

基于 JAVA 的图形编辑程序的结构设计研究

徐超, 张珂, 江元

(安徽大学 电子科学与技术学院, 安徽 合肥 230039)

摘要:面向对象的代表语言 JAVA 日趋流行, 它的应用范围也越来越广泛。在此背景下, 文中针对 JAVA 在图形编辑领域中的应用从结构方面进行了探讨和研究, 以接口的灵活运用为手段实现了结构的优化, 并突出了接口在程序构筑中的“桥梁”作用。可以看出, 以对象为主题构建的主体结构和接口的适当穿插, 使整个图形编辑程序在实现了多种功能的前提下, 结构清晰, 简明高效, 并且可以在需要的时候很方便地对程序功能进行扩充。另外, 文中的例子也反映出基于 JAVA 的编程过程中类的继承和接口的实现两者间重要的互补性, 通过两者的结合, 往往可以使疑难杂症迎刃而解。

关键词:面向对象; 图形编辑; 接口

中图分类号: TP311

文献标识码: A

文章编号: 1673-629X(2009)07-0108-03

Research on Structure of Graphic Processing Program in JAVA

XU Chao, ZHANG Ke, JIANG Yuan

(Department of Electronic Science and Technology, Anhui University, Hefei 230039, China)

Abstract: The representative of the object oriented language JAVA becomes increasingly fashionable and its scope of application is used more and more widely. In this situation, the design of the structure of JAVA-based programming in graphic processing is being discussed and researched in the following article. The structure is optimized by flexible using of the interface and the “bridge” function of the interface is also exhibited in the usages of the project. It can be seen that building the main structure under the object as the theme and using the interface appropriately so that the whole graphic processing program, to achieve in a variety of functions under the premise of the structure, is clear, concise and efficient, and, when necessary, it can be very convenient for the procedure function expansion. In addition, also reflects the relation between class extending and interface overriding in JAVA-based programming, through the combination of the two, complicated problems can often be solved.

Key words: object oriented; graphic processing; interface

0 引言

面向对象技术代表了一种全新的程序设计思路。面向对象程序设计和问题求解力求符合人们日常自然的思维习惯, 尽量分解、降低问题的难度和复杂性, 从而提高整个求解过程的可检测性、可控制性和可维护性, 以此达到以较小代价和较高效率获得较满意效果之目的^[1]。作为当前一种广泛使用的面向对象编程语言, JAVA 具有多方面的特点。其中分布计算为程序应用到互连网提供了极大的便利; 稳定性和安全性成为了程序实用性的有效保证; 而跨平台平滑移植无疑是 JAVA 最诱人的特点^[2]。想开发一款容易在互联网上推广, 并且可以方便地移植到其他平台上的应用程序,

JAVA 显然是您的最优选择。当然, 这其中 JAVA 语言本身的其他优势特点也是原因之一。

在互联网上搜索图形编辑软件, 会随即出现五花八门的软件类别。但仔细一看, 这些软件并不是真正意义上的“图形编辑”, 而几乎都是侧重于图片、图像的编辑。例如其中热门的代表 Photo Shop, ACDSee 等等。同时, 真正以图形编辑为处理方式的软件几乎都侧重于较专业的设计, 例如 AutoCAD^[3]。在这种背景下, 完善而面向普通用户的图形编辑程序设计就成为了拥有巨大探索前景的研究工作。

1 图形编辑类程序的基本结构搭建思想

类是组成 JAVA 程序的基本要素。类是用来定义对象的模板, 它封装了一类对象的状态和方法, 并包含被创建对象的状态描述和方法的定义。定义类的最终目标是要使用这些类, 创建并使用某类的对象是使用该类的最主要方法^[4]。

收稿日期: 2008-10-16; 修回日期: 2008-12-24

基金项目: 安徽省自然科学基金(2005KJ0042D)

作者简介: 徐超(1962-), 男, 安徽亳州人, 教授, 硕士生导师, 研究方向为网络智能应用。

在开始创建图形编辑程序的图形类时,用一个抽象类定义为所有图形类的父类,在此类中定义所有图形具有的公共属性和公共方法名称。在继承了此类的各个具体的图形子类中给公共属性赋值并覆盖公共方法。在主程序中通过各图形子类的对象调用这些属性和方法完成图形的各种编辑^[5]。

如图 1 所示,以抽象类 ShapeInfo 作为图形父类,在其中定义公共属性 a、b,公共方法 draw(),各种具体的图形类,三角形、矩形等等类继承 ShapeInfo 类,根据特定的需要为 a、b 赋值,并用自己的实际绘制方法覆盖 draw()^[6]。

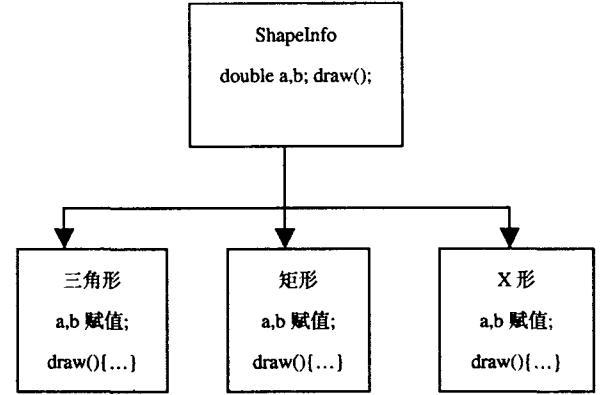


图 1 图形编辑程序基本结构

在主程序中创建各类的对象如下:

```
Shape triangle = new Triangle();
Shape rectangle = new Rectangle();
Shape x = new X();
```

通过创建的具体图形类的对象调用公共方法 draw() 实现具体的画法。

```
triangle.draw();
rectangle.draw();
x.draw();
```

2 具体图形编辑程序的结构设计

对于一般几何图形的编辑, JAVA 类库提供了备用的基本属性和基本方法,可以在需要时直接调用。在 JAVA 基本类库中,各具体的几何图形都继承自自己的抽象父类,这些父类又同时实现了 Shape 接口,所以,可以用默认类库实现简单的图形编辑^[7]。但对于一个实际的图形编辑程序,其复杂性远大于默认类库可以直接实现的功能,因此就需要对基本的结构进行调整和优化。

因为 JAVA 中不支持多重继承,所以无法在创建具体的图形子类时既继承此图形的抽象父类又选择性地继承包含需要的其他属性的类。但是如果把所需要的属性分别放在各个图形子类中定义,又无端地增加了

程序的重复性和繁冗性,因为这些属性基本都是所有图形所共有的。然而,在 JAVA 中,一个类虽然无法继承多个类,却可以同时实现多个接口。于是,可利用接口实现了程序的优化和整合。

如图 2 所示,把需要添加的图形类的公共方法 draw() 定义到一个接口 DrawInfo 中,并让所有的具体图形类实现这个接口。因为各图形类继承自库中的父类,所以已经不需要再重新定义如基本结构中叙述的各种边长、半径等变量,同时又覆盖了接口中的公共方法 draw(),因此已经建立了一个实际可操作的图形编辑程序的框架。

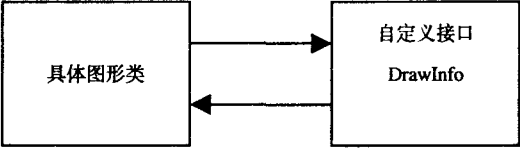


图 2 通过接口添加公共方法

为了实现真正的“画”,仅有框架是不够的,还需要获取鼠标的初始点和结束点等参数作为编辑图形的标准和依据。在实现编辑图形的过程中,需要在主程序中获取这两个参数,但如何传递到各图形封装类中就成为了一个问题。从结构上看,各图形封装类是作为一个包导入到主程序中的,包中的方法和参数可以通过具体的对象引用从而在主程序中得到体现,而包之外的参数是无法通过引用进而影响包中的任何情形的^[8]。

想解决这个问题无疑需要一个“桥梁”来作为主程序到图形封装类的数据传输通道。而从各图形封装类都实现了自定义接口这点,“桥梁”的任务很自然的落到了接口身上。在接口中定义获取和改变鼠标初始点 startPoint 和结束点 endPoint 的方法名称如下:

```
public void setStartPoint(Point startPoint);
public void setEndPoint(Point endPoint);
public Point getStartPoint();
public Point getEndPoint();
在各图形类中覆盖这些方法:
public void setStartPoint(Point startPoint) {
    this.startPoint = startPoint;
}
public void setEndPoint(Point endPoint) {
    this.endPoint = endPoint;
}
public Point getStartPoint() {
    return startPoint;
}
public Point getEndPoint() {
```

```
return endPoint;
```

```
}
```

在主程序中通过图形类对象调用这些方法,通过方法中的局部变量和方法体达到向图形类中的参数赋值的效果,通过获取的方法达到对鼠标位置点引用的效果。

3 图形多元化编辑程序的结构设计

实现了图形的简单编辑,还需要为编辑手法加上众多元素如颜色、填充与否等等满足不同环境下的需求。但各图形类无法再继承,接口中又无法直接定义公共属性变量,如何才能实现多元化属性与封装类的整合呢?

如图 3 所示,在一个自定义类 ShapeInfo 中定义需要用到公共属性变量:

```
public class ShapeInfo(){
    int 公共属性 1;
    boolean 公共属性 2;
    ...
}
```

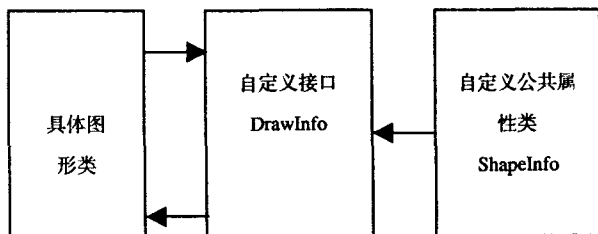


图 3 图形多元化编辑程序的基本结构

在自定义的接口 DrawInfo 中声明改变 ShapeInfo 中各属性的方法名称。

```
public interface DrawInfo(){
    public void setShapeInfo(ShapeInfo shapeInfo);
    ...
}
```

通过各具体图形类中 ShapeInfo 的对象实现对各个图形类公共属性的调用,通过主程序中各具体图形类的对象的方法实现对各个图形类公共属性的更改。

在主程序中,可以在最初只定义图形类的一般名称 Shape,根据具体的情况再把 Shape 实例化为具体图形类的对象,而在之后的实际编辑操作中,仍然以对象名 Shape 作为记号进行各种图形类属性和方法的调用。

需要注意的是,在实现接口的 setShapeInfo(ShapeInfo shapeInfo)方法时,需要通过公共属性类

ShapeInfo 的对象这个媒介形成 ShapeInfo 类中各属性变化的结果,而并非直接改变 ShapeInfo 类的对象本身。

在 Shape 被实例化时,可以把刚创建的对象存入一个数组,方便以后的取出处理,实现对已编辑的图形进行进一步的设置和更新^[9]。

4 结束语

运用 JAVA 面向对象的特点封装多种图形类并利用公共接口和程序启动类架起桥梁,可以使整个程序的结构清晰,效率提高:接口的灵活运用弥补了 JAVA 语言禁止多重继承的规定,提高了程序的灵活性;以对象为主题构建的整体结构可以使程序方便地加入新对象的处理语句,而以接口为连接手段实现的多种编辑功能也可以根据需要随时添加新的元素,从这一点上,程序的可扩展性得到反映。

互联网的普及和平台的多元化使 JAVA 语言的流行锐不可挡,而在 JAVA 的图形编辑功能本身并非鹤立鸡群的前提下,文中分析了基于 JAVA 的一般图形编辑程序的基本结构,并对较为复杂的编辑程序的结构优化进行了探讨,同时也以接口为切入点,讨论了实际编写程序时可以用到的一些构造技巧,由此揭示了 JAVA 语言在图形编辑领域存在的巨大潜力和众多可能性。

参考文献:

- [1] 黎明,杨先凤. JAVA 面向对象程序设计教程[M]. 成都:电子科技大学出版社,2005.
- [2] JAVA introduction[EB/OL]. 2000. <http://www.sdau.edu.cn/support/html/jindex.htm>.
- [3] 徐承志,张振东. 基于 OO 技术图形编辑器的设计[J]. 电脑学习,2004(6):25-26.
- [4] 王涛. ‘类’的思想在 Java Application 中的应用[J]. 计算机技术与发展,2006,16(8):4-5.
- [5] Eckel B. Thinking in JAVA(2nd)[M]. London: Prentice Hall, 2000.
- [6] 徐明浩. Thinking JAVA Programming with Mr. Seo[M]. 北京:人民邮电出版社,2005.
- [7] 杜冬,喻德军,唐旭冬. Java 2D 的图形图像处理技术[J]. 中国民航飞行学院学报,2004,15(4):35-38.
- [8] O'Reilly. Java Cook Book[EB/OL]. 2008. <http://download.csdn.net/download/209369/fan0316>.
- [9] 张红鹰,刘连忠. 基于 Java 的可扩展矢量图形编辑器设计与实现[J]. 计算机系统应用,2005(9):64-67.