Java进阶1 第1天

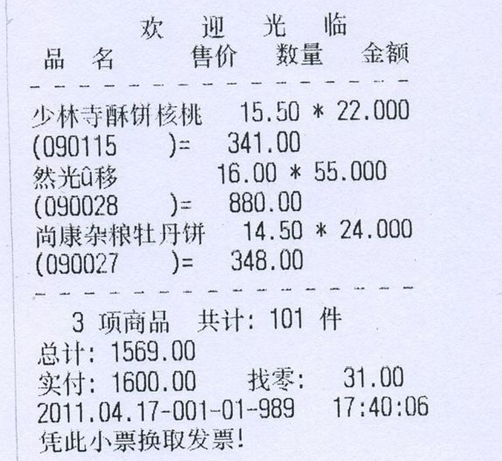
**【学习目标】理解、了解、应用、记忆**

通过今天的学习，参训学员能够：（解释的时候说出二级目标的掌握程度）

1. **【应用】独立编写超市购物小票案例**
2. 【理解】阐述超市购物小票需求及所用到的知识点
3. 【应用】独立编码定义商品项类
4. 【应用】独立编码完成小票数据初始化功能
5. 【应用】独立编码完成购物小票主干逻辑
6. 【应用】独立编码完成为商品数量和金额赋值功能
7. 【应用】独立编码完成购物小票打印功能
8. **【理解】阐述并解释面向对象相关内容**
9. 【应用】独立编码完成一个Person类的定义
10. 【应用】能够动手操作Eclipse中的三个视图
11. 【理解】阐述基本数据类型在内存中如何存储
12. 【理解】阐述引用数据类型在内存中如何存储
13. 【理解】解释this关键字的作用及用法
14. 【理解】阐述匿名对象的使用格式及使用场景

# 基础复习超市购物小票案例

## 超市购物小票需求分析



票脚：统计和提醒信息

票体：主体内容

票头：说明信息

模拟真实购物逻辑，以上述图片为需求原型，根据不同购买物品，完成购物小票内容打印到控制台。(简化项目逻辑，票脚部分只完成数据统计)

**数据：**

 将超市购物小票案例中零散数据(名称、货号、单价、数量、计价单位、金额)封装为货物对象。

 所有的货物对象放置到集合中统一管理。

**逻辑：**

用户循环进行三个操作：

 输入购买数量，代表为所购买货物的数量赋值，从而计算每项商品金额

 打印小票，将已有数据打印

 退出系统(因为该程序为循环操作，无法终止，如果不想再进行操作，则退出系统)

## 超市购物小票知识点分析

**复习知识点:**

变量定义：记录部分数据

集合的使用：用于存储多个完整数据

运算符的使用：计算变量

流程控制语句的使用：

if语句：用于判断是否购买了商品

switch语句：用于进行三个操作(输入购买数量、打印小票、退出系统)的分支

for语句：switch外部使用for循环，当执行一个操作后，继续进行下一个操 作选择

方法封装：

由于代码量比较多，可以将输入购买数量、打印小票封装为方法使用

字符串/键盘录入：

打印小票完整过程均为字符串的拼写及打印，数量输入为键盘录入Scanner 完成

**步骤:**

定义GoodsItem类(名称,货号,单价,数量,计价单位,金额)

成员位置定义集合，定义方法向集合中存储所有GoodsItem数据

实现主干逻辑

打印欢迎语句

使用for/while进行循环进行操作

使用switch给出操作选择：1输入购买数量2打印小票3退出

完成switch中三个case的逻辑

运用方法封装功能：将1输入购买数量、2打印小票封装为独立方法，在case中 调用，提高代码可读性。

## 超市购物小票商品项类的定义

商品项类，将要购买的商品封装到一个GoodItems类中,编码如下:

### 案例代码一

|  |
| --- |
| **package** com.igeek\_03;  /\*\*  \* **@ClassName**: GoodsItem  \* **@Description**: 商品项类  \* **@date** 2017年11月13日 上午10:41:21  \* Company www.igeekhome.com  \*  \* 定义商品项类:  \* 名称,货号,单价,数量,计价单位,金额  \*  \*/  **public** **class** GoodsItem {  // 成员变量  /\*\*  \* **@Fields** name : 名称  \*/  **private** String name;  /\*\*  \* **@Fields** id : 货号  \*/  **private** String id;  /\*\*  \* **@Fields** price : 单价  \*/  **private** **double** price;  /\*\*  \* **@Fields** number : 数量  \*/  **private** **int** number;  /\*\*  \* **@Fields** unit : 计价单位  \*/  **private** String unit;  /\*\*  \* **@Fields** money : 金额  \*/  **private** **double** money;  // 构造方法  /\*\*  \* **@Title**: GoodsItem  \*/  **public** GoodsItem() {  }  /\*\*  \* **@Title**: GoodsItem  \* **@param** name  \* **@param** id  \* **@param** price  \* **@param** number  \* **@param** unit  \* **@param** money  \*/  **public** GoodsItem(String name, String id, **double** price, **int** number, String unit, **double** money) {  **this**.name = name;  **this**.id = id;  **this**.price = price;  **this**.number = number;  **this**.unit = unit;  **this**.money = money;  }  /\*\*  \* **@return** the name  \*/  **public** String getName() {  **return** name;  }  /\*\*  \* **@param** name the name to set  \*/  **public** **void** setName(String name) {  **this**.name = name;  }  /\*\*  \* **@return** the id  \*/  **public** String getId() {  **return** id;  }  /\*\*  \* **@param** id the id to set  \*/  **public** **void** setId(String id) {  **this**.id = id;  }  /\*\*  \* **@return** the price  \*/  **public** **double** getPrice() {  **return** price;  }  /\*\*  \* **@param** price the price to set  \*/  **public** **void** setPrice(**double** price) {  **this**.price = price;  }  /\*\*  \* **@return** the number  \*/  **public** **int** getNumber() {  **return** number;  }  /\*\*  \* **@param** number the number to set  \*/  **public** **void** setNumber(**int** number) {  **this**.number = number;  }  /\*\*  \* **@return** the unit  \*/  **public** String getUnit() {  **return** unit;  }  /\*\*  \* **@param** unit the unit to set  \*/  **public** **void** setUnit(String unit) {  **this**.unit = unit;  }  /\*\*  \* **@return** the money  \*/  **public** **double** getMoney() {  **return** money;  }  /\*\*  \* **@param** money the money to set  \*/  **public** **void** setMoney(**double** money) {  **this**.money = money;  }  } |