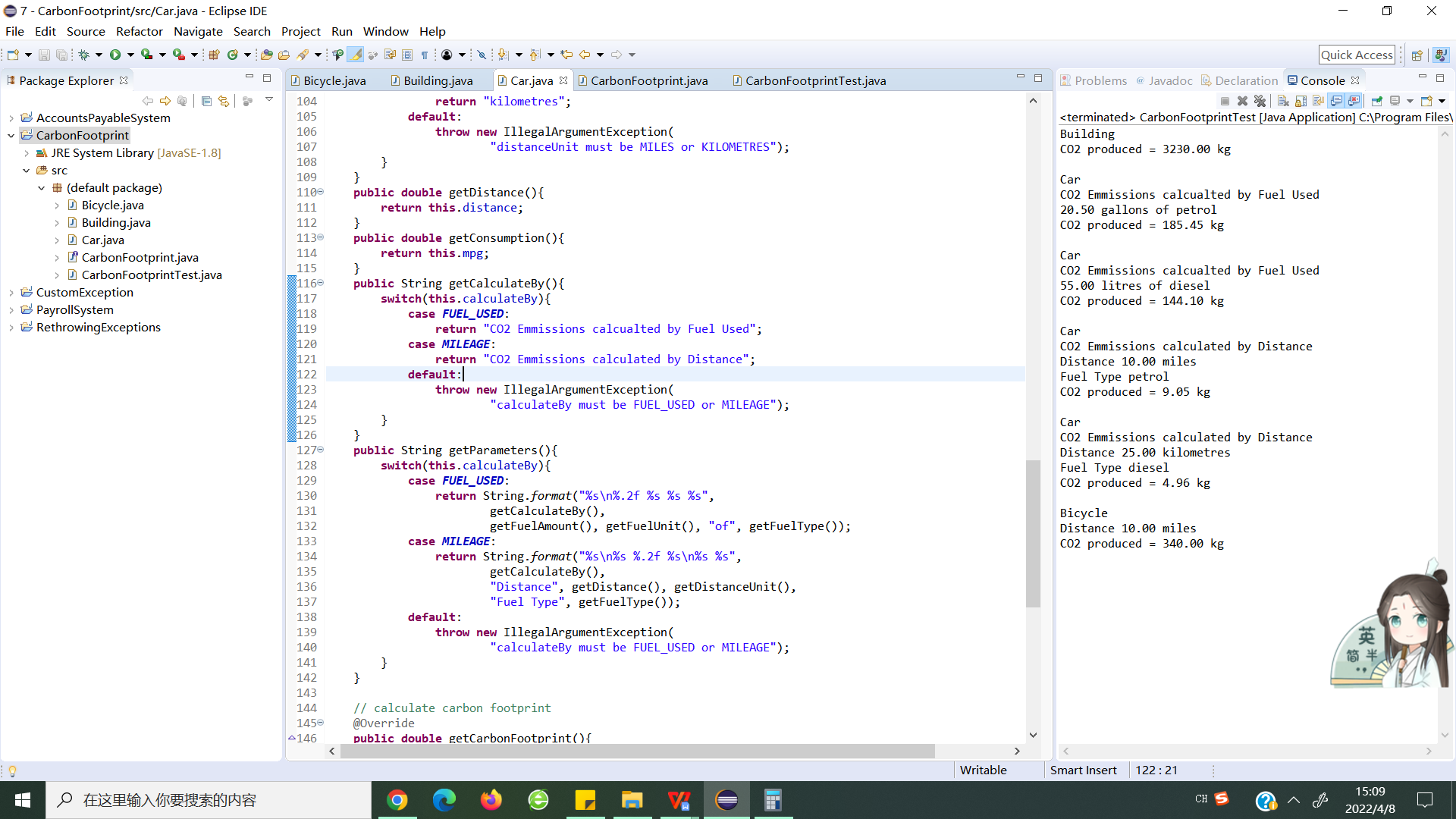
（1）设计思路

① 使用接口，可以为可能不同的类指定类似的行为。创建三个与继承类不相关的小类，Building, Car和Bicycle。给每个类一些与其他类没有共同点的独特的适当属性和行为。编写一个拥有getCarbonFootprint方法的接口CarbonFootprint。让每个类都实现该接口，getCarbonFootprint方法为该类计算适当的碳足迹。

② 编写应用程序，创建三个类的每个对象，将这些对象的应用放在ArrayList<CarbonFootprint>>中，然后遍历ArrayList<CarbonFootprint>，多态调用每个对象的getCarbonFootprint方法。对于每个对象，打印识别信息和对象的碳足迹。

**（三）过程截图**

（1）全屏截图

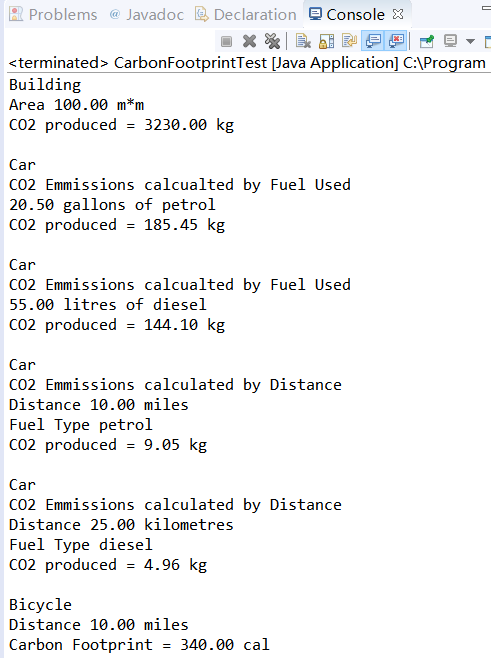


（2）运行结果

① 查找资料，对于Building类，中国现有城镇总建筑存量约650亿平方米，这些建筑在使用过程中排放了约21亿吨二氧化碳，平均每平方米排放约32.3kg的二氧化碳。通过构造函数参数对建筑的面积和高度进行初始化，并计算总排放。

② 对于Car类，每升汽油的二氧化碳排放是2.39kg，每升柴油的二氧化碳排放量是2.62kg，通过构造函数参数确定汽车使用燃油的单位（是加仑还是升），燃油的类别（是柴油还是汽油），燃油量。距离的单位，距离，燃油类别，每加仑可以行驶多少英里。

③ 对于自行车，通过构造函数参数初始化骑行距离，利用卡路里计算公式算出碳排放。



**题目4：自定义异常的定义、拋出和捕获**

**（一） 实验环境**

操作系统：Windows 10；

IDE：Eclipse Java 2018-12

编程语言：Java；

**（二） 实现过程**

（1）设计思路

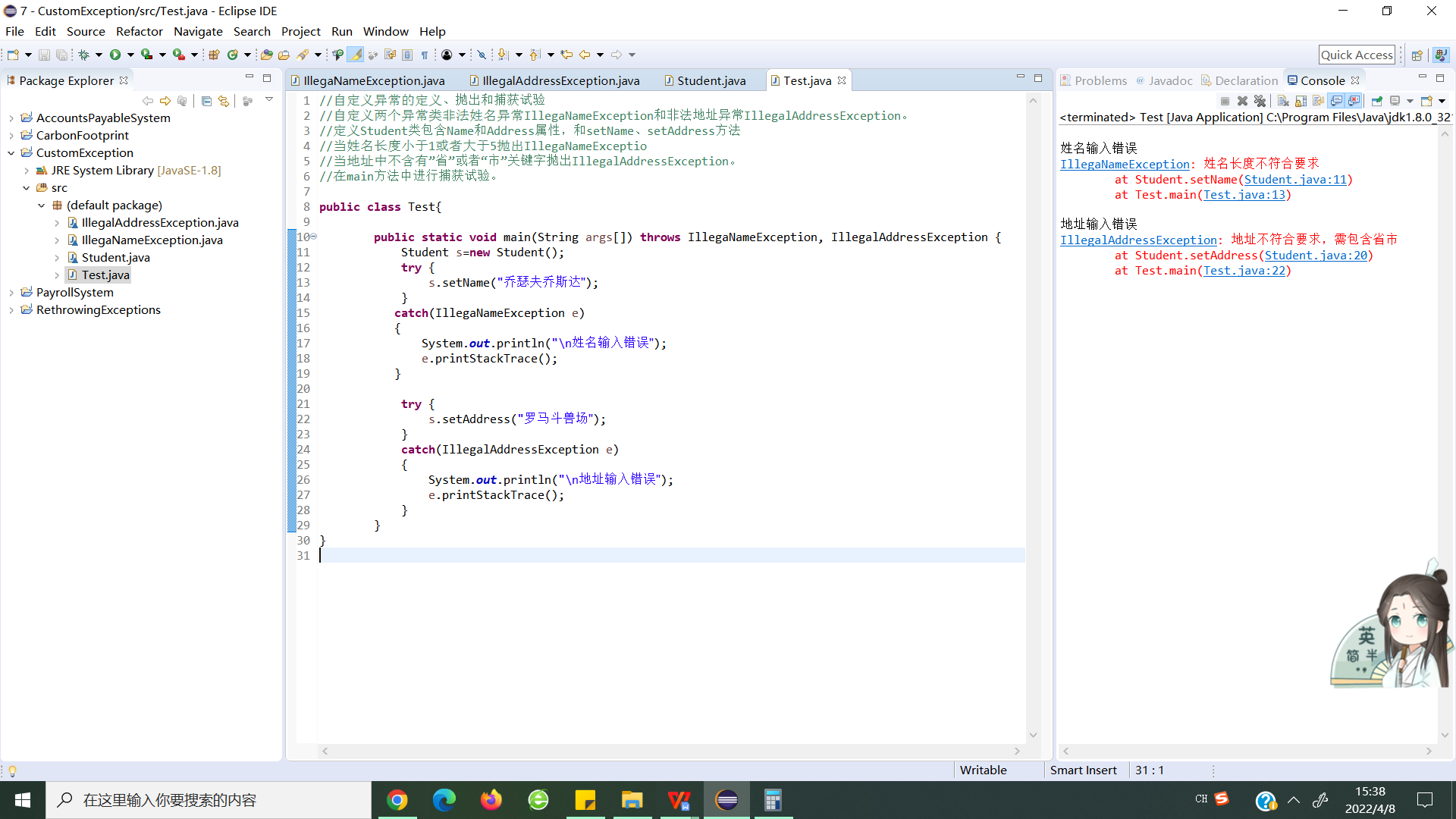
① 自定义两个异常类：非法姓名异常IllegaNameException和非法地址异常IllegalAddressException。

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。

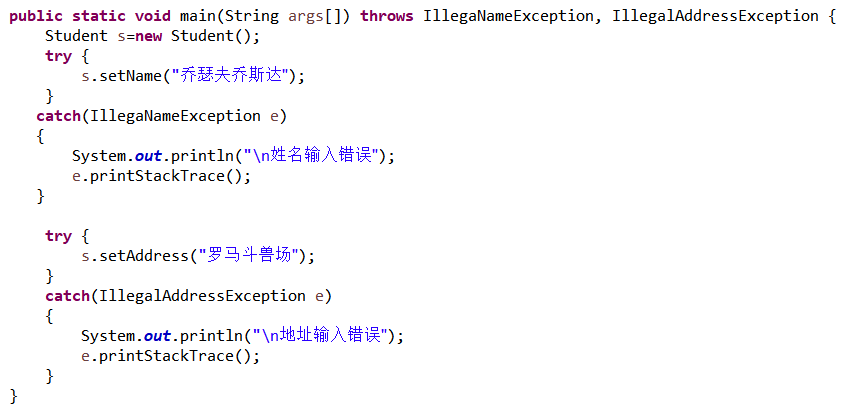
③ 编写程序抛出这两种异常，在main方法中进行捕获并合理地处理。

**（三）过程截图**

（1）全屏截图



（2）运行结果





**题目5：Rethrowing Exceptions**

**（一） 实验环境**

操作系统：Windows 10；

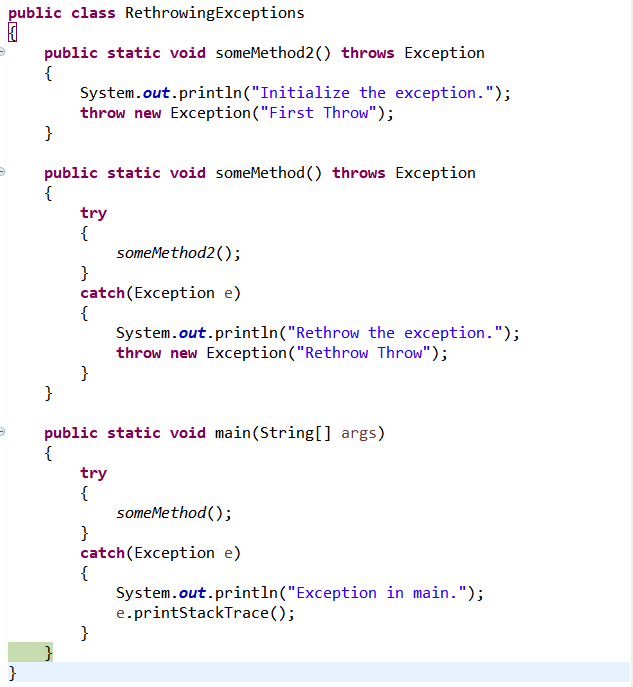
IDE：Eclipse Java 2018-12

编程语言：Java；

**（二） 实现过程**

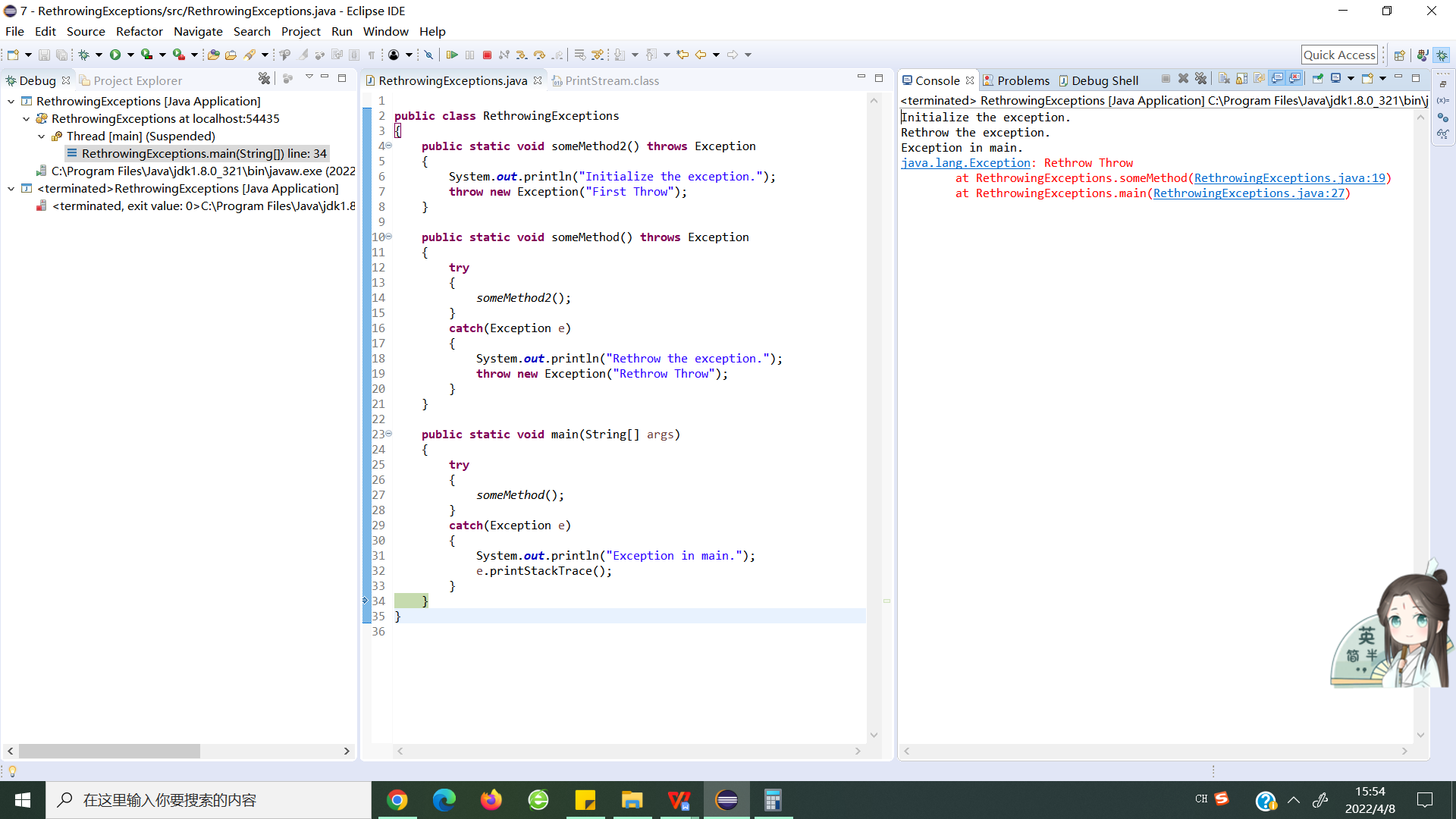
（1）设计思路

① 编写一个说明重新抛出异常的程序。定义方法someMethod 和 someMethod2。方法 someMethod2 最初引发异常。方法 someMethod 调用 someMethod2，捕获异常并重新抛出它。 从方法 main 调用 someMethod，并捕获重新抛出的异常。 打印此异常的堆栈跟踪。



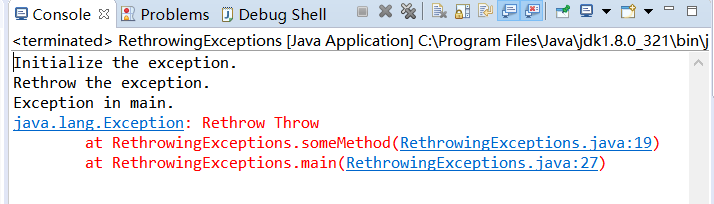
**（三）过程截图**

（1）全屏截图



1. 运行结果

方法 someMethod2 最初引发异常。方法 someMethod捕获异常并重新抛出它。 从方法 main 捕获重新抛出的异常。 打印此异常的堆栈跟踪。



**题目6：完善上周的模拟考试题目，补充必要的异常处理。**

**（一） 实验环境**

操作系统：Windows 10；

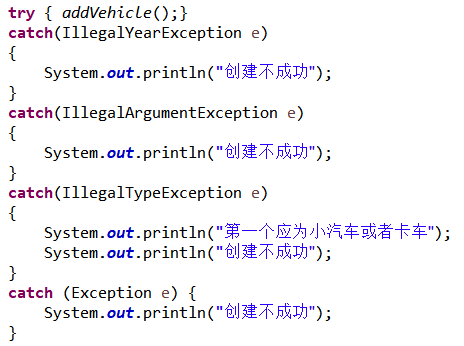
IDE：Eclipse Java 2018-12

编程语言：Java；

**（二） 实现过程**

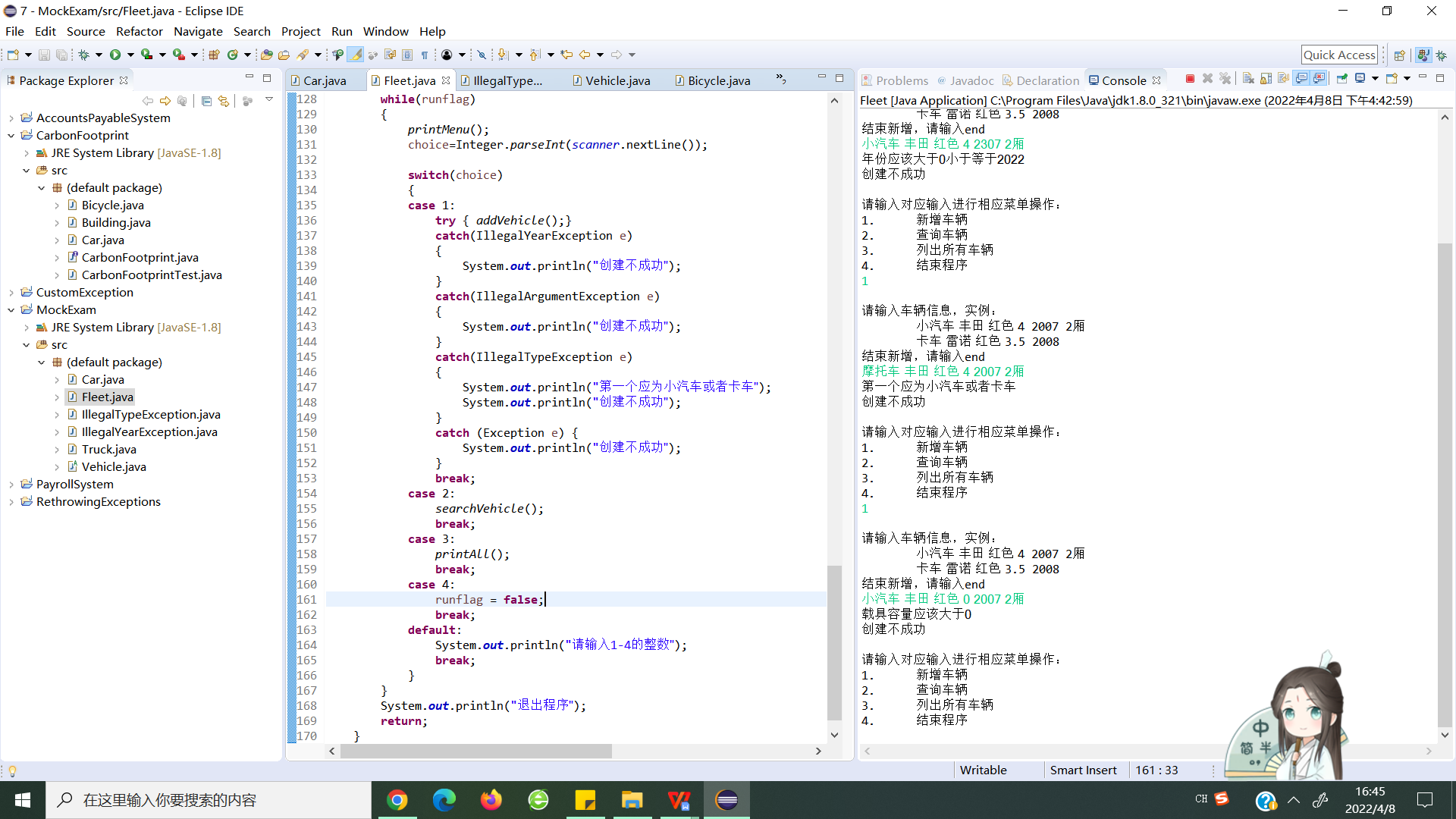
（1）设计思路

① 新增了Exception两个子类IllegalTypeException、IllegalYearException。前者用来抛出输入车辆类型不正确的错误，后者用来抛出输入年份不正确的错误。同时用IllegalArgumentException避免输入的载具容量小于或者等于0。



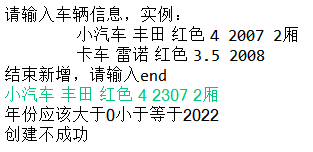
**（三）过程截图**

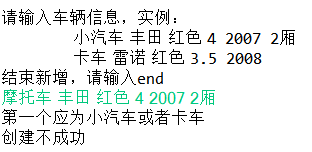
（1）全屏截图

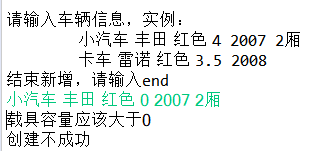


（2）运行结果

输入小汽车的出厂年份为2307年，捕获输入年份异常并输出，输入载具类型为摩托车，捕获输入类型异常并输出，输入小汽车载客量为0，捕获输入异常并输出。







**三、实验总结与心得记录**

在本次实验过程中，我练习了多态，熟悉了java的语法，熟悉了java类的定义，实例化和调用。体会到了JAVA语言的优点。