



北京大学出版社 正宗Java丛书

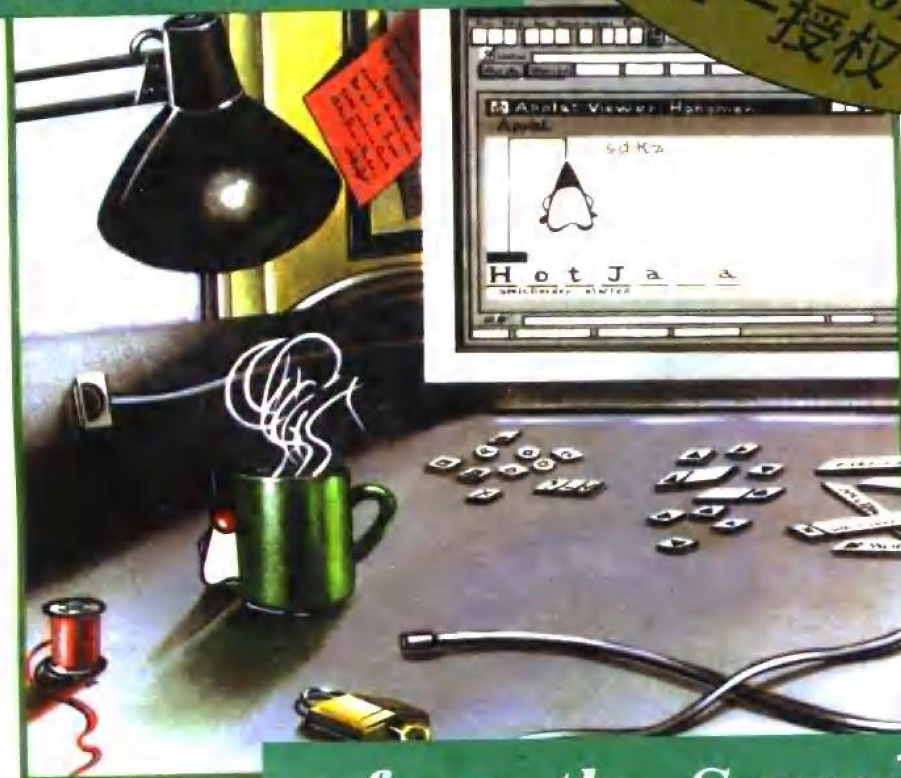
Java类库手册

〔美〕 Patrick Chan, Rosanna Lee 著

玄伟剑 等译

The Java Series

Java Soft
唯一授权



...from the Source™



北京大学出版社
艾迪生维斯理出版有限公司

著作权合同登记 图字:01-97-1092 号

图书在版编目(CIP)数据

Java 类库手册/[美]钱(Chan,P.), (美), 李(Lee,R.)著. —北京:北京大学出版社, 1997. 8

ISBN 7-301-03476-8

I. J… II. ① 钱… ② 李… III. Java 语言-软件工具-手册 IV. TP311.56-62

中国版本图书馆 CIP 数据核字 (97) 第 14169 号

本书原版英文版由 Addison Wesley Longman, Inc. 出版, 版权归该公司所有(Copyright ©1997 by Addison Wesley Longman, Inc.).

本书由 Addison Wesley Longman, Inc. 授权北京大学出版社在中国出版发行。未经出版者书面允许, 不得以任何形式复制或抄袭本书内容。

版权所有, 侵权必究。

书 名: Java 类库手册

著作责任者: 玄伟剑等译

责任编辑: 张豫夫 郭佑民

标准书号: ISBN 7-301-03476-8/TP·349

出版者: 北京大学出版社

地 址: 北京市海淀区中关村北京大学校内 100871

电 话: 出版部 62752015 发行部 62559712 编辑部 62752032

排版者: 盛达电脑激光照排中心

印刷者: 北京大学印刷厂

发行者: 北京大学出版社

经销者: 新华书店

787×1092 毫米 16 开本 76 印张 2460 千字

1997 年 10 月第一版 1997 年 10 月第一次印刷

定 价: 156.00 元

前 言

如何使用本书

本书是作为参考书而非指导书编写的。书的结构安排在于减少您查找一个类或类成员的时间。书中不对 Java 语言的任何部分进行介绍。另有一些书可供您学习语言使用。Ken Arnold 和 James Gosling 编写的《Java™ 程序设计语言》，以及 James, Gosling, Bill Joy 和 Guy Steele 编写的《Java™ 语言详述》都是学习 Java 语言较好的资料。

第一部分:Java 包总览

这部分章节简要介绍每个包以及所有在其中出现的类,包括反映这些类的继承层次的图表。

第二部分:Java 类库按字母顺序的参考

这部分章节把所有的类按字母顺序排列。本书在结构上最明显的特征也许就在于类的排列顺序。大多数含 API 的 Java 书按字母顺序排列每个包的类,然后按字母顺序排列这些包。这种格式的问题在于查找一个类需要花两步或更多步。如果您不大了解 Java 包,要查找一个类基本上得逐个遍查八个包。如果您对 Java 包有所了解,首先得找到相应的包,然后才能找到要找的类,即使有一个所有类的索引,也得先找到索引,然后才能找到类。

本书按字母顺序排列类,不考虑包名。这使得查找一个类就像在字典中查找一个单词一样简单直接。

每一章描述一个类。内容包括类描述、类层次图、类的范例、成员概述以及对该类各成员的描述。

类描述

在类描述中,我们描述类的所有属性。例如,Graphic 类的属性包括当前颜色、字体、绘制方式、原点及剪裁矩形。我们集中描述一个类的所有属性及其含义,而不是把属性描述分散到整个成员描述中,这样更易于了解一个类的所有功能。

成员描述中用到的所有术语都在类描述中进行介绍和描述。如果您发现某个成员的描述不够详细,那么您可以查阅类描述以得到更多信息。

类层次图

在 Java API 中每个类都有一个类图,类图展示了类的所有祖先、兄弟、直接后代以及该类实现的所有接口。在这些图中,如果一个包名出现在一个类名或接口名之前,那么这个类或接口与当前类不在同一包中。

在类图中,我们直观地区分三种 Java 实体:

1. 类
2. 抽象类
3. 接口

我们用基于实体被实现程度的助记符表示该实体。例如,在一个类中,所有的类方法均已实现,所以类是“充满的”。在一个抽象类中,有一个或更多的方法尚未实现,所以抽象类是“部分充满的”。最后,一个接口没有任何已实现的方法(只有方法符号),所以接口是“空的”。

范例

原则上我们希望 Java API 的每一成员都有不同的范例。由于时间所限,我们已尽力做到的是保证每个成员至少在一个范例中出现。

我们力求每个范例都尽可能说明了成员的典型用法。例如,在“按钮”的范例中,我们不仅说明了如何创建一个按钮,还说明了如何处理按钮事件。在某些情况下,我们还尽量说明了 Java API 中其它一些类的用法。例如,在 Graphics. draw Oval () 的范例中,我们不仅说明如何画一个椭圆,还说明如何用 Data Input Stream 类从标准输入设备读入整数来定位椭圆。只要不使范例混淆,我们认为适当介绍 Java API 中其它的一些类是帮助您熟悉 Java API 中所有可用类的一种好途径。

大多数范例使用相同的名称和结构,这主要是为了便于更易理解它们。当您多读一些范例之后,您就会习惯它们的结构,要费神理解的东西就越来越少了。

成员概述

每个类的成员概述部分用于帮助您使用一个不熟悉的类。它把类成员归结成特定于该类的类别。例如,在 List 类中,选择方法类别列举了所有有关选择的方法。这是为了建立该类成员的一个快速总览。所以,除方法名外,它不包括任何语法信息。

成员描述

在每个类的章节中,成员描述按字母顺序排列,而不管它们是哪种方法或域。这是为了能尽快查找成员。

在成员描述中,重载方法都放在一起,因为它们有非常类似的功能。通常我们提供了不同的重载形式以方便程序员说明参数。例如,某些重载通过提供公用缺省值来消去参数。我们用“如果参数 P 未被说明,则其缺省值为 3.14”这种形式的短语来描述缺省参数。其它的重载采用不同的值表示法。例如,一种重载可把某个特定的参数作为整数,而另一种重载可能把该参数作为一个包含整数的字符串。

每个成员描述包括以下的部分或全部域:

- 用途 对该成员用途的简要描述。
- 语法 对该成员的语法描述。
- 描述 对该成员的详尽描述。
- 参数 该成员接收的参数。如果存在,按字母顺序列出。
- 返回值 该成员的返回值(如果存在)。
- 异常 该成员抛出的异常和错误。如果存在,按字母顺序列出。
- 参见 其它相关的类或成员。如果存在,按字母顺序列出。
- 覆盖 该成员覆盖的方法(如果存在)。
- 范例 说明该成员用法的程序范例。有时是对说明该方法用法的其它成员或类范例的引用。

如何获取范例

本书的所有程序范例均已有 Solaris 或 Windows 95 平台(或两者兼可)Java 1.0.2 发布版的 FCS 版本上编译运行通过。大多数完整的范例可联机获取。您可以使用以下的 URL 获取这些范例以及有关本书的其它信息。

<http://aw.com/cp/chan-lee.html>

本书约定

Lucida San Typewriter 字体用于范例、语法声明、类名、方法名以及域名。

斜体用于定义一个新术语以及表示强调。

致谢

我们感谢使本书得以完成的所有人员。

Mike Hendrickson, 本书的主编,他协调组织了整项工作。更重要的,在过去几个月里,他给了我们精神上的莫大支持。

Lisa Friendly, 作为丛书编辑, 她使这项工作得以开始。而且, 任何时候她都充分信任我们。

Katie Duffy, Simone Payment, Avanda Peters, Laura Michaels 和 Marty Rabinowitz, 协助完成了本书, 从与校阅人员交流协商, 到制作图表和类层次图, 直至润饰本书的版式。

尽管本书内容极其繁多, Tom Wrensch 和 Maurice Fitzgerald II, 仍审阅了全书。另外, Kevin Kluge, Doug Kramer, Bhavesh Mehta, James Robins 和 Roland Schemers 提供了有用的反馈信息。

在整理本书的初稿时, Jayashree Vasudevan 给了我们极为有益的帮助。

最重要的, 感谢 Patrick 领导的 Java 组全体成员的伟大友谊和他们创造的这段美妙的工作时光。

最后, 我们不能忘记我们 3 岁的 Melissa 和 5 岁的 Kevin。一个又一个周末, 我们不停地工作, 工作, 一直工作了六个月。他们一直耐心地忍受着我们的“现在不行”, 一直和我们在一起。现在, 书已经出版, 我们准备做些什么呢? 那还用说, 我们要到迪斯尼乐园去!

Patrick Chan

Rosanna Lee

1996 年 6 月

目 录

前 言	(Ⅹ)
-----------	-----

Java 包概述

java.applet	(1)
java.awt	(2)
java.awt.image	(7)
java.awt.peer	(9)
java.io	(11)
java.lang	(14)
java.net	(20)
java.util	(23)

按字母排序的 Java 类库

AbstractMethodError	(25)
Applet	(27)
AppletContext	(41)
AppletStub	(46)
ArithmeticException	(48)
ArrayIndexOutOfBoundsException	(50)
ArrayStoreException	(52)
AudioClip	(54)
AWTError	(55)
AWTException	(56)
BitSet	(57)
Boolean	(63)
BorderLayout	(67)
BufferedInputStream	(72)
BufferedOutputStream	(79)
Button	(83)
ButtonPeer	(89)
ByteArrayInputStream	(90)
ByteArrayOutputStream	(94)

Canvas	(101)
CanvasPeer	(106)
CardLayout	(107)
Character	(113)
Checkbox	(125)
CheckboxGroup	(136)
CheckboxMenuItem	(141)
CheckboxMenuItemPeer	(146)
CheckboxPeer	(147)
Choice	(149)
ChoicePeer	(157)
Class	(158)
ClassCastException	(163)
ClassCircularityError	(165)
ClassFormatError	(167)
ClassLoader	(168)
ClassNotFoundException	(174)
Cloneable	(176)
CloneNotSupportedException	(177)
Color	(179)
ColorModel	(192)
Compiler	(198)
Component	(200)
ComponentPeer	(237)
Container	(243)
ContainerPeer	(261)
ContentHandler	(262)
ContentHandlerFactory	(264)
CropImageFilter	(267)
DatagramPacket	(273)
DatagramSocket	(277)
DataInput	(282)
DataInputStream	(287)
DataOutput	(295)
DataOutputStream	(299)
Date	(307)
Dialog	(321)
DialogPeer	(328)

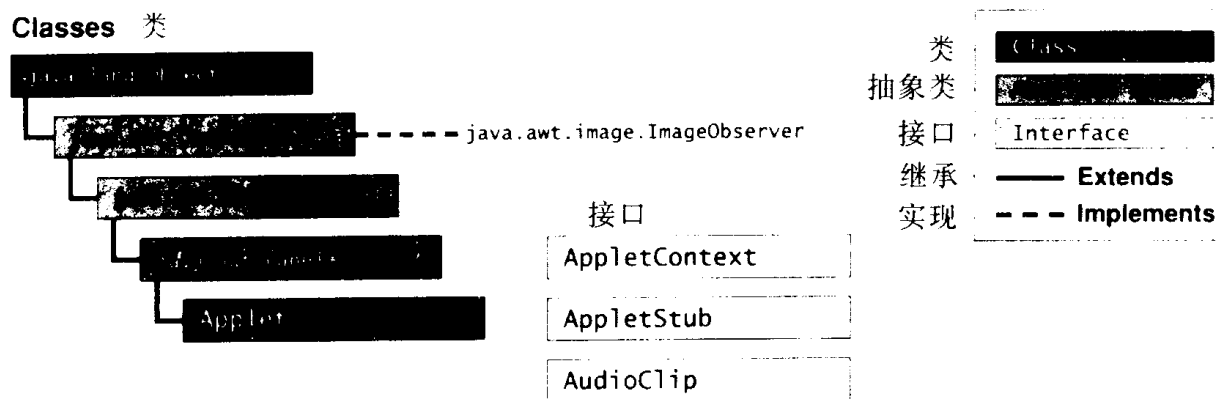
Dictionary	(329)
Dimension	(335)
DirectColorModel	(338)
Double	(345)
EmptyStackException	(353)
Enumeration	(355)
EOFException	(357)
Error	(359)
Event	(360)
Exception	(385)
File	(389)
FileDescriptor	(401)
FileDialog	(404)
FileDialogPeer	(411)
FileInputStream	(413)
FilenameFilter	(418)
FileNotFoundException	(420)
FileOutputStream	(422)
FilteredImageSource	(426)
FilterInputStream	(430)
FilterOutputStream	(437)
Float	(441)
FlowLayout	(449)
Font	(454)
FontMetrics	(464)
Frame	(476)
FramePeer	(491)
Graphics	(493)
GridBagConstraints	(535)
GridBagLayout	(552)
GridLayout	(568)
Hashtable	(572)
IllegalAccessError	(580)
IllegalAccessException	(582)

IllegalArgumentException	(584)
IllegalMonitorStateException	(586)
IllegalThreadStateException	(588)
Image	(590)
ImageConsumer	(600)
ImageFilter	(610)
ImageObserver	(618)
ImageProducer	(624)
IncompatibleClassChangeError	(631)
IndexColorModel	(632)
IndexOutOfBoundsException	(640)
InetAddress	(642)
InputStream	(647)
Imsets	(653)
InstantiationError	(658)
InstantiationException	(660)
Integer	(662)
InternalError	(671)
InterruptedException	(672)
InterruptedIOException	(674)
IOException	(676)
Label	(678)
LabelPeer	(684)
LayoutManager	(685)
LineNumberInputStream	(692)
LinkageError	(699)
List	(700)
ListPeer	(713)
Long	(716)
MalformedURLException	(725)
Math	(727)
MediaTracker	(736)
MemoryImageSource	(746)
Menu	(752)
MenuBar	(757)
MenuBarPeer	(765)
MenuComponent	(767)

MenuComponentPeer	(770)
MenuContainer	(771)
MenuItem	(775)
MenuItemPeer	(781)
MenuPeer	(782)
NegativeArraySizeException	(784)
NoClassDefFoundError	(786)
NoSuchElementException	(788)
NoSuchFieldError	(790)
NoSuch MethodError	(792)
NoSuchMethodException	(794)
NullPointerException	(796)
Number	(798)
NumberFormatException	(801)
Object	(803)
Observable	(815)
Observer	(820)
OutOfMemoryError	(822)
OutputStream	(823)
Panel	(827)
PanelPeer	(830)
PipedInputStream	(831)
PipedOutputStream	(836)
PixelGrabber	(839)
Point	(846)
Polygon	(851)
PrintStream	(856)
Process	(862)
Properties	(868)
ProtocolException	(873)
PushbackInputStream	(875)
Random	(881)
RandomAccessFile	(885)
Rectangle	(899)
RGBImageFilter	(910)

Runnable	(922)
Runtime	(924)
RuntimeException	(931)
Scrollbar	(933)
ScrollbarPeer	(943)
SecurityException	(945)
SecurityManager	(947)
sequenceInputStream	(958)
ServerSocket	(961)
Socket	(968)
SocketException	(974)
SocketImpl	(976)
SocketImplFactory	(985)
Stack	(987)
StackOverflowError	(991)
StreamTokenizer	(993)
String	(1007)
StringBuffer	(1023)
StringBufferInputStream	(1031)
StringIndexOutOfBoundsException	(1035)
StringTokenizer	(1037)
System	(1042)
TextArea	(1050)
TextAreaPeer	(1057)
TextComponent	(1059)
TextComponentPeer	(1071)
TextField	(1073)
TextFieldPeer	(1079)
Thread	(1081)
ThreadDeath	(1102)
ThreadGroup	(1103)
Throwable	(1117)
Toolkit	(1121)
UnknownError	(1132)
UnknownHostException	(1133)
UnknownServiceException	(1135)

UnsatisfiedLinkError	(1137)
URL	(1139)
URLConnection	(1150)
URLEncoder	(1169)
URLStreamHandler	(1172)
URLStreamHandlerFactory	(1175)
UTFDataFormatException	(1177)
Vector	(1179)
VerifyError	(1191)
VirtualMachineError	(1192)
Window	(1193)
WindowPeer	(1202)



applet 框架涉及两种实体：小应用程序上下文(applet context)和小应用程序(applet)。applet 上下文是负责装载和运行 applet 的应用程序。例如,applet 上下文可能是 Web 浏览器或 applet 开发环境,如图 1 所示。

applet 是一种可嵌入窗口,带有几个可供 applet 上下文初始化启动和终止 applet 的额外方法。

该包具有创建 applet 所必需的类以及 applet 用来同其 applet 上下文通信的类。

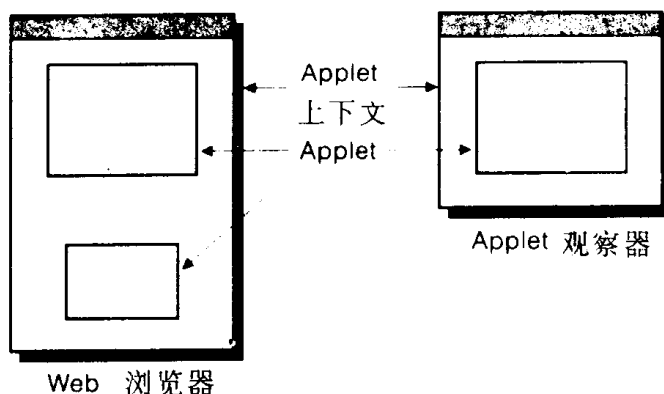


图 1 applet 上下文和 applet 的关系

Applet

一个类通过子类化此类而成为 applet。它包含 applet 可以覆盖的方法以及从 URL 获取媒体的方法。

Applet	用于创建 applet。所有的 applet 必须是该类的子类。
--------	----------------------------------

声音

该类包含在 URL 上获取和播放声音片段的方法。声音片段可以被一次或循环不断地播放。

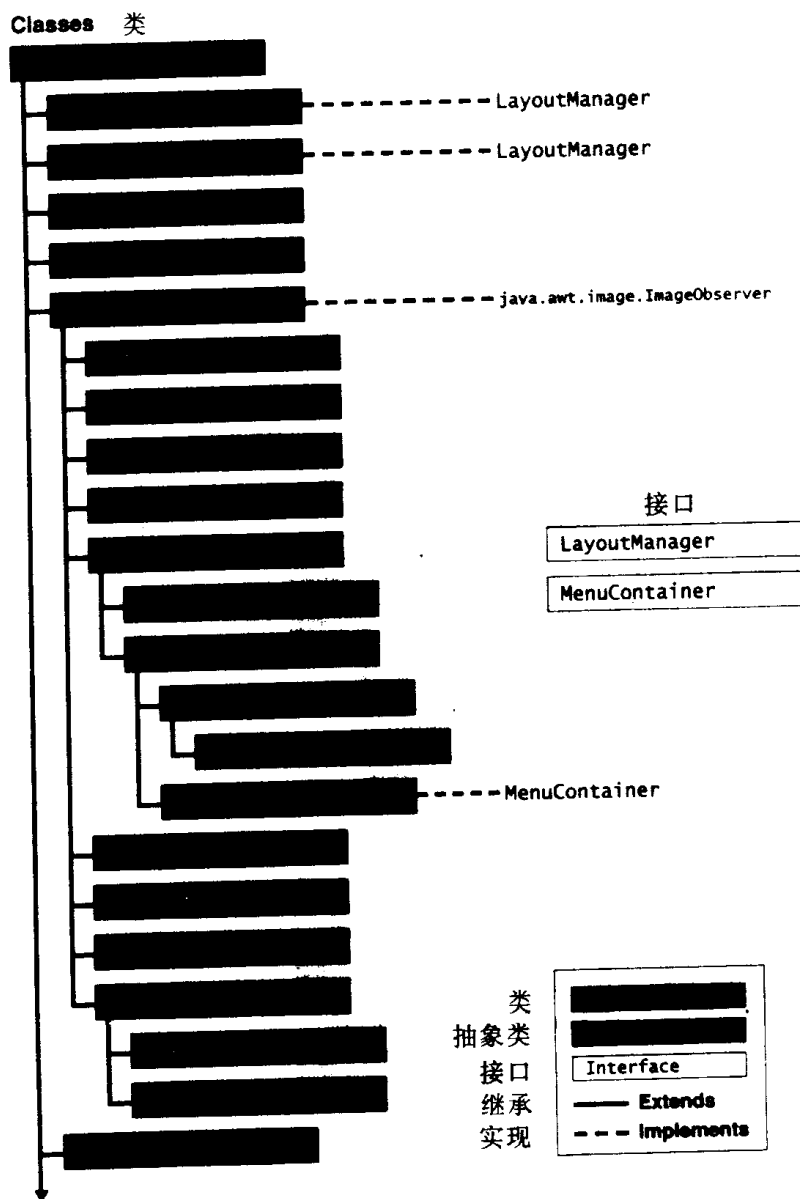
Audio Clip	用来播放和停止声音片段。
------------	--------------

Applet 上下文

这些接口指定了所有 applet 上下文必须提供给 applet 的方法。例如,在 applet 上下文的状态条显示状态信息的方法以及发现 applet 上下文可能包含的其它 applets 的方法。

Applet Context	applet 用它同 applet 上下文通信。
Applet Stub	在 Applet Context 的实现中使用。不直接使用。

java. awt



这组类用于在绘制表面上绘制图形、文本和图像。绘制表面可以包括屏幕、屏幕外图像以及打印机，如图 2 所示。

Graphics	用来绘制直线和矩形等基本图形，也可用来绘制图像。
Color	代表一种颜色，具有将 RGB 和 HSB 值相互转换的方法。
Font	代表一种字体。
Font Metrics	用来确定一种字体的信息，例如高度和字符宽度。
Image	代表一幅图像，具有获取其大小的方法。
Media Tracker	用来预装载图像。

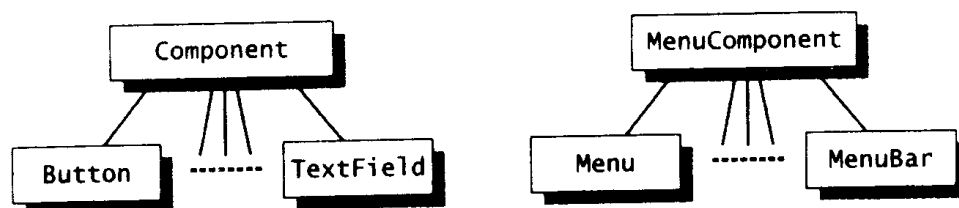


图 3 Component 和 MenuComponent

组件

这些类代表 AWT 提供的用来建立用户界面的组件。每种组件产生不同的事件集，所以你必须查阅文档来确定组件产生的事件类型。

在这个组及下述容器组中的所有类都是 Component 或 MenuComponent 的子类，如图 3 所示。要了解所有组件的公共属性集，可见这些类。

Button	一种当单击时产生事件的组件。
Canvas	一种通常用来绘制图形的组件。
Check box	一种维护布尔状态的组件。
Check box Group	用来实现单选的按钮组。
Check box Menu Item	一种维护布尔状态的菜单项。
Choice	一种实现下拉列表框的组件。
Component	组件的超类。
Label	一种显示文本字符串的组件。
List	一种显示项目列表的组件。
Menu	一种包含菜单项的菜单。
Menu Bar	一种包含菜单的菜单栏。
Menu Component	菜单组件的超类。
Menu Container	菜单容器的超类。
Menu Item	一种可插入到菜单中的菜单项。
Scroll bar	一种实现滚动条的组件。
Text Area	一种用来编辑多行文本字符串的组件。
Text Component	文本组件的超类。
Textfield	一种用来编辑单行文本字符串的组件。

容器

容器可以包含组件和其它容器。容器也可以具有控制容器中组件可视摆放的布局管理器。一些容器是最上层窗口,不能被嵌入到另一个容器中。

Container	容器的超类。
Dialog	对话框的超类。
File Dialog	用来选择已存在文件或命名新文件的对话框。
Frame	具有标题、菜单栏和边框的最上层窗口。
Panel	可嵌入其它容器的容器。
Window	不具有标题、菜单栏和边框的最上层窗口。

布局

AWT 提供了一些有用的布局管理器。功能最多但也最难使用的是 Grid Bag Layout 布局管理器。如果它们都不合适,你可以使用 Layout Manayer 接口建立自己的布局管理器。

Border Layout	一种沿每条边和在中心放置组件的布局管理器。
Card Layout	一种同一时间内显示一个组件的布局管理器。
Flow Layout	一种从左到右,从上到下放置组件的布局管理器。
Grid Bag Constraints	用来规定 GridBag Layout 对象中的限制。
Grid Bag Layout	一种在单元尺寸可变的网格中放置组件的布局管理器。
Grid Layout	一种在单元尺寸固定的网格中放置组件的布局管理器。
Layout Manager	布局管理器必须实现的接口。

几何

这些类用来持有值(value)的集合。例如,Rectangle 对象持有代表矩形各边位置的 4 个值。

Dimension	用来指定矩形的尺寸(宽度和高度)。
Insets	用来指定矩形的内框(顶、左、底和右)。
Point	用来使用 x,y 坐标指定一个点。
Polygon	用来持有点的数组。
Rectangle	用来指定矩形的位置和尺寸(x,y;宽度和高度)。

事件

事件(event)是一种在用户同组件交互作用时由组件产生的对象。组件可以产生许多种鼠标和键盘事件。事件自动沿组件层次向上流动并被一个或多个事件处理者(event handler)处理。图 4 展示了一个组件层次以及一个事件在该层次中的流动。图 4 展示了用户事件首先被传递到按钮组件,然后传遍它的所有祖先。在事件沿层次向上流动时,任何正在接收该事件的组件都可以处理该事件,并因而阻止事件继续向上流动。

关于事件处理程序的描述见 Component. handleEvent()。Event 类封装了 AWT 事件的所有细节。

Event	封装一个事件的所有信息。
-------	--------------

工具箱