

## 第 8 天 Java 基础语法

### 今日内容介绍

◆ Eclipse 开发工具

◆ 超市库存管理系统

## 第1章 Eclipse 开发工具

Eclipse 是功能强大 Java 集成开发工具。它可以极大地提升我们的开发效率。可以自动编译，检查错误。在公司中，使用的就是 Eclipse 进行开发。

### 1.1Eclipse 的下载、安装、卸载

- 下载 <http://www.eclipse.org>
- 安装（只需要解压后就能使用）
- 卸载（只需要将文件夹删除就可以了）
- 注意： 开发软件的安装目录中，尽量不要出现空格与中文

### 1.2Eclipse 的使用

在当前阶段，我们只需要完成最基本的 Java 文件编写，其他功能跟随日常教学慢慢渗透学习。  
使用步骤：

- 1. 开启，选择工作空间，如下图
  - 工作空间：指的是将来编写的 java 源文件存放的位置
  - 下图中的复选框“☐”，如果选中，表示使用当前工作空间为 Eclipse 的默认工作空间，以后启动 Eclipse 时，不在提示选择工作空间。

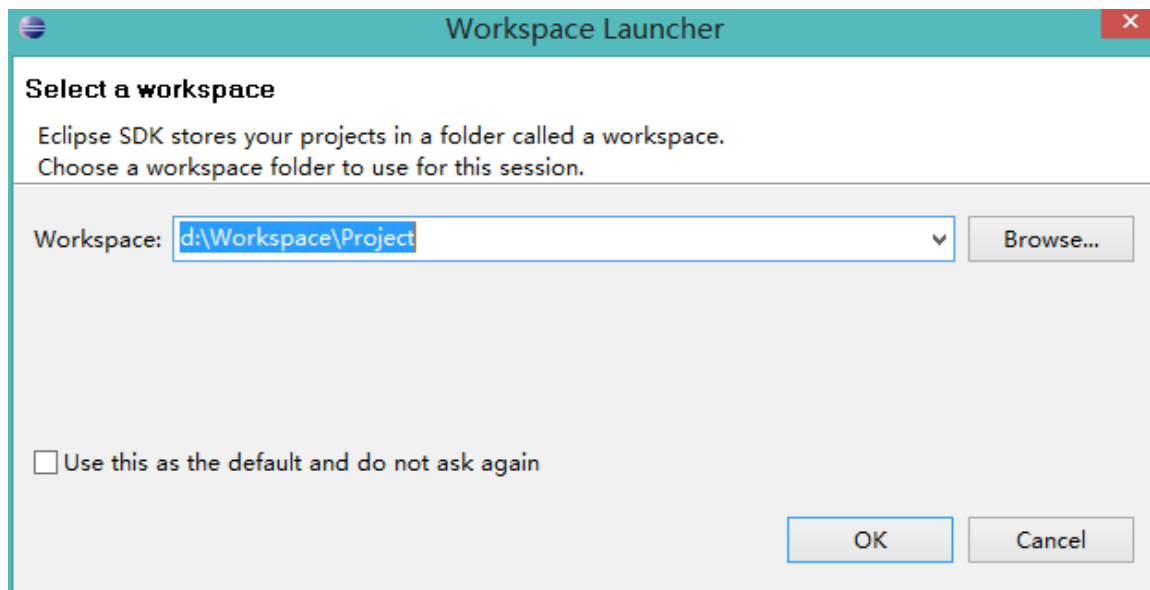


图1-1 选择工作空间

- 2. 在左侧空白区域右键 或者 选中 File 菜单栏， 选择 new--> 创建 java 项目编写项目名称，点击 Finish 完成。如下图

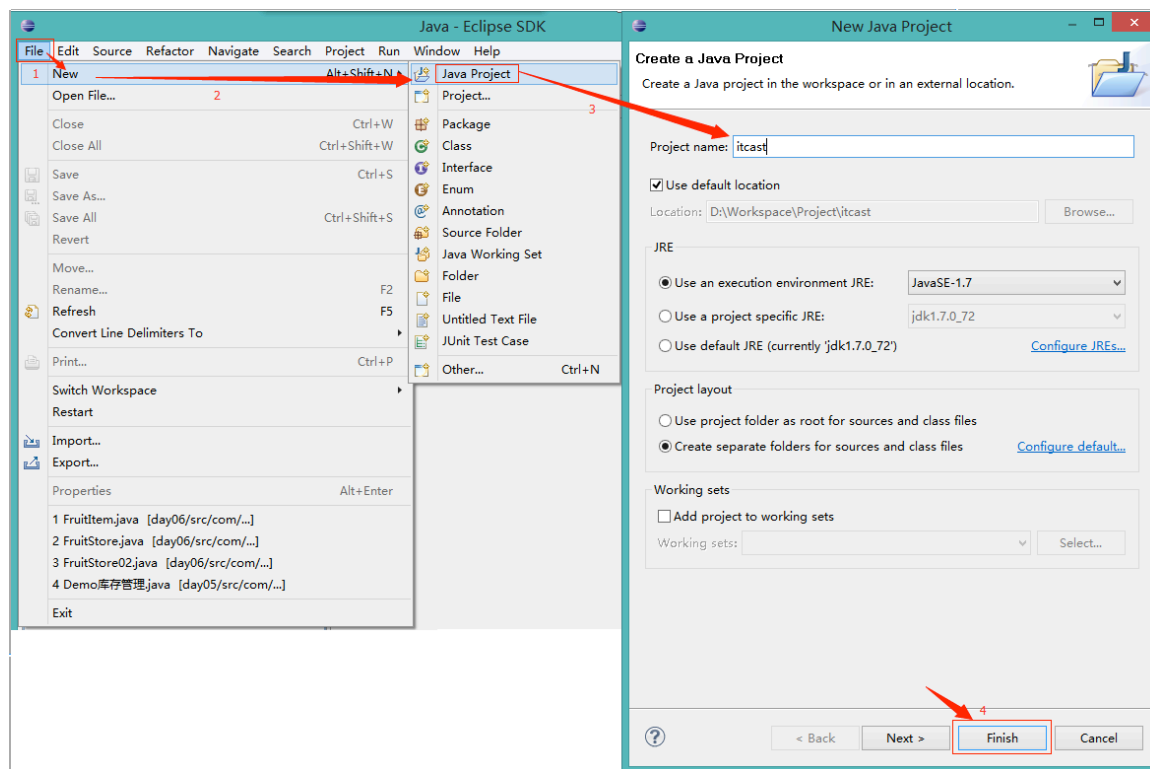


图1-2 创建 java 项目

- 3. 展开项目，选中源包 SRC 后，在项目的 src 目录下右键创建类，指定类的名字 Hello，创建 main 方法并编写源代码 `System.out.println("HelloWorld")`;

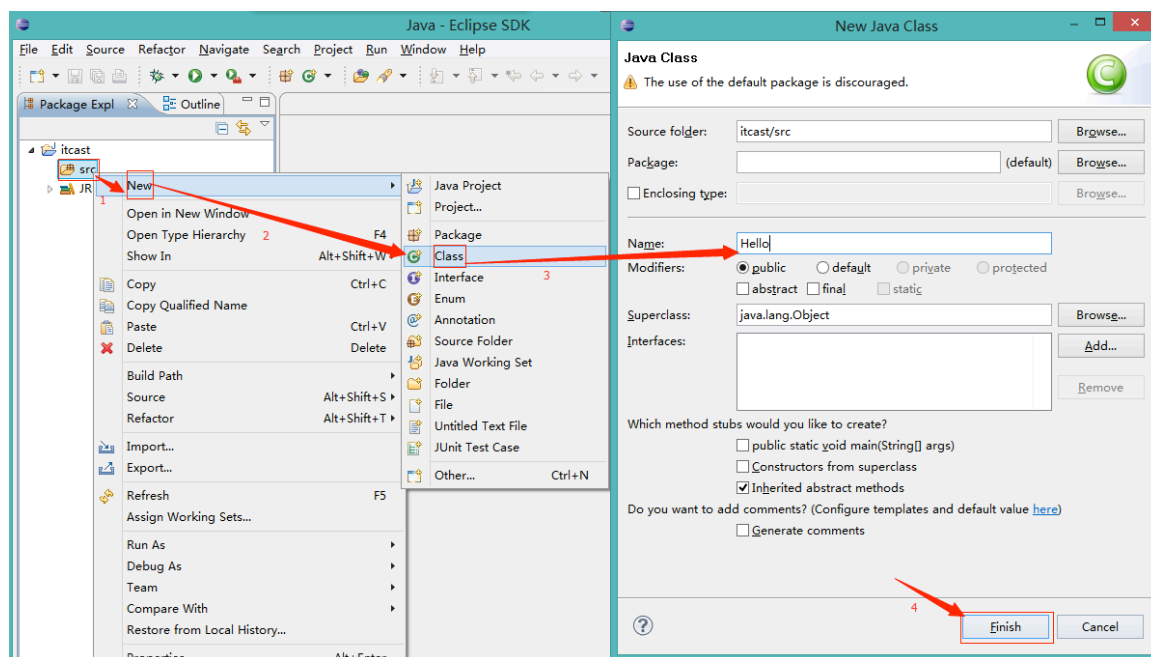


图1-3 创建类

- 4. 编译与执行
  - 编译：自动编译，当 java 代码保存的时候，自动 编译 class 文件
  - 执行：
    - ◆ 方式 1：点击菜单工具栏中的 绿色带有三角形的 run 按钮 运行
    - ◆ 方式 2：点击菜单栏中 Run， 点击 Run 运行 快捷键是 ctrl+F11
    - ◆ 方式 3：选中要运行的 java 文件，或者在编写代码的空白区域，右键选择 Run As --> 运行 java 程序

## 1.3 针对 Eclipse 空间的简单配置

- 程序的编译环境和运行环境
  - 编译环境：Window -- Preferences - Java - Compiler
  - 运行环境：Window -- Preferences - Java - Installed JREs

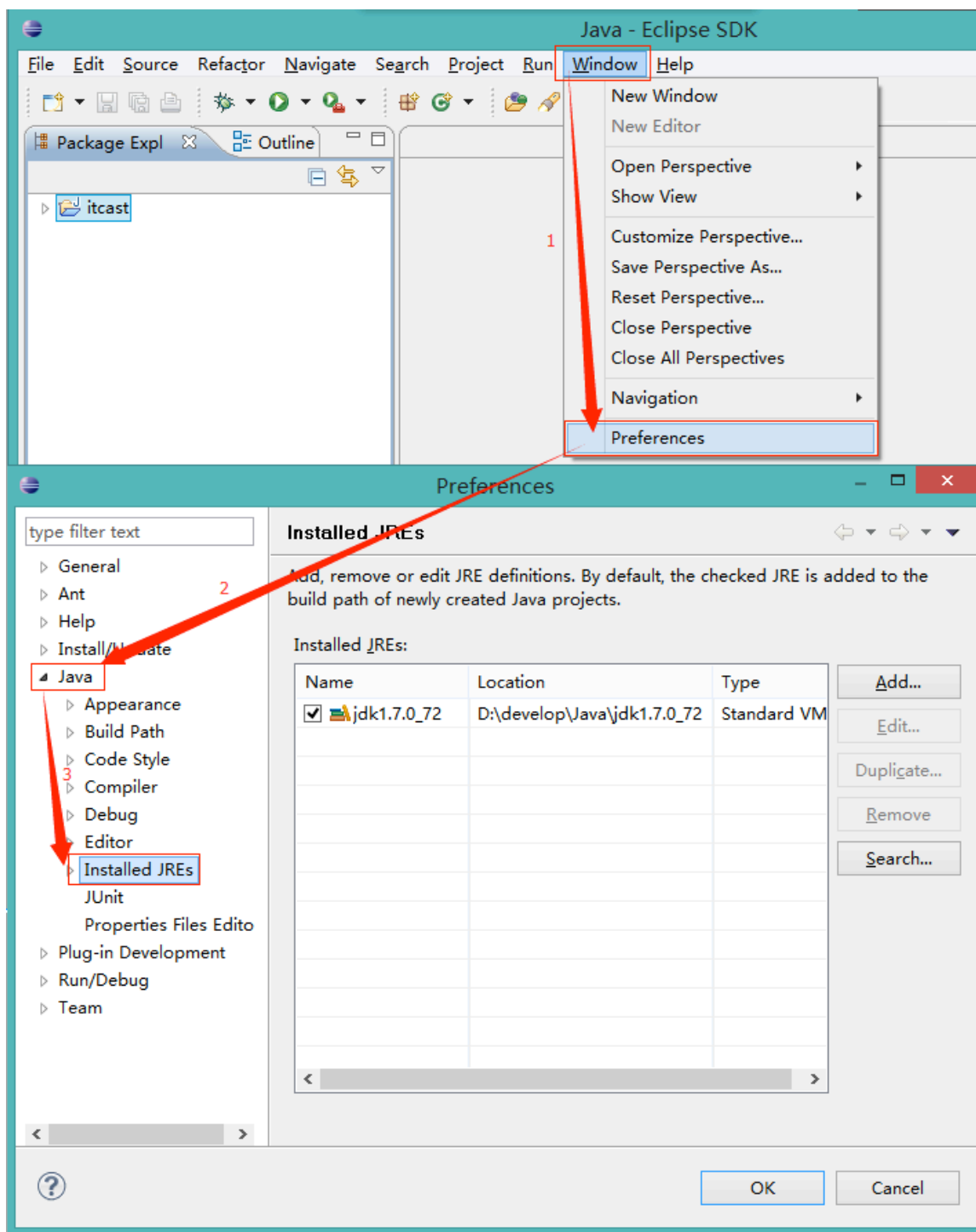


图1-4 编译环境与运行环境配置

- 如何显示行号与隐藏行号
  - 显示：在代码区域的左边空白区域，右键 -- Show Line Numbers
  - 隐藏：将上面的操作再做一遍

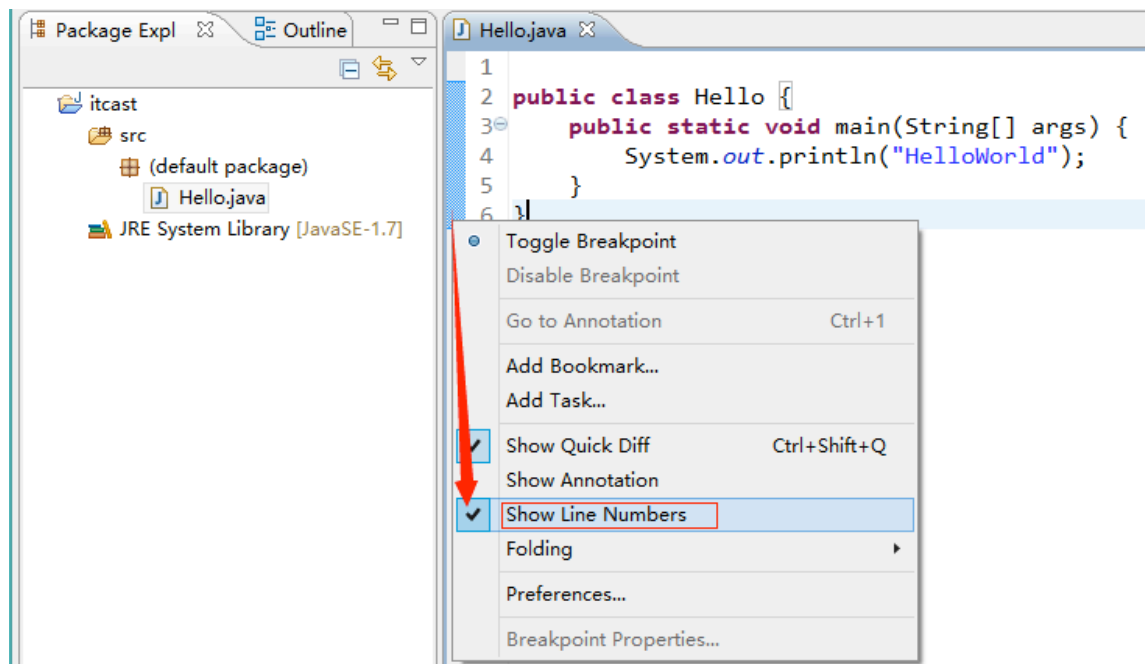


图1-5 行号显示与隐藏

- 如何更改字体大小与颜色
  - 修改 java 代码的字体大小与颜色
    - ◆ Window -- Preferences -- General -- Appearance -- Colors and Fonts -- Java -- Java Editor Text Font 选中后，点击 Edit，进行修改
  - 修改其他字体大小与颜色
    - ◆ Window -- Preferences -- General -- Appearance -- Colors and Fonts -- Basic -- Text Font 选中后 点击 Edit，进行修改

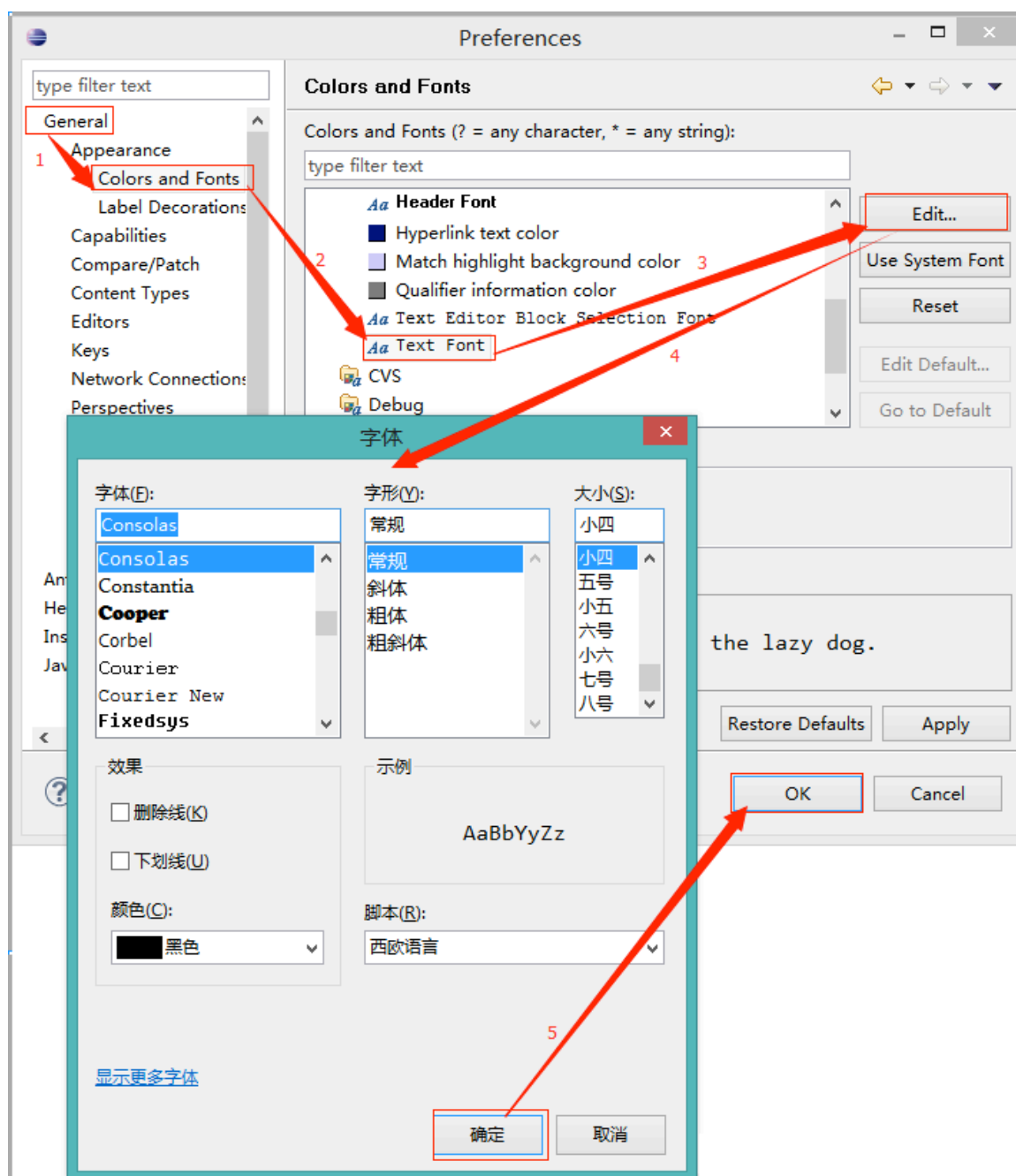
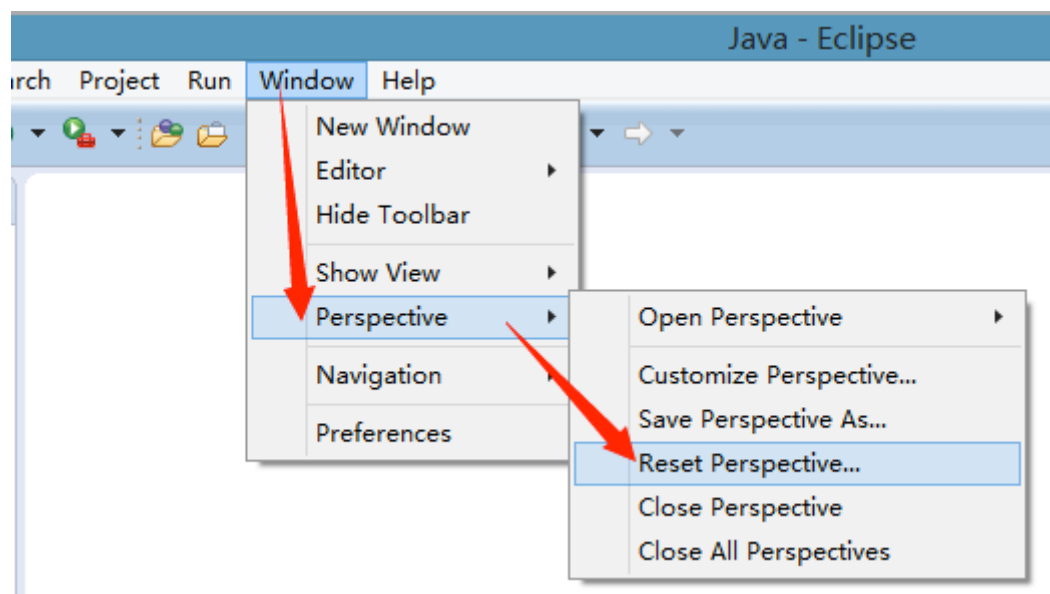
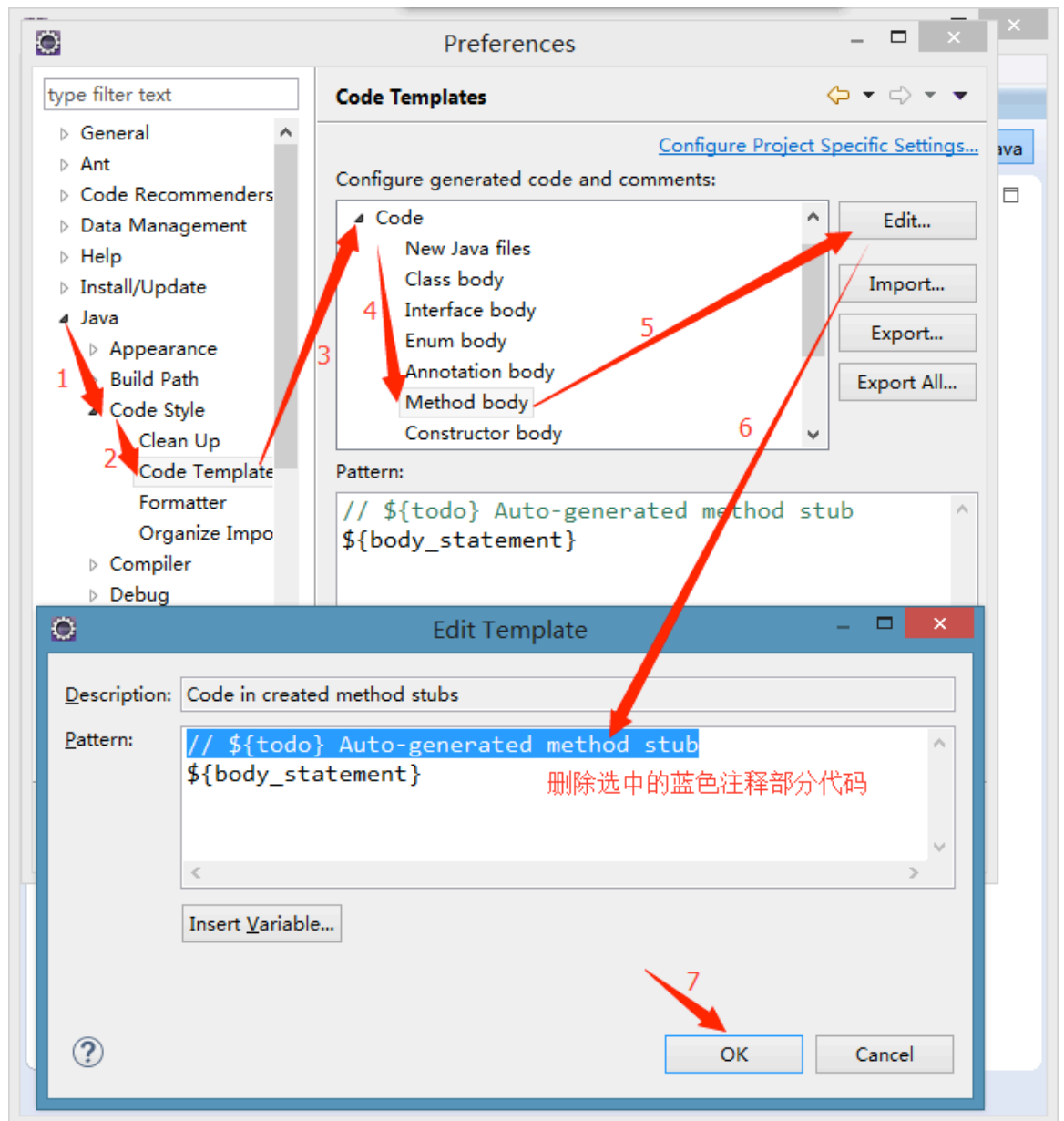


图1-6 设置字体

- 显示的窗口乱了，还原默认显示模式
  - Window - Perspective -- Reset Perspective

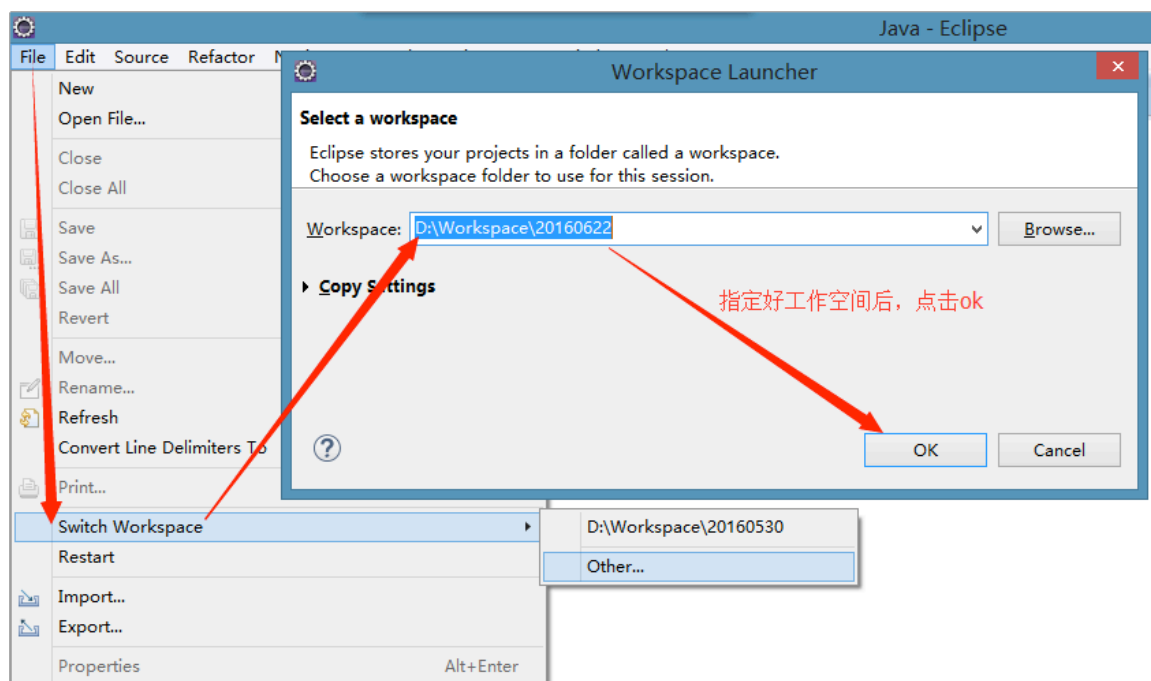


- 如何去掉默认注释
  - Window -- Preferences -- Java -- Code Style -- Code Templates -- Comments -- Methods, 点击 Edit , 将注释部分删除 (不建议删除)
  - Window -- Preferences -- Java -- Code Style -- Code Templates -- Code -- Method body, 点击 Edit , 将注释部分删除



- 切换工作空间
  - File - Switch Workspace - 指定工作空间 - ok





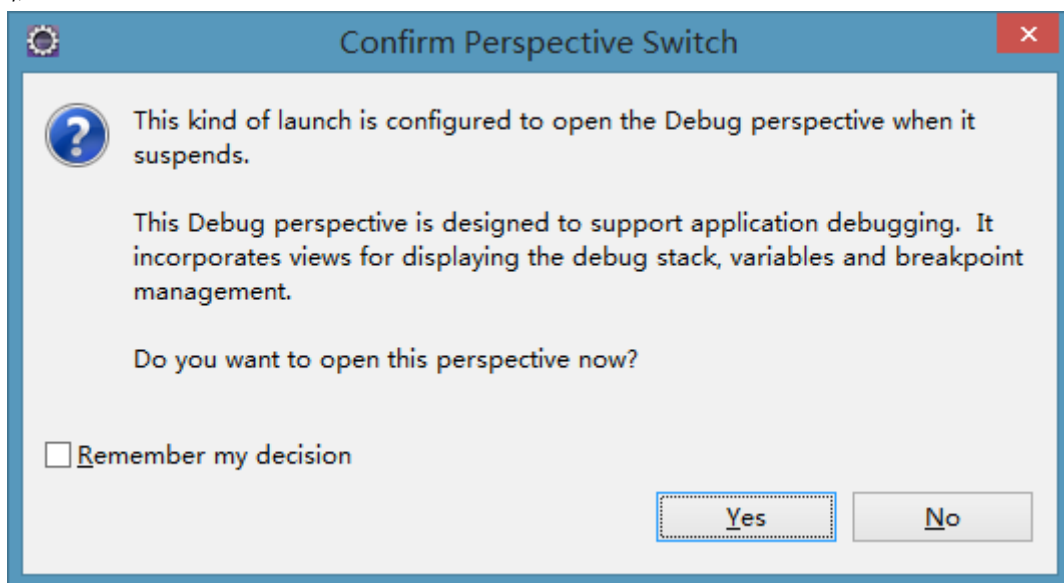
## 1.4 快捷键

- 内容辅助键 Alt+/
  - 自动补齐 main 方法 main 然后 Alt+/
  - 自动补齐输出语句 syso 然后 Alt+/
- 格式化 Ctrl+Shift+f
  - 代码区域右键 -- Source - Format
- 自动导包: Ctrl+Shift+o
  - 如果当前类在多个包中都存在, 这时候, 使用 Ctrl+shift+o, 进行选择一个包导入即可。
- 注释
  - 单行注释
    - ◆ 加注释: 先选中需要注释的内容, 然后 Ctrl+/
    - ◆ 取消注释: 先选中需要取消注释的内容, 然后 Ctrl+/
  - 多行注释
    - ◆ 加注释: 先选中需要注释的内容, 然后 Ctrl+Shift+/
    - ◆ 取消注释: 先选中需要取消注释的内容, 然后 Ctrl+Shift+\

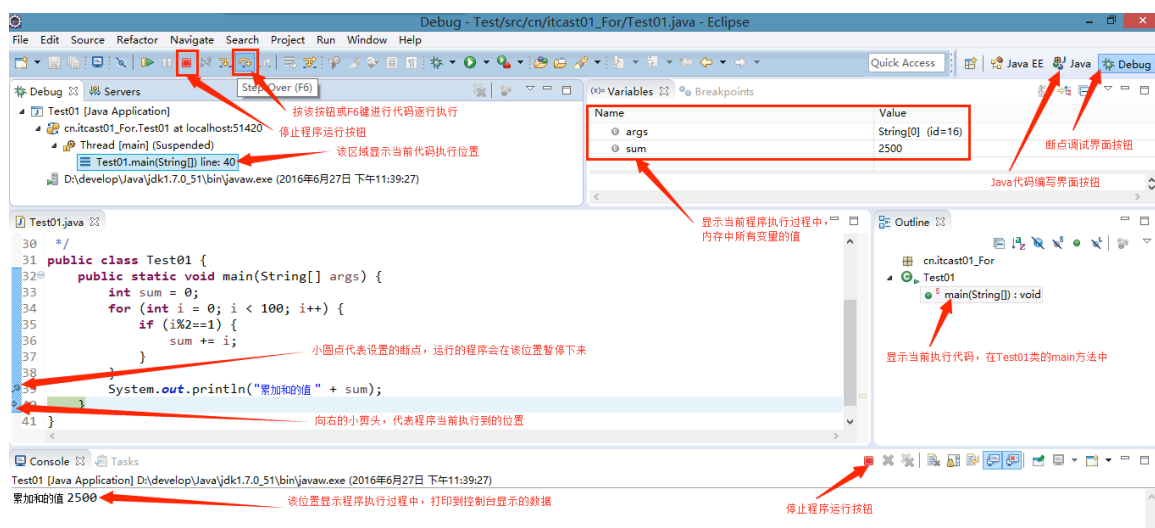
## 1.5 断点调试

- 断点: 就是一个标记
- 如何设置断点
  - 代码区域左边的空白位置 双击即可
- 在哪里加
  - 想看哪个语句的执行结果就在该语句前面加

- 怎么运行断点的程序
  - 代码区域右键 -- Debug as -- Java Application
  - 显示一个页面，提示是否进入调试页面？ 是否记住当前操作？ 是(Yes)是（勾选复选框）



- 弹出了断点页面



- 点哪里
  - ◆ 点击 工具栏中的 Step Over（执行下一行）
  - ◆ 快捷键 F6 看到每一步程序的执行过程
- 看哪里
  - ◆ 看代码区域： 对照看程序执行步骤
  - ◆ 看 Debug 区域： 对照看程序执行步骤
  - ◆ 看 variable 区域： 观察变量的创建、复制、销毁
- 如何消除断点
  - ◆ 把设置断点的方式再执行一遍
  - ◆ 切换到 Debug 视图，选择 Breakpoints -- Remove All Breakpoints 就是 2 个 XX 的图标

## 1.6 项目的删除

- 在项目区域删除
  - 在左侧 PackageExplorer 区域对指定项目右键，点击 delete 删除
- 在硬盘上删除
  - 在左侧 PackageExplorer 区域对指定项目右键，点击 delete 删除,同时选中下面的在硬盘上删除>Delete project contents on disk)

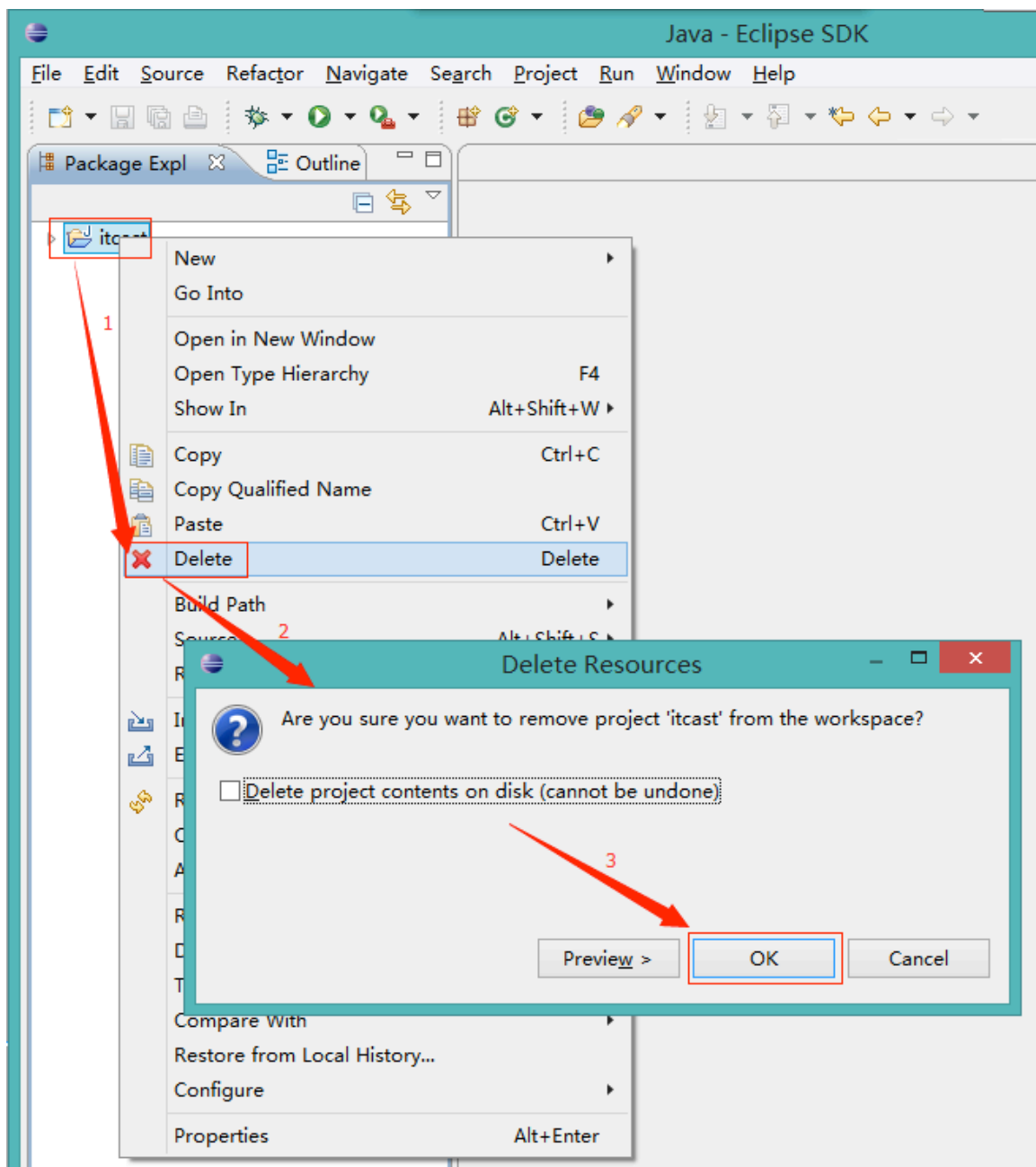


图1-7 项目删除

## 1.7 项目的导入

- 在左侧的 PackageExplorer 区域右键，选择 import
- 选择 General -- Existing Projects into Workspace
- 选择后，点击 next
- 在新的窗口中，点击 Browse 找到需要导入的项目（注意，找到项目的名字位置就可以了），选中要导入的项目，点击 Copy 复选框，点击 Finish 完成。
- 注意事项
  - Eclipse 中已经存在的同名项目不能再次导入

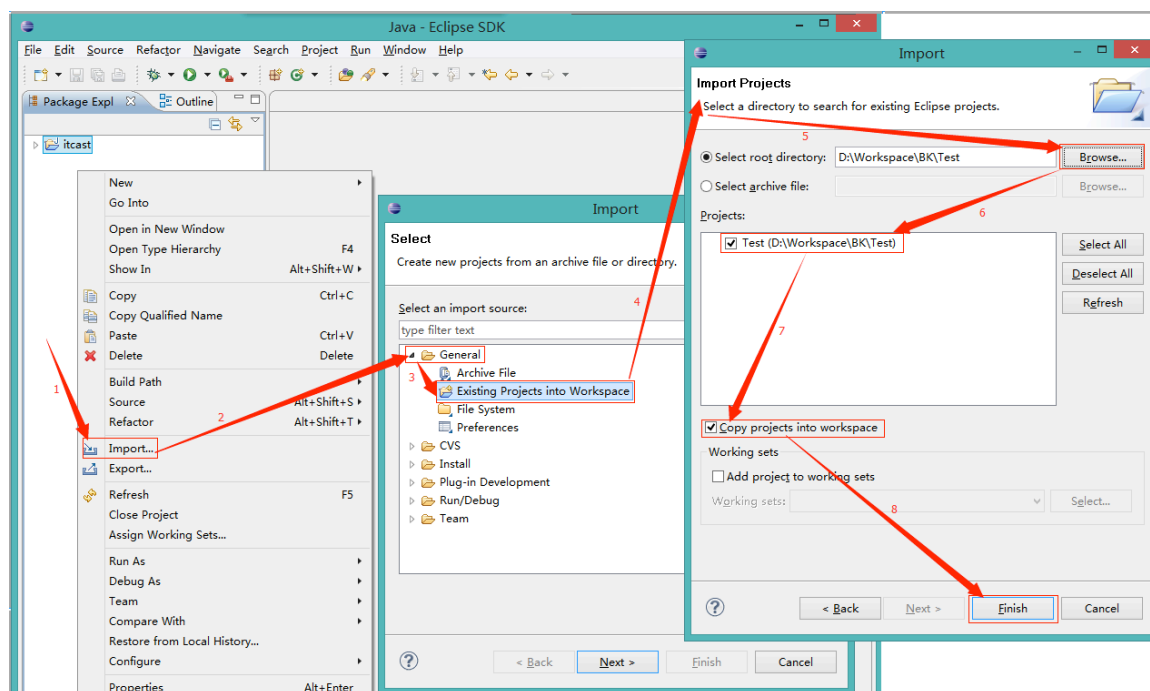


图1-8 项目导入

## 第2章 超市库存管理系统

### 2.1 案例介绍

模拟真实的库存管理逻辑，完成超市管理系统的日常功能实现，见下图

```
=====欢迎光临itcast超市=====
1:货物清单 2: 添加新货物 3:删除货物 4: 修改货物
5: 退出系统
请您输入要操作的功能序号:

-----
输入1: 货物清单
=====商品库存清单=====
商品编号    商品名称        商品单价
9001        少林寺酥饼核桃  120.0
9002        尚康杂粮牡丹饼  20.0
9003        新疆原产哈密瓜  7.0

输入2: 添加新货物
请输入新水果的名称:斯柯达苹果醋
请输入新水果的编号:9523
请输入新水果单价:19.9

输入3: 删除货物
请输入您要删除的水果编号:9523
水果信息删除完毕!

输入4: 修改货物
请输入您要修改信息的水果编号:9002
请输入新的水果ID:2002
请输入新的水果名称:; 味道好凤梨干
请输入新的水果单价:6.5
水果信息更新完毕!

输入5: 退出系统
```

## 2.2 案例需求分析

根据案例介绍，我们进行分析，首先需要有一个功能菜单，然后输入功能序号后，调用序号对应的功能方法，实现想要的操作。分析步骤如下

- 1. 完成超市商品初始化。创建商品，将商品添加到集合
- 2. 显示来到超市能做的操作，也就是显示主菜单
- 3. 根据接收到的功能选项，执行对应的功能
  - 3.1. 库存货物查询
  - 3.2. 添加新货物
  - 3.3. 删除货物
  - 3.4. 修改货物
  - 3.5. 退出系统, 结束 main 方法的运行
- 4. 循环，回到 2. 显示主菜单

## 2.3 实现代码步骤

每种库存商品都拥有多项商品信息，为了方便管理每种商品的信息，我们对商品信息进行封装，编写 FruitItem.java 文件

```
public class FruitItem {  
    int ID;           //商品编号  
    String name;       //商品名称  
    double price;      //商品单价  
    double number;     //商品数量  
    double money;       //商品金额  
}
```

上述代码中，对商品信息（编号、名称、单价、数量、金额）进行了封装。这样做的好处在于以后只要找到这个商品，就能够知道该商品的每项信息了。

编写 FruitStore.java，完成如下功能：

- 编写 main 主方法

```
public static void main(String[] args) {  
    //定义数组,记录每个商品信息  
    ArrayList<FruitItem> list = new ArrayList<FruitItem>();  
    /*  
    * 1.超市商品初始化  
    */  
    init(list);  
  
    while(true){  
        /*  
        * 2.显示主菜单  
        */  
        mainMenu();  
        /*  
        * 3.根据接收到的功能选项,执行对应的功能  
        */  
        chooseFunction(list);  
    }  
}
```

上述代码中，通过 while(true) 循环，完成用户调用不同功能。实现重复功能选择操作。

- 1. 超市商品初始化。创建商品，将商品添加到集合

```
public static void init(ArrayList<FruitItem> list) {  
    FruitItem item = new FruitItem();  
    item.name = "少林寺酥饼核桃";  
    item.ID = 9001;  
    item.price = 120;  
  
    FruitItem item2 = new FruitItem();
```

```
        item2.name = "尚康杂粮牡丹饼";
        item2.ID = 9002;
        item2.price = 20;

        FruitItem item3 = new FruitItem();
        item3.name = "新疆原产哈密瓜";
        item3.ID = 3;
        item3.price = 9007;

        list.add(item);
        list.add(item2);
        list.add(item3);
    }
```

上述代码中，完成商品集合的初始化操作，添加了 3 件商品。

- 2. 显示来到超市能做的操作，也就是显示主菜单

```
public static void mainMenu() {
    System.out.println("===== 欢迎 光 临  itcast  超 市  
=====");

    System.out.println("1:查询货物  2: 添加新货物  3:删除货物  4: 修改货物");
    System.out.println("5: 退出系统");
}
```

上述代码，用来完成用户操作界面的显示。

- 3. 根据接收到的功能选项，执行对应的功能

```
public static void chooseFunction(ArrayList<FruitItem> list) {
    System.out.println("请您输入要操作的功能序号: ");
    Scanner sc = new Scanner(System.in); //键盘输入对象
    String choose = sc.next();
    //根据接收到的选择，执行对应的功能
    switch (choose) {
        case "1": //3.1.库存货物查询
            showFruitList(list);
            break;
        case "2": //3.2.添加新货物
            addFruitItem(list);
            break;
        case "3": //3.3.删除货物
            delFruitItem(list);
            break;
        case "4": //3.4.修改货物
            updateFruitItem(list);
            break;
        case "5": //3.5.退出系统,退出 JVM
    }
```

```
        System.out.println("退出系统");
        return;
    default:
        System.out.println("对不起，没有您输入的功能，请重新选择");
        break;
    }
}
```

上述代码中，参数 `list` 为库存商品集合。当键盘输入功能序号后，通过 `switch` 语句匹配后，调用对应的方法完成对应功能的实现。

### ● 3.1. 库存货物查询

```
public static void showFruitList(ArrayList<FruitItem> list) {
    System.out.println("===== 商 品 库 存 清 单
=====");

    System.out.println("商品编号\t商品名称\t\t商品单价");
    //查询每种库存商品信息
    for (int i = 0; i < list.size(); i++) {
        FruitItem item = list.get(i);
        System.out.println(item.ID + "\t" + item.name + "\t" + item.price);
    }
}
```

上述代码中，用来查询所有库存商品信息。

### ● 3.2. 添加新货物

```
public static void addFruitItem(ArrayList<FruitItem> list) {

    //创建新获取对象
    FruitItem newItem = new FruitItem();
    Scanner sc = new Scanner(System.in); //键盘输入对象
    //提示输入信息
    System.out.print("请输入新水果的名称:");
    newItem.name = sc.next();
    System.out.print("请输入新水果的编号:");
    newItem.ID = sc.nextInt();
    System.out.print("请输入新水果单价:");
    newItem.price = sc.nextDouble();

    //向货物集合中添加新的物品项
    list.add(newItem);
}
```

上述代码中，通过键盘录入，将录入的商品信息存储到库存商品集合中。

### ● 3.3. 删除货物

```
public static void delFruitItem(ArrayList<FruitItem> list) {
```



```
System.out.print("请输入您要删除的水果编号:");
Scanner sc = new Scanner(System.in); // 键盘输入对象
int fruitID = sc.nextInt();
// 删除集合元素
for (int i = 0; i < list.size(); i++) {
    FruitItem thisItem = list.get(i);
    if (thisItem.ID == fruitID) {
        list.remove(thisItem);
        System.out.println("水果信息删除完毕!");
        return;
    }
}
System.out.println("对不起, 没有这个编号的水果!");
}
```

上述代码中, 通过键盘录入, 在库存集合中查找对应的商品编号, 如果找到该商品, 将该商品删除, 否则, 提示没有这个编号的商品。

#### ● 3.4. 修改货物

```
public static void updateFruitItem(ArrayList<FruitItem> list) {

    System.out.println();
    System.out.print("请输入您要修改信息的水果编号:");

    Scanner sc = new Scanner(System.in); // 键盘输入对象
    int fruitID = sc.nextInt();

    // 更新集合元素
    for (int i = 0; i < list.size(); i++) {

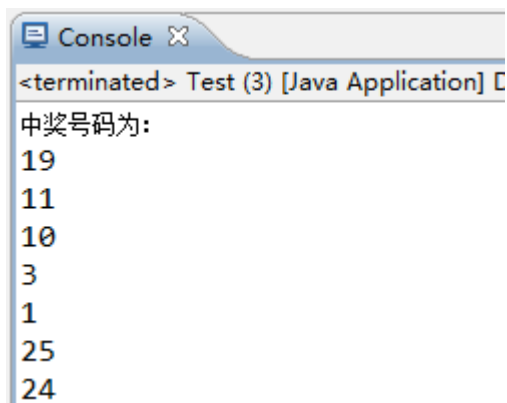
        FruitItem thisItem = list.get(i);
        if (thisItem.ID == fruitID) {
            System.out.print("请输入新的水果 ID:");
            thisItem.ID = sc.nextInt();
            System.out.print("请输入新的水果名称:");
            thisItem.name = sc.next();
            System.out.print("请输入新的水果单价:");
            thisItem.price = sc.nextDouble();
            System.out.println("水果信息更新完毕!");
            return;
        }
    }
    System.out.println("对不起, 没有这个编号的水果!");
}
```

上述代码中, 通过键盘录入, 在库存集合中查找对应的商品编号, 如果找到该商品, 则键盘录

入修改商品信息，否则，提示没有这个编号的商品。

## 第3章 趣味乐园

如下图所示，完成对应的双色球摇奖功能：



- 双色球摇奖
  - 随机摇出 1-33 之间的 7 个随机数，不能重复
  - 提示：
    - ◆ 存储双色球可使用 ArrayList 集合完成
    - ◆ 随机摇出双色球可使用 Random 类 nextInt() 方法、ArrayList 集合 remove() 方法完成