

J A V A

系 列 丛 书



JAVA 语言

APPLET 编程技术

王克宏

主编

李京华 柳菁 蒋长浩 编著



清华大学出版社

496651

Java 系列丛书之四

Java 语言 Applet 编程技术

王克宏

主编

李京华 柳 菁 蒋长浩 编著



清华大学出版社

(京)新登字 158 号

38-021

内 容 简 介

本书介绍了如何使用 Java 语言编写 Applet 程序。内容包括 Applet 编程的基础,以及在 Applet 程序中应用较为广泛的绘图、图形用户界面、线程、多媒体和网络通讯等功能。书中提供了一系列 Applet 的小例子,使读者的学习更加方便。

本书适用于对 Java 语言编程的基础内容有一定了解的读者,可以作为 Java 语言的培训教材和程序开发人员的参考书。

版权所有,翻印必究。

本书封面贴有清华大学出版社激光防伪标签,无防伪标签者不得销售。



书 名: Java 语言 Applet 编程技术
作 者: 李京华 柳菁 蒋长浩
出 版 者: 清华大学出版社(北京清华大学校内,邮政编码:100084)
责任编辑: 徐培忠
印 刷 者: 北京市清华园胶印厂
发 行 者: 新华书店总店北京科技发行所
开 本: 787×1092 1/16 印张:9.75 字数:198 千字
版 次: 1997 年 6 月第 1 版 1997 年 6 月第 1 次印刷
书 号: ISBN 7-302-02485-5/TP·1256
印 数: 00001—10000
定 价: 15.00 元

序 言

近年来 Internet 以迅猛之势发展,已成为全世界规模最大的计算机网络,网上资源丰富,为世界各国千万用户所瞩目,我国也已于 1994 年 5 月正式入网,并已有成千上万使用者上网工作,如何充分、合理地利用 Internet 的网络与信息资源,为我国社会的各方面服务,是我们应予以重视和研究的问题。

Java 的出现正迎合了 Internet 发展形势的需要,它所体现的简单、不依附于平台、面向对象、分布式、可靠性、安全性、可移植性、动态性、多线程等特性,为 Internet 的使用提供了一种良好的开发和运行环境,成为 Internet 适用、新型的编程语言。Java 出现后立即为世界各大公司所注目,纷纷购买 Java 使用权,并在剧烈的市场竞争中展开了大规模的研究与产品开发。Java 语言及其相关技术同样也引起了我国学术界、产业界和应用界的广泛关注和兴趣。因此 Java 丛书的出版发行,是符合时宜之举,必将受到大家的欢迎。

根据编著者的计划,Java 丛书将包括:《Java 语言入门》,《Java 虚拟机规范》,《Java 语言编程技术》,《Java 语言 Applet 编程技术》,《Java 语言 API 类库》,《Java 软件包的使用》,《Java 语言 SQL 接口》,《Java 语言调试技术》及《HotJava 使用指南》等十多部书,达几百万字。详细深入地介绍 Java 语言及其相关技术并在短期内出齐,为我国广大读者研究和掌握 Java 提供了一种系统而全面的文献资料,无疑是一种十分有意义的事。Java 作为新生事物,尚在不断发展完善之中,因此我希望我们对 Java 及其相关技术不只停留于被动的学习、跟踪与使用,还应以积极主动的态度,通过应用实践和深入分析研究,参与开发创新,为计算机的网络应用做出我们自己的贡献。

清华大学计算机科学与技术系王克宏教授及其知识工程科研组,在 Internet 上进行多层次的研究开发工作已有一段时间,取得了阶段性成果,并曾在不久前召开的“中国计算机学会第九次全国学术大会”的全体会议上作报告,受到与会者的欢迎。相信他们在结合自己研究工作基础上编写的这套 Java 系列丛书,必将有助于我国信息科学技术的发展。我高兴地丛书作序并祝他们成功!

张淑祥
1996. 8

* 本文作者为中国科学院院士、中国计算机学会理事长

开辟 Java 计算的新时代

Sun 公司发明的 Java 语言风靡全球。据 Forrester Research 公司对《幸福》杂志评出的世界 1000 家大型企业进行的调查,目前在它们中间已有 62% 正在使用 Java 从事开发工作;有 42% 已将 Java 纳入自己企业一年内的战略开发计划。这个调查结果表明,Java 已为大家所接受。

计算机自从 50 年前诞生以来,经历了几个发展时代:终端/主机计算、PC 计算、客户机/服务器计算,而现在已发展到一个新的阶段,即 Java 计算时代。这个时代的特点就是以 Java 为代表的网络计算。

Java 带来的是一场革命。这是第一个真正独立于平台的计算方案,它能充分发挥 Internet 的作用。Java 计算是实现“一个世界,一个网络”构想的关键。这样一个透明的、全球连接的和信息交换的网络,可将所有最新的计算技术、电话、出版/媒体和娱乐集成一体。过去妨碍这一构想成为现实的是计算平台不能兼容,而 Java 语言却以其许多优秀特性使之成为允许各类系统相互兼容和共享应用环境的桥梁。这使各类软件“一次写成,到处可用(write once/run anywhere)”,这样,相同的软件可在不同器件上运行,无论是 PC 机、苹果机、UNIX 计算机,还是顶置盒、PDA(个人数据助理)、移动电话,乃至智能元器件,无一例外。Java 的这一贡献将使全人类受益,因而给人们的生活方式带来极大的改变。

清华大学王克宏教授是最早研究 Java 语言的中国学者之一。在他的周围,许多热心于 Java 计算的青年学者、博士生、硕士生和本科生组成了一支朝气蓬勃、积极进取的生力军,他们思想活跃、工作勤奋、通过努力把 Java 语言的开发和应用提高到一个新的水平。据我所知,在中国的诸多大专院校和科研院所,包括北京航空航天大学、上海复旦大学,也都有许多学者致力于 Java 的研究和开发。由于他们的辛勤努力和积极探求,在 Java 平台上开发和使用的 Java 应用将给中国的信息业带来新的发展,使 Java 计算在中国进一步得到推广。

《Java 系列丛书》就是 Java 开发园中的一朵奇葩,它不倜傥,以其朴实无华、实用有效的风格展示于人。我们希望由王克宏教授主编的该套丛书能赢得广大读者的喜爱,希望能有更多的中国智者和用户在 Java 计算的时代熟悉 Java,掌握 Java,利用 Java 的诸多优点,编写出更多更美的应用软件,从而造福于人类。

谨以此愿祝贺 Java 系列丛书的出版发行。

李永超

1996. 10

* 本文作者为 Sun 公司中国/香港地区副总裁

前 言

自从于 1995 年问世以来,Java 语言以其独特的优势迅速风靡了计算机界,并且逐步向各个应用领域发展,掀起了计算机软件的新浪潮。编写在主页上运行的 Applet 程序是 Java 语言的重要应用途径之一。本书作为 Java 系列丛书的第四册,介绍的就是使用 Java 语言编写 Applet 程序的各种技术。

本书各章的题目和内容简介如下:

第 1 章:Applet 入门。介绍了 Applet 的基础知识及其与 HTML 语言配合使用的方法。

第 2 章:绘图。介绍了使用 AWT 类库绘制简单的图形和文本的方法。

第 3 章:图形用户界面。介绍了使用 AWT 类库制作各种图形用户界面的技术。

第 4 章:线程。介绍了如何使用多线程技术增强程序功能。

第 5 章:多媒体。介绍了处理图象,显示动画和播放声音文件的方法。

第 6 章:网络。介绍了 Applet 如何与其所在的主机和其他主机进行通讯。

本书的第 1 章由蒋长浩编写,第 2 章的主要部分、第 4 章和第 5 章由柳菁编写,第 2 章的一部分、第 3 章和第 6 章由李京华编写。李京华还负责全书的校对工作。

由于 Java 语言还是一种新兴的计算机语言,我们对其学习和研究也很有限,在编写的过程中难免出现各种错误,希望广大读者谅解并予以批评指正。

编 者

目 录

第 1 章 Applet 入门	1
1.1 Applet 基础和相关 HTML 简介	1
1.2 Applet 安全基础	4
1.3 Applet 的生命周期和四种基本方法	5
1.4 将 Application 转换为 Applet	6
1.5 向 Applet 传递参数	11
1.6 Applet 与环境的联系	13
1.7 既是 Applet 又是 Application	16
第 2 章 绘图	20
2.1 基本绘图方法	20
2.2 绘制几何图形	21
2.3 填充几何图形	23
2.4 文本	25
2.5 字体	26
2.6 字模	28
2.7 颜色	30
2.8 绘图模式	31
2.9 图形实例	33
2.10 动画图形	34
第 3 章 图形用户界面	37
3.1 编程基础	37
3.2 常用组件简介	42
3.3 布局方式	62
3.4 多窗口编程	71
3.5 事件处理	75
3.6 编程实例	80
第 4 章 线程	87
4.1 引入线程	87
4.2 深入掌握线程	92
4.3 线程通信	99

第 5 章 多媒体	106
5.1 图象	106
5.2 动画	116
5.3 声音	120
5.4 编程实例	121
第 6 章 网络	128
6.1 服务器程序	128
6.2 客户程序	131
6.3 多客户的服务器	135
6.4 客户和其他主机之间的通讯	137
6.5 CGI 编程简介	142

第 1 章 Applet 入门

1.1 Applet 基础和相关 HTML 简介

我们写这一部分时,假设读者已经对 Java 语言的基本要素及 HTML (HyperText Markup Language)有些初步的了解(刚刚接触 Java 语言的读者,建议先从本系列丛书的第一册《Java 语言入门》开始读起)。有了这些基础的读者将发现原来用 Java 语言编写 Applet 是这样的简单和充满乐趣。本章中有一些绘图(包括显示文本)和图形用户界面的内容,你如果现在对它们不了解也不要紧,因为这并不太影响本章的学习,在第 2 章和第 3 章中将介绍这些内容。下面就让我们开始轻松美妙的 Applet 旅游吧。

1.1.1 Applet 基础

我们知道 Java 程序可分为 Application(应用程序)和 Applet(小应用程序)。两者的区别在于 Application 和其他高级语言编写的程序没有什么本质的区别,虽然它通过 Java 解释器解释执行;而 Applet 是一种特殊的 Java 程序,它通常通过支持 Java 的网页浏览器下载后执行。当然读者也可以通过像 JDK (Java Development Kits)这样的 Java 开发工具中的 appletviewer 工具浏览 Applet 的效果。要观看 Applet 的效果离不开一个必要的 HTML 文件。这个 HTML 文件中关于 Applet 的信息至少应包含以下三点:

1. class 文件(编译后的 java 文件)的名字;
2. class 文件的地址;
3. 如何在网页上显示 Applet。

然后浏览器就可以执行网络远端或者是本地机上的 Applet 了。

HTML 文件除了包含有关 Applet 的信息以外,还可以包含像普通 HTML 文件包含的那些多种字体、列表、链接等元素。增加 Applet 只是使网页更加富有生气,如添加声音、动画等这些最吸引人的特征,它并不会改变 HTML 文件中与 Applet 无关的元素。

下面通过一个非常简单的例子来说明 Applet 是怎样被执行的。

例 1-1

```
import java.awt. * ;
import java.applet. * ;

public class DemoOfAppletRunning extends Applet
{
    public void paint (Graphics g)
    {
        g.drawString("This is a demo of Applet running in the browser!",60,20);
    }
}
```

```
}  
}
```

为运行这个 Applet, 须要在 HTML 文件中加入以下两行:

```
<APPLET CODE="DemoOfAppletRunning.class" WIDTH=100 HEIGHT=100>  
</APPLET>
```

完整的 HTML 文件如下:

```
<HTML>  
<HEAD>  
<TITLE>This is a demo of Applet</TITLE>  
</HEAD>  
<BODY>  
<APPLET CODE="DemoOfAppletRunning.class" WIDTH=100 HEIGHT=100>  
</APPLET>  
</BODY>  
</HTML>
```

把这个文件存为 Demo.html。

图 1.1 是 Applet 运行的效果。

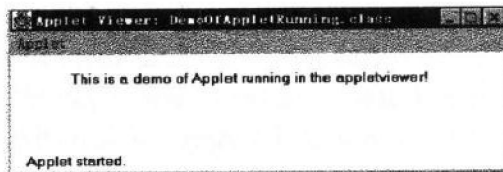


图 1-1

可以通过 JDK 中的 appletviewer 或者 Netscape 等支持 Java 的浏览器来观看。

1. 通过 appletviewer, 直接在命令行中键入 appletviewer Demo.html 就可以了。

2. 通过浏览器, 可以从 File|open 中打开 Demo.html。

须要补充的是, 上述两种方法各有其优缺点:

1. appletviewer 是一个弹出式的快速工具, 但它并不能显示 HTML 文件中与 Applet 无关的文字或者图片等, 它只能显示 HTML 文件中包含的 Applet 的运行情况。如果一个 HTML 文件中包含多个 Applet, appletviewer 将弹出多个窗口以对应各个 Applet。

2. 支持 Java 的浏览器一般能够显示包括 Applet 在内的各种元素, 包括文字、图片、链接等。但是当浏览器运行 Applet 的同时, 它会将一些限制 Applet 下载的安全机制加给被执行的 Applet。这样用 appletviewer 看到的 Applet 的运行情况也许会 and 用浏览器看到的有所不同。

3. 在调试 Applet 的时候须要多次调用浏览工具。在用浏览器时, 当对 Applet 的代码进行修改并编译, 再重新载入时浏览器显示出来的 Applet 仍然是修改前的模样, 必须退出浏览器再重新进入后, 这时打开修改后的程序才能看到新的 Applet。而使用 appletviewer, 可以及时地观看到修改后的最新效果。

1.1.2 HTML 的 Applet 标签(tag)

上面已经提到了运行 Applet 必须有一个 HTML 文件伴随。在这个文件里,有关 Applet 的信息将被传递给浏览器,或者能够运行 Applet 的工具。在 HTML 中有一些基本的标签与 Applet 相联系,现在逐一加以介绍。

上文曾提到 Applet 标签在 HTML 中最基本的形式为:

```
<APPLET CODE="DemoOfAppletRunning.class" WIDTH=100 HEIGHT=100>
</APPLET>
```

其中<APPLET>和</APPLET>标明 Applet 标签的起始和终止。CODE 指明了 class 文件的名字,WIDTH 和 HEIGHT 指明 Applet 所占窗口部分的大小。

下面是与 Applet 有关的一些 HTML 标签和属性:

- WIDTH,HEIGHT

这两个属性是必不可少的,它们分别指明 Applet 所占窗口部分的宽度和高度。它们的值是以像素为单位的。

- ALIGN

这个属性是用来指明 Applet 的对齐方式的,见表 1.1。

表 1.1

属性值	作用
LEFT	将 Applet 置于页面的左侧,相邻的文本紧跟在 Applet 的右边
RIGHT	将 Applet 置于页面的右侧,相邻的文本紧靠在 Applet 的左边
BOTTOM	Applet 的底边与同一行内的文本的底边相对齐
TOP	将 Applet 的顶端与同一行内的顶端相对齐
TEXTTOP	将 Applet 的顶端与同一行内的文本的顶端相对齐
MIDDLE	将 Applet 的中线与同一行的底线相对齐
ABSMIDDLE	将 Applet 的中线与同一行的中线相对齐
BASELINE	将 Applet 的底线与同一行的底线相对齐
ABSBOTTOM	将 Applet 的底线与同一行的底边相对齐

图 1.2 是 ALIGN 属性的具体实例。

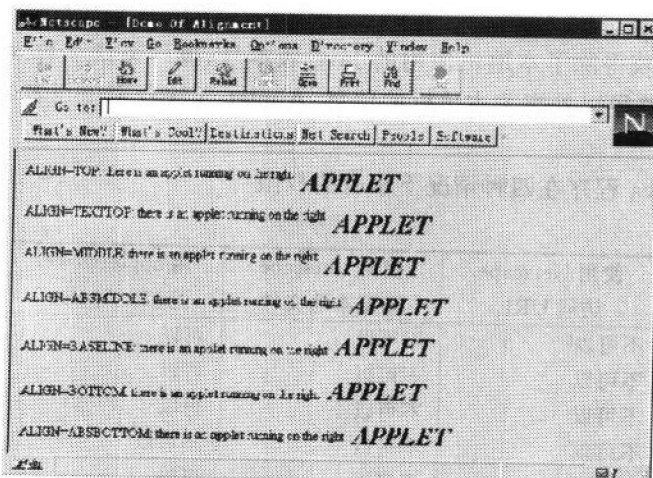


图 1-2

- VSPACE,HSPACE

这个属性指明在 Applet 上下(VSPACE)和两边(HSPACE)分别留有多大的空间。

还有四个属性是与 Applet 代码密切相关的。它们是：

- CODE

这个属性是指明 Applet 编译后产生的 class 的文件名。可以写明它的相对路径名(参考 CODEBASE)或者绝对路径名,也可以是网络 URL。

- CODEBASE

这个属性是告诉浏览器 Applet 的 class 文件所在目录。有了这个地址,浏览器将根据它来查找 CODE 属性中指明的 class 文件。比如在 JavaApplet 目录里有一个 AppletDemo.class 文件,则可以这样表示：

```
<APPLET CODE = " AppletDemo.class" CODEBASE = " JavaApplet" WIDTH = 150  
HEIGHT=200>  
</APPLET>
```

- NAME

这个属性很少用到。它是用在当你想在同一个主页的两个 Applet 之间建立联系时使用的,用来标识区分两个 Applet。必须调用 AppletContext 类里的 getApplet()方法来获得 Applet 的名字属性值。

有关 Applet 的另一个十分有用的标签 PARAM,将在后面通过例子向读者介绍。

1.2 Applet 安全基础

由于 Applet 从远程机上下载而在本地机上运行的特殊性,安全问题显得十分重要。为此,Applet 在运行时要受到更多的限制,诸如：

1. Applet 永远无法运行本地机上的可执行程序。

2. Applet 除了与所在的服务器联系以外,无法再同任何其他的服务器取得联系。

3. Applet 无法对本地机上的文件系统进行读写操作。(目前 Netscape 是这样做的,以后也许会有其他的浏览器减少这方面的限制,因为 Sun 公司并没有把这作为 Java 规范的一部分。)

4. Applet 无法获得除了本地机使用的 Java 版本号、操作系统名称及版本号、文件名分隔符(‘/’或‘\’)和路径以外有关本地机的信息。Applet 无法获得使用者的名字和 E-mail 地址等。

表 1.2 说明 Java 程序在四种情况下享有的权限

表 1.2

操 作	使用 Netscape 访问 URL	使用 Netscape 浏览本地文件	用 appletviewer 运行 Applet	本地机运行 Application 程序
读取本地机文件	不可以	不可以	可以	可以
修改本地机文件	不可以	不可以	可以	可以
获得文件信息	不可以	不可以	可以	可以
删除文件	不可以	不可以	不可以	可以
运行其他程序	不可以	不可以	可以	可以

续表

操 作	使用 Netscape 访问 URL	使用 Netscape 浏览本地文件	用 appletviewer 运行 Applet	本地机运行 Application 程序
读取用户名属性	不可以	可以	可以	可以
与服务器连接	可以	可以	可以	可以
与其他主机连接	不可以	可以	可以	可以
安装 Java 程序库	不可以	可以	可以	可以
调用 exit 方法	不可以	不可以	可以	可以
创建弹出式窗口	可以	可以	可以	可以

1.3 Applet 的生命周期和四种基本方法

如果你想编写实用的 Applet 程序,那么对 Applet 类中的四种基本方法: `init()`、`start()`、`stop()`、`destroy()` 一定要熟悉。下面介绍如何实现这些基本方法,以及它们的具体工作情况。

1.3.1 `init()` 方法

这个方法主要是为 Applet 的正常运行做一些初始化工作,就像建筑大厦要打地基一样。当一个 Applet 被系统调用时,系统首先执行的就是该方法内的代码。通常可以在该方法中完成从主页向 Applet 传递参数,添加用户界面的基本组件等。

1.3.2 `start()` 方法

系统在调用完 `init()` 方法之后,将自动调用 `start()` 方法。而且,每当用户离开包含该 Applet 的主页后又再返回时,系统又会再执行一遍 `start()` 方法。这就意味着 `start()` 方法可以被多次执行,而不像 `init()` 方法。因此,可把只希望执行一遍的代码放在 `init()` 方法中。可以在 `start()` 方法中开始一个线程,如继续一个动画、声音等等,后面将具体介绍。如果你的 Applet 中没有什么工作希望在用户离开该主页时暂停,那就不须要实现该方法。

1.3.3 `stop()` 方法

这个方法在用户离开 Applet 所在页面时执行。因此,它也是可以被多次执行的。它使你可以在用户并不注意 Applet 的时候,停止一些耗用系统资源的工作以免影响系统的运行速度,且并不需要人为地去调用该方法。如果你的 Applet 中不包含动画、声音等程序时通常也不必实现该方法。

1.3.4 `destroy()` 方法

与对象的 `finalize()` 方法不同,Java 在浏览器关闭的时候才调用该方法。Applet 是嵌在 HTML 文件中的,所以不必关心何时 Applet 被关闭,它在浏览器关闭的时候自动执行。在 `destroy()` 方法中一般可以要求收回占用的非内存独立资源。(如果在 Applet 仍在运行时浏览器被关闭,系统将先执行 `stop()` 方法,再执行 `destroy()` 方法。)

1.4 将 Application 转换为 Applet

将一个图形界面的 Java 应用程序转换成能够嵌入 HTML 主页且在浏览器中运行的 Applet 是很容易的。图 1-3 表示几个类的继承关系。

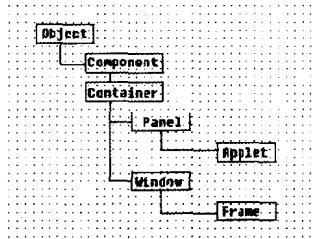


图 1-3

其中 Applet 类和 Frame 类都是 Container 类的子类。因此,可通过如下语句往用户界面中添加组件。

```
add(new Button("Perform"));
```

另外有几点需要特别注意的地方:首先是缺省的 Frame 类和 Applet 类的外观管理器(Layout Manager)是不同的。在 Application 的 Frame 中缺省使用的外观管理器是 BorderLayout,而 Applet 中缺省使用的外观管理器是 FlowLayout。这就意味着,如果没有在应用程序中设定外观管理器,则在将应用程序转换成 Applet 的过程中就须要将缺省的 FlowLayout 外观管理器改为 BorderLayout 外观管理器。

下面是将 Application 程序转换成 Applet 程序的几个基本步骤:

1. 创建一个包含 APPLET 标签的 HTML 文件。
2. 去掉应用程序中的 main() 方法。通常 main() 方法中包含生成一个新窗体对象的代码,而这在 Applet 当中系统在生成一个 APPLET 标签的同时是自动完成的。main() 方法中通常设定窗体的大小,而这也可以由 APPLET 标签中的 WIDTH 和 HEIGHT 属性完成。
3. 类名继承 Applet 类,而不是 Frame 类。
4. 用 init() 方法代替 Application 程序中的构造方法。当浏览器创建 Applet 类对象的时候,它自动执行 init() 方法。
5. 如果应用程序中缺省使用了 BorderLayout 外观管理器,按如下方法将它在 init() 方法中重新设定:

```
public ClassName extends Applet
{
    public void init()
    {
        setLayout(new BorderLayout());
        ...
    }
}
```

6. 如果 Application 程序中有 setTitle() 方法, 必须将其去掉。因为 Applet 程序中没有标题栏。(当然可以在 HTML 文件中的 TITLE 标签中设定标题。)如果应用程序中使用了菜单, 就要在小应用程序中用按钮(Button)来替换。因为 Applet 中无法直接使用菜单。

下面通过一个程序来具体说明转换过程。

这是一个计算从 1 到任一个指定的正整数累加求和的程序。(当然运算结果不能超过 Java 语言中支持的最大整数)

例 1-2

```
import java.awt.* ;

public class CalculateSum extends Frame
{
    Button btnExit=new Button("Exit");
    Button btnRun= new Button("Run");
    Label lblIntro =new Label ("Please input positive integer here:");
    Label lblResult =new Label ("The result is :");
    TextField txtIntro = new TextField (" 0");
    TextField txtResult =new TextField (" 0");
    Panel p1=new Panel ();
    Panel p2=new Panel ();
    Panel p3=new Panel ();

    public CalculateSum()
    { resize(350,200);

        p1.add(lblIntro);
        p1.add(txtIntro);

        p2.add(lblResult);
        p2.add(txtResult);

        p3.add(btnRun);
        p3.add(btnExit);

        add("North",p1);
        add("Center",p2);
        add("South",p3);
        setTitle("This is a Calculator !");
    }

    public boolean action(Event evt,Object obj)
    {
        if(evt.target instanceof Button)
```

```

        {
            String btnName = (String) obj;
            if (btnName.equals("Exit"))
                System.exit(0);
            else if (btnName.equals("Run"))
            {
                int sum = 0;
                for (int i = 1; i <= Integer.parseInt(txtIntro.getText().trim()); i++)
                    sum = sum + i;
                txtResult.setText(String.valueOf(sum));
            }
        }
    }

    return super.action(evt, obj);
}

public static void main(String arg[])
{
    (new CalculateSum()).show();
}
}

```

其运行效果如图 1-4 所示：

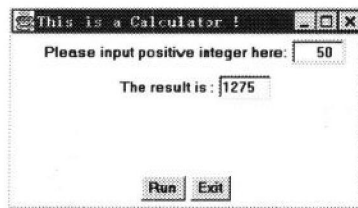


图 1-4

接着,按照前面所介绍的步骤将其转换成 Applet 程序。

第 1 步,创建一个包含 APPLET 标签的 HTML 文件。代码如下：

```

<HTML>
<TITLE>Calculate Sum</TITLE>
<BODY>
<APPLET CODE="CalculateSum.class" width=320 height=300>
</APPLET>
</BODY>
</HTML>

```

第 2 步,去掉应用程序中的 main() 方法。

第 3 步,将类名继承 Applet 类。即将代码中的 public class CalculateSum extends Frame 改为

```
public class CalcualteSum extends Applet
```

注意在程序开头要加入 import java.applet.Applet;

第4步,用 `init()` 方法来代替应用程序的构造方法,并在其中修改外观管理器为 `BorderLayout`。

原代码中构造方法如下:

```
public CalculateSum( )
{
    resize(350,200);

    p1.add(lblIntro);
    p1.add(txtIntro);

    p2.add(lblResult);
    p2.add(txtResult);

    p3.add(btnRun);
    p3.add(btnExit);
    add("North",p1);
    add("Center",p2);
    add("South",p3);
    setTitle("This is a Calculator !");
}
```

用 `init()` 方法来代替它:

```
public void init( )
{
    resize(350,200);
    setLayout(new BorderLayout());

    p1.add(lblIntro);
    p1.add(txtIntro);

    p2.add(lblResult);
    p2.add(txtResult);

    p3.add(btnRun);
    p3.add(btnExit);

    add("North",p1);
    add("Center",p2);
    add("South",p3);
    setTitle("This is a Calculator !");
}
```

第5步,去掉构造方法中的 `setTitle("This is a Calculator !");`;

至此,将 `Application` 程序转换成 `Applet` 程序的过程就完成了。转换后的完整代码如下: