**哈尔滨工业大学（威海）**

**Java语言实验报告**

|  |  |  |  |
| --- | --- | --- | --- |
| 课程名称 | Java语言 | 课程编号 | SE33601 |
| 实验名称 | 异常处理及文件内容读取 | | |
| 实验类型 | 验证与编程 | 实验学时 | 2学时 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 班号 | 1811103 | 学号 | 181110315 | 姓名 | 王少博 |
| 开始 | 2019.10.16 | 截止 | 2019.10.27 | 成绩 |  |
| 批语 |  | | | | |

目录

[1 实验目的 1](#_Toc22067555)

[2 实验内容 1](#_Toc22067556)

[2.1 手工编写shopping.txt文件 2](#_Toc22067557)

[2.2 读shopping.txt到发票系统 2](#_Toc22067558)

[2.3 实验步骤 2](#_Toc22067559)

[3 实验结果 5](#_Toc22067560)

# 1 实验目的

本次实验有三个主要目的：

1. 掌握文件读写的基本流程并进行异常处理；
2. 加深理解IO设计中的具体类与功能类，以及如何用功能类包装具体类；
3. 掌握通过查询API方式，实现文件的读写。

# 2 实验内容

在实验3发票系统中，根据业务需求，一张发票可以分成三个部分：头部、主部和尾部。发票的格式大致如图1-4所示（如果扩展了其他语言类型，发票样式会更丰富）：

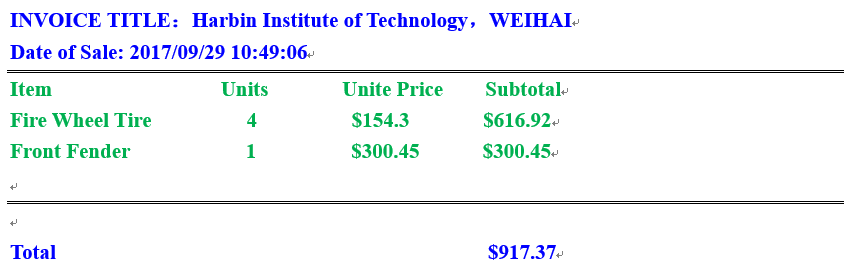


图1 英文发票样式

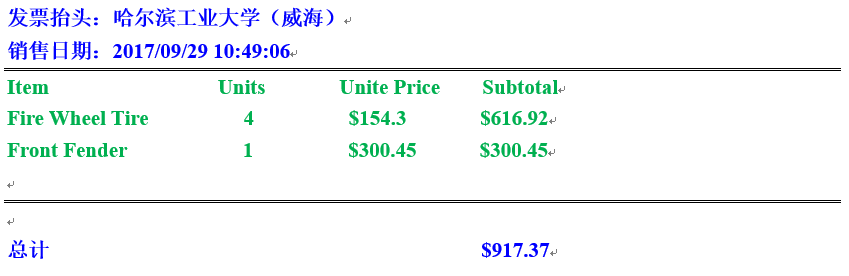


图2 中文发票样式



图3 英文头尾在外，中文头尾在里的发票样式

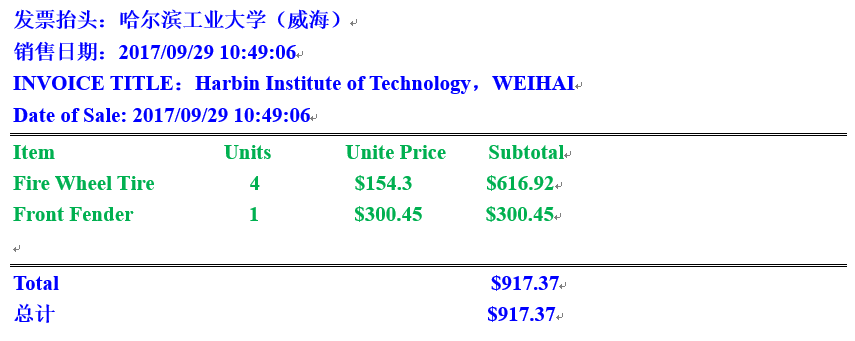


图4 中文头尾在外，英文头尾在里的发票样式（不要求，但应该能实现）

根据实验3第二个实验内容的业务需求，发票的主体部分，都采用英文；根据购买物品的不同，发票主体会产生不同的行，这部分必须设计成能够不断添加的，而不是如图1-图4所显示的仅仅Fire Wheel Tire和Front Fender两项；除外，同一物品不必合并，例如图5所示；为了满足国际商务需要，对发票的头尾也进行了要求，例如，不同语言的发票头尾需要组合使用（不存在一种语言的头，另一种语言发票尾的情况）；在满足OCP的前提下，发票头尾所用语言能够不断扩展等要求。

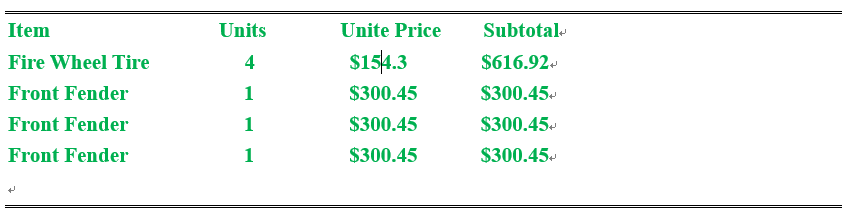


图5 发票行扩展示意图（不用合并同类商品）

## 2.1 手工编写shopping.txt文件

本次实验，要求改变发票行的输入方式，改成由文件读取发票行。假设在D：\下存在一个shopping.txt文件，其内容是多个发票行组成的，如图6所示：

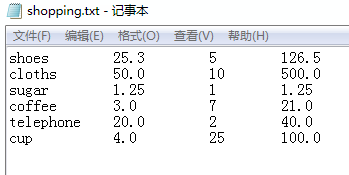


图6 shopping.txt文件内容

请自己编写该shopping.txt文件，并放到D盘根目录下。如果操作系统没有设置D盘，可放C盘或者其他盘符，使用Mac，Linux等非windows系统，请自行定义路径。

## 2.2 读shopping.txt到发票系统

完成shopping.txt编写后，请按仔细阅读如下要求：

1. 在实验03的基础上，完成本实验类图的设计。不允许修改已经存在的类及其实现代码（测试类除外），只允许在实验03的基础上增加文件操作相关的类，以完成文件读取，并对读取形成的发票主体进行包装。
2. 代码必须对可能出现的异常进行处理。
3. 可以与同学讨论方案，但是源代码必须自己写；可以查看API，或者通过搜索引擎搜索，选择合适的“java IO”。
4. 凡是不从文件shopping.txt读取信息，而是直接在代码中写购物内容的一律零分。
5. 凡是偷懒不进行异常处理的（包括在main方法中抛出异常的），一律零分；

## 2.3 实验步骤

* **实验步骤1：绘制新增文件操作类后的整体类图（不必绘制测试类）：**

|  |
| --- |
|  |

* **实验步骤2：编写代码：**

**新增加的文件操作类的代码：**

|  |
| --- |
| import java.io.FileInputStream;  import java.io.FileNotFoundException;  import java.util.Scanner;  public class DIY\_SCANNER {  private FileInputStream fileInputStream;  private Scanner scanner;  public DIY\_SCANNER(String filename) throws FileNotFoundException {  fileInputStream = new FileInputStream(filename);  scanner = new Scanner(fileInputStream);  }  public void ReadFromTXT(Invoice invoice) {  while (scanner.hasNext()) {  String str = scanner.nextLine();  Scanner sc = new Scanner(str);  String invoiceStr = sc.next();  Double invoiceDouble = sc.nextDouble();  Integer invoiceNum = sc.nextInt();  sc.nextDouble();  invoice.addInvoiceLine(new InvoiceLine(invoiceStr, invoiceDouble, invoiceNum));  }  }  public void setFileInputStream(FileInputStream fileInputStream) {  this.fileInputStream = fileInputStream;  }  public void setScanner(Scanner scanner) {  this.scanner = scanner;  }  public FileInputStream getFileInputStream() {  return this.fileInputStream;  }  public Scanner getScanner() {  return this.scanner;  }  } |

**其他各类的代码粘贴如下：**

**Invoice类的代码：**

|  |
| --- |
| import java.io.Serializable;  import java.util.ArrayList;  public abstract class Invoice {  private String name;  private String time;  private ArrayList<InvoiceLine> invoiceLineList;  public abstract void print();  public void addInvoiceLine(InvoiceLine invoiceLine) {  invoiceLineList.add(invoiceLine);  }  public void removeInvoiceLine(InvoiceLine invoiceLine) {  invoiceLineList.remove(invoiceLine);  }  public ArrayList<InvoiceLine> getInvoiceLineList() {  return invoiceLineList;  }  public double getTotal() {  double total = 0;  for (int i = 0; i < getInvoiceLineList().size(); i++) {  total += getInvoiceLineList().get(i).getSubtotal();  }  return total;  }  public String getName() {  return name;  }  public void setName(String name) {  this.name = name;  }  public String getTime() {  return time;  }  public void setTime(String time) {  this.time = time;  }  public Invoice() {  invoiceLineList = new ArrayList<InvoiceLine>();  }  public Invoice(String name) {  setName(name);  }  } |

**InvoiceLine类的代码：**

|  |
| --- |
| public class InvoiceLine {    private String item;  private double unitPrice;  private int units;  public InvoiceLine(String item, double unitPrice, int units) {  setItem(item);  setUnitPrice(unitPrice);  setUnits(units);  }  public void printLine() {  System.out.println(getItem() + "\t\t" + getUnits() + "\t\t$" + getUnitPrice() + "\t\t\t$" + String.format("%.3f", getSubtotal()));  }  public double getSubtotal() {  return getUnitPrice() \* getUnits();  }  public String getItem() {  return item;  }  public void setItem(String item) {  this.item = item;  }  public double getUnitPrice() {  return unitPrice;  }  public void setUnitPrice(double unitPrice) {  this.unitPrice = unitPrice;  }  public int getUnits() {  return units;  }  public void setUnits(int units) {  this.units = units;  }  } |

**BodyDecorator类的代码：**

|  |
| --- |
| public abstract class BodyDecorator extends Invoice {  private Invoice invoice;  public double getTotal() {  return getInvoice().getTotal();  }  public BodyDecorator(Invoice invoice) {  setInvoice(invoice);  }  public Invoice getInvoice() {  return invoice;  }  public void setInvoice(Invoice invoice) {  this.invoice = invoice;  }  } |

**Body类的代码：**

|  |
| --- |
| public class Body extends Invoice {  public void print() {  System.out.println("Item" + "\t\t" + "Units" + "\t\t" + "Unite Price" + "\t\t" + "Subtotal" + "\t\t");  for (int i = 0; i < getInvoiceLineList().size(); i++) {  getInvoiceLineList().get(i).printLine();  }  }  } |

**ChineseSyle类的代码：**

|  |
| --- |
| public class ChineseStyle extends BodyDecorator {  public ChineseStyle(Invoice invoice) {  super(invoice);  }  public void print() {  System.out.println("发票抬头: " + getName());  System.out.println("销售日期: " + getTime());  getInvoice().print();  System.out.println("总计:" + "\t\t\t\t\t\t\t$" + String.format("%.3f", getTotal()));  }  } |

**EnglishStyle类的代码：**

|  |
| --- |
| public class EnglishStyle extends BodyDecorator {  public EnglishStyle(Invoice invoice) {  super(invoice);  }  public void print() {  System.out.println("INVOICE TITLE: " + getName());  System.out.println("Date Of Sale: " + getTime());  getInvoice().print();  System.out.println("Total:" + "\t\t\t\t\t\t\t$" + String.format("%.3f", getTotal()));  }  } |

**测试类的代码：**

|  |
| --- |
| import java.util.Date;  public class Client {  public static void main(String[] args) {  Invoice invoice = new Body();  try{  DIY\_SCANNER dsc = new DIY\_SCANNER("D:/subject/java/shopping.txt");  dsc.ReadFromTXT(invoice);  invoice = new ChineseStyle(invoice);  invoice.setName("迪克");  invoice.setTime(new Date().toString());  invoice = new EnglishStyle(invoice);  invoice.setName("Dick");  invoice.setTime("公元21世纪");  invoice.print();  } catch (Exception e){  System.out.println("找不到文件");  }finally{  System.out.println("--------------------------------------------------------------------------");  }  }  } |

# 3 实验结果

* 运行结果（1）：中文的发票头尾的发票截图

|  |
| --- |
|  |

* 运行结果（2）：英文头发票尾的发票截图

|  |
| --- |
|  |

* 运行结果（3）：中英文结合的样式发票截图（如图3英文头尾在外，中文头尾在里的发票样式）

|  |
| --- |
|  |