

西工大计算机学院高级语言程序设计

实 验 报 告

学号： 2020303245 姓名： 夏卓 班级： 10012006

课程名称	高级语言程序设计	实验课次	Java 第 2 次
实验项目	实验五 包、接口、类库	实验时间	2020/4/8
指导老师评议	<p>成绩评定：（满分为 100 分）</p> <p>指导教师评语：</p> <p>指导教师签名：</p>		
实验内容（要求、算法、步骤和方法）	请参见 实验说明书		
实验结果	<p>具体代码请参见附件中电子版的 java 源程序。源程序的简要说明如下：</p> <hr/> <p>Java 实验：</p> <p><u>DefiPackage.java、TestPackage.java</u></p> <p>说明： 在 DefiPackage.java 的第一句中输入 package 类名；可以创建一个包，编译后的字节码存储在此包中。如果需要在其他包中使用该包中的类，需要添加 import 语句将该包中的类导入，并且需要保证此类是 public 类型</p> <p><u>TestInterface.java</u></p> <p>说明： 创建接口 Speakable 和 Runner，然后创建两个类 Dog 和 Person 实现该接口</p> <p><u>Testabstract.java</u></p> <p>说明： 创建抽象类 Animal，然后创建 Dog 和 Person 类继承该抽象类</p> <p><u>interfaceDemo.java</u></p> <p>说明：</p>		

在 Vehicle 接口中添加两个带有一个参数的方法 start(String name) 和 stop(String name)，然后在两个名称分别为 Bike 和 Bus 的类中实现 Vehicle 接口

SubClass.java、DemoPackage.java

说明：

在 SubClass.java 中创建 ParentClass 类并让 SubClass 类继承此类，在 SubClass 的构造方法中调用父类的构造方法完成对父类变量的赋值，并在父类的构造方法中输出该变量，在 DemoPackage 中，由于不在同一个包中，因此需要先 import MainPackage.SubClass;

```
public class SubClass extends ParentClass {  
    public SubClass(int data) {  
        super(data);  
    }  
}
```

实验结果: (包括运行结果, UML 图, 以及实验要求中需要书面回答的内容)

DefiPackage.java、TestPackage.java

```
D:\Java_homework\lab2\Java&UML_Report_2020303245_夏卓_J05&J06&U01&U02\2020303245_夏卓_J05>javac -d . DefiPackage.java  
D:\Java_homework\lab2\Java&UML_Report_2020303245_夏卓_J05&J06&U01&U02\2020303245_夏卓_J05>dir  
驱动器 D 中的卷是 Data  
卷的序列号是 9EBF-A686  
  
D:\Java_homework\lab2\Java&UML_Report_2020303245_夏卓_J05&J06&U01&U02\2020303245_夏卓_J05 的目录  
2022/04/08 16:47 <DIR> .  
2022/04/08 16:47 <DIR> ..  
2022/04/07 19:40 132 DefiPackage.java  
2022/04/08 16:47 <DIR> p1  
1 个文件 132 字节  
3 个目录 579, 336, 753, 152 可用字节  
  
D:\Java_homework\lab2\Java&UML_Report_2020303245_夏卓_J05&J06&U01&U02\2020303245_夏卓_J05>cd p1  
D:\Java_homework\lab2\Java&UML_Report_2020303245_夏卓_J05&J06&U01&U02\2020303245_夏卓_J05\p1>dir  
驱动器 D 中的卷是 Data  
卷的序列号是 9EBF-A686  
  
D:\Java_homework\lab2\Java&UML_Report_2020303245_夏卓_J05&J06&U01&U02\2020303245_夏卓_J05\p1 的目录  
2022/04/08 16:47 <DIR> .  
2022/04/08 16:47 <DIR> ..  
2022/04/08 16:47 416 DefiPackage.class  
1 个文件 416 字节  
2 个目录 579, 336, 753, 152 可用字节
```

```
D:\Java_homework\lab2\test5>java TestPackage  
in method display()
```

```
D:\Java_homework\lab2\test5>
```

TestInterface.java

```
D:\Java_homework\lab2\test5>java TestInterface
狗的声音:汪、汪!
狗用四肢跑步
人们见面时经常说:您好!
人用两腿跑步
鸟的叫声:咕咕咕
```

- 该程序编译后生成了 5 个字节码文件，分别是:Speakable.class、Runner.class、Person.class、Dog.class、TestInterface.class

Testabstract.java

```
D:\Java_homework\lab2\test5>java Testabstract
狗的声音:汪、汪!
狗用四肢跑步
人们见面时经常说:您好!
人用两腿跑步
```

- 不同之处在于：由于 Java 不支持多继承，因此需要用一个抽象类 Animal 同时包含 speak() 和 run() 两个抽象方法，然后让 Person 类和 Dog 类继承此抽象类，这需要两个类之间有较为紧密的联系；而使用接口可以分别建立两个接口，并同时实现这两个接口，因此可以方便地在关系不紧密的类之间共享接口。

interfaceDemo.java

```
D:\Java_homework\lab2\test5>java interfaceDemo
自行车从宿舍楼出发
自行车在教学楼停下
公交车从西工大出发
公交车在市中心停下
```

SubClass.java、DemoPackage.java

```
D:\Java_homework\lab2\test5>java DemoPackage
data from ParentClass:3
```

实 验 心 得 体 会	<p>1. 实验中遇到的问题以及解决方法</p> <p>上机作业（2）中，需要在父类的构造方法中输出他的属性，而又要求在子类中完成对父类变量的赋值，为避免输出父类属性时父类变量可能还未完成初始化，考虑到调用子类构造方法时会先调用父类的构造方法，因此我在子类构造方法中调用了父类含参构造方法以完成初始化工作</p> <p>2. 实验体会</p> <p>通过本次实验我掌握了创建包与引用包的方法，创建并实现接口的方法，以及知道了抽象类与接口的不同之处，还明白了构造方法调用的顺序。创建包需要在开头添加 Package 类名，引用包需要将其 import 进来，还需保证此类需是 public 属性。</p>
----------------------------	---

课程名称	高级语言程序设计	实验课次	Java 第 2 次
实验项目	实验六 异常处理	实验时间	2020/4/8
指导老师评议	<p>成绩评定：（满分为 100 分）</p> <p>指导教师评语：</p> <p>指导教师签名：</p>		
实验内容（要求、算法、步骤和方法）	请参见 实验说明书		
实 验 结	<p>具体代码请参见附件中电子版的 java 源程序。源程序的简要说明如下：</p> <p>-----</p> <p>Java 实验： CatchDemo.java 说明： 利用 try-catch-finally 语句同时捕获数组越界和被 0 除的异常并进</p>		

果	<p>行相应处理</p> <p>TestException.java 说明： 创建用户自定义异常，捕获并处理 num<0 的异常</p> <p>StringToNum.java 说明： 输入一行字符串，若为 int 型或 double 型则将其转化为相应的数据类型，否则产生异常，输出”含有非法字符输入”以提醒用户重新输入</p> <hr/> <p>实验结果:(包括运行结果, UML 图, 以及实验要求中需要书面回答的内容)</p> <p>CatchDemo.java</p> <pre> D:\Java_homework\lab2\test11>java CatchDemo 4 / 2 is 2 Finally 已执行 Can't be divided by zero Finally 已执行 16 / 4 is 4 Finally 已执行 32 / 4 is 8 Finally 已执行 Can't be divided by zero Finally 已执行 128 / 8 is 16 Finally 已执行 No matching element found. Finally 已执行 No matching element found. Finally 已执行 </pre> <ul style="list-style-type: none"> • 异常处理语句 try-catch-finally 的处理机制：先运行 try 中的程序，如果产生了异常则 catch 语句捕获该异常，进入相应的 catch 语句执行程序，无论是否发生异常，最后都会执行 finally 语句，程序不会中断退出。 • 如果没有异常处理，直接输出两个数组对应元素相除的结果，运行截图如下： <pre> D:\Java_homework\lab2\test11>java CatchDemo 4 / 2 is 2 Exception in thread "main" java.lang.ArithmeticException: / by zero at CatchDemo.main(CatchDemo.java:6) </pre> <p>会产生运行时异常，这是因为整数除零导致了 ArithmeticException</p>
---	---

异常，由于没有异常处理程序，此异常被 Java 虚拟机捕获，抛出异常后结束了程序的运行

TestException.java

```
D:\Java_homework\lab2\test11>java TestException
登记出错，类别:3
本次登记操作结束
```

- 本程序中的 throws 语句表明将此方法运行时可能产生的用户自定义异常抛出给调用函数处理，throw 语句表明若 num 为负数则产生一个用户自定义异常对象，并主动将其抛出
- 本程序的用户自定义异常继承了 Java 中的异常类，在构造方法中调用了父类的构造方法，并设置了自己的出错类别
- 本程序在产生异常的 regist() 方法中将异常抛出，在调用此方法的 manager() 方法中捕获此异常，并输出错误信息
- 如果将程序中的“public void regist(int num) throws MyException”改为“public void regist(int num)”，会出现

```
D:\Java_homework\lab2\test11>javac -encoding UTF8 TestException.java
TestException.java:17: 错误: 未报告的异常错误UserException; 必须对其进行捕获或声明以便抛出
        throw new UserException("人数为负值，不合理", 3);
        ^
TestException.java:25: 错误: 在相应的 try 语句主体中不能抛出异常错误UserException
    } catch (UserException e) {
      ^
2 个错误
```

StringToNum.java

```
D:\Java_homework\lab2\test11>java StringToNum
2.432
已成功转化为double型数据:2.432
转换结束
```

```
D:\Java_homework\lab2\test11>java StringToNum
-123
已成功转化为int型数据:-123
转换结束
```

```
D:\Java_homework\lab2\test11>java StringToNum
1.32a
含有非法字符输入
转换结束
```

实 验 心 得 体 会	<p>3. 实验中遇到的问题以及解决方法</p> <p>上机实验中，需要判断是整数类型还是浮点数类型，并进行相应的转换，我通过查找字符串中是否含有小数点‘.’来进行区别。</p> <p>4. 实验体会</p> <p>通过本次实验我掌握了异常的概念及异常处理的机制，能够通过 try-catch-finally 语句处理异常，能够实现用户自定义异常。需要注意 throws 与 throw 的不同之处，如果没有异常处理语句而产生了异常，会导致程序中断退出。</p>
----------------------------	---

课程名称	高级语言程序设计	实验课次	UML 第 1 次
实验项目	实验一 UML 建模基础	实验时间	2020/4/8
指导老师评议	<p>成绩评定：（满分为 100 分）</p> <p>指导教师评语：</p> <p>指导教师签名：</p>		

实验内容（要求、算法、步骤和方法）	请参见 实验说明书
实验结果	<p>具体代码请参见附件中电子版的 java 源程序。源程序的简要说明如下：</p> <hr/> <p>UML 实验： 1 1.mdl 说明： 根据相应关系绘图 关联关系为实直线箭头 依赖关系为虚直线箭头 泛化关系为实直线空心箭头 实现关系为虚直线空心箭头</p> <hr/> <p>实验结果：(包括运行结果，UML 图，以及实验要求中需要书面回答的内容) 1 1.mdl</p> <div style="text-align: center;">  <pre> classDiagram class Class1 class Class2 class Class3 class Class4 class Class5 class Class6 class Class7 class Class8 class Class9 interface interface Class2 --> Class1 : 关联关系 Class3 -.-> Class4 : 依赖关系 Class5 -- > Class6 : 泛化关系 Class7 .. > interface : 实现关系 Class9 o-- Class8 : 聚合关系 </pre> </div> <p>• 可见性： + ：表示 public 属性； - ：表示 private 属性； # ：表示 protected 属性； ~ ：表示 package 属性。</p>

	<ul style="list-style-type: none"> • UML 在软件工程中的作用： UML 拥有足够的表达能力可以处理现代软件开发中出现的所有概念，可以用来对软件密集系统进行可视化建模，也能为面向对象开发系统的产品进行说明、可视化、构造和编制文档，适用于系统开发过程中从需求规格描述到系统完成后测试的不同阶段。 • 使用 UML 建模的必要性： <ol style="list-style-type: none"> 1. UML 简单、直观具有国际化标准性。对于大规模、复杂系统的设计，团队间使用 UML 交流比用文字交流方便得多，也便于系统文档的制定和项目的管理 2. UML 作为一种建模语言，它使开发人员专注于建立产品的模型和结构，而不必关心用什么程序语言和算法实现。 3. UML 可以贯穿软件开发周期的每一个阶段，适于数据建模、业务建模、对象建模、组件建模 4. UML 中的各种模型可以帮助开发人员更好的理解业务流程，建立更可靠、更完善的系统模型，也利于用户对产品的理解。
实 验 心 得 体 会	<p>5. 实验中遇到的问题以及解决方法</p> <p>一开始对 rose 的操作界面和使用方法掌握不熟，有些线形没有找到，后来在网上查阅学习后才掌握了初步的使用方法</p> <p>6. 实验体会</p> <p>本次实验内容比较简单，只是最开始做实验时理论课还没上到这部分，只能靠自学，一开始感觉这软件界面好老，画的图也不甚美观，但找了会其他的代替品也各有各的问题，因此最终还是用上了 Rational Rose, 习惯之后感觉也还不错</p>

课程名称	高级语言程序设计	实验课次	UML 第 1 次
实验项目	实验二 UML 类图	实验时间	2020/4/8
指导老师评议	成绩评定：（满分为 100 分） 指导教师评语： 指导教师签名：		
实验内容（要求、算法、步骤和方法）	请参见 实验说明书		
实验结果	具体代码请参见附件中电子版的 java 源程序。源程序的简要说明如下： <hr/> UML 实验： <u>2 3.mdl</u> 说明： 由于类 A 由类 B 的一个实类和类 C 的 1 个或多个实类构成，因此 A 类与 B 类、C 类是聚合关系，多重性为：一个 A 类包含 1 个 B 类，及 1 个或多个 C 类 <u>2 4.mdl</u> 说明： 课程与学生、老师、教材之间均为关联关系 <u>2 5.mdl</u> 、 <u>2 6.mdl</u> 说明： 建立商品类 Merchandise，具有商品基本属性，管理系统 ManageSystem 中包含多个商品类，具有添加，删除商品，修改商品信息等方法 若需要对损坏商品打折，需要在商品类中添加布尔属性 IsDamaged 来判断商品是否损坏，在管理系统类中添加 discount(double rate) 方法，按照 rate 折对所有损坏商品打折 在管理系统中添加询问方法，分别按颜色和外形访问商品，返回满足该属性的商品类数组 <u>2 7.mdl</u> 说明： rose 中静态属性用下划线标识，Data 为一个类，Invoice 与其是关联关系 <hr/> 实验结果：(包括运行结果，UML 图，以及实验要求中需要书面回答的内容) • 类的定义及三要素		

类是面向对象系统组织结构的核心，是对一组具有相同属性、操作、关系和语义的事物的抽象。类的三要素包括：类的名称、类的属性和类的操作

- 类的属性和方法的可见性有：

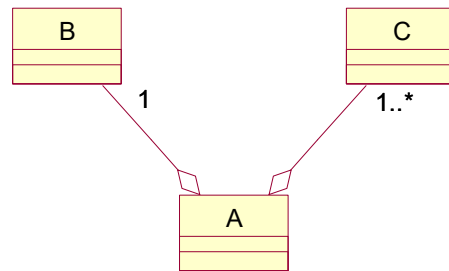
public 属性，在 UML 中用 ‘+’ 表示；

private 属性，在 UML 中用 ‘-’ 表示；

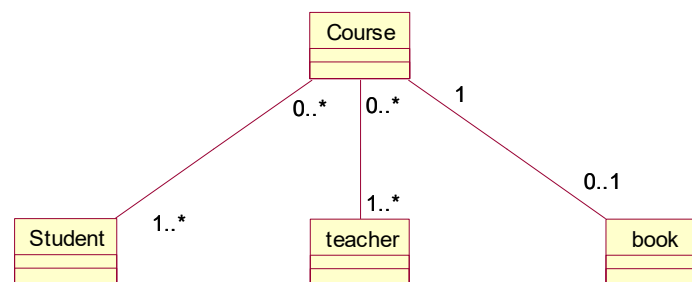
protected 属性，在 UML 中用 ‘#’ 表示；

package 属性在 UML 中用 ‘~’ 或空白符表示。

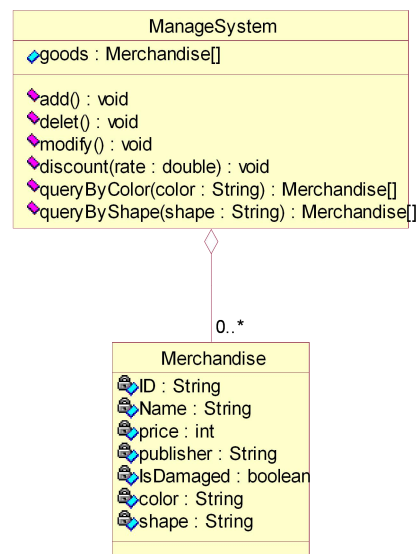
2 3. mdl



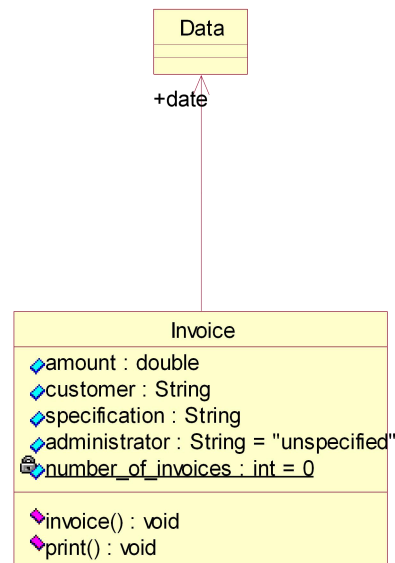
2 4. mdl



2 5. mdl、2 6. mdl

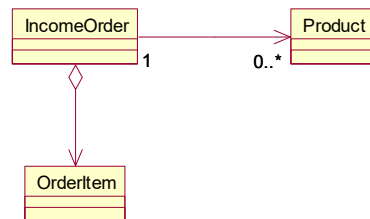


2 7. mdl



• 第 8 题

IncomeOrder 中含有 OrderItem，因此 IncomeOrder 与 OrderItem 应该是关联关系，Product 是产品信息，入库单包含多个产品信息，而一个产品信息对应一个入库单，因此正确的类图应是：



• 类图在建模过程中的重要作用

类图的作用是对系统的静态视图进行建模，当对系统的静态视图进行建模时，通常使用以下三种方式来使用类图。

1. 为系统的词汇建模。
2. 模型化简单的协作。
3. 模型化逻辑数据库模式。

在设计数据库时，通常将数据库模式看作为数据库概念设计的蓝图，在很多领域中，都需要在关系数据库或面向数据库中存储永久信息。系统分析者可以使用类图来对这些数据库进行模式建模。

• 在设计绘制类图的过程中应注意的问题

首先应当注意的是类之间的关系的连线问题，应该明确何种关系运用何种直线连接，其次是类属性的声明格式规范，设计类图应注意类图的简洁性和完整性，需要注意类之间的联系，特别是关联关系还可以细分为聚合关系和组合关系，他们之间的生命周期不同，线条的绘制也不同

<p>实 验 心 得 体 会</p>	<p>7. 实验中遇到的问题以及解决方法</p> <p>一开始对于类的多重性关系没有搞清楚，不知道放在哪里，后来明白了它表示这个关联的类可以有几个实例，借助理论课上老师对拳击手与手之间的举例，我明白了他的使用方法以及如何绘制。</p> <p>8. 实验体会</p> <p>通过本次实验，我掌握了 UML 中类的表示方法以及类与类之间的各种关系代表的含义及表示方法，理解了类的多重性关系。实验中我还进行了 Java 语言的逆向工程，成功生成了 UML 图形，体会到了工程中 UML 建模的必要性和便利性。</p>
--	---