

<<Java程序设计教程>>

图书基本信息

书名：<<Java程序设计教程>>

13位ISBN编号：9787566403377

10位ISBN编号：7566403370

出版时间：安徽大学出版社

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：<http://www.tushu007.com>

<<Java程序设计教程>>

书籍目录

第1章 Java概述 1.1 Java语言简介 1.1.1 Java语言的发展 1.1.2 Java的工作原理 1.2 编写Java程序 1.2.1 建立Java开发环境 1.2.2 Java程序规范 1.2.3 接触Java程序 1.3 小结 习题1 第2章 Java语言基础 2.1 Java的关键字、标识符、各种数据类型 2.1.1 关键字 2.1.2 标识符 2.1.3 数据类型 2.2 Java的各种运算符和表达式 2.2.1 算术运算符 2.2.2 关系运算符 2.2.3 位运算符 2.2.4 其他运算符 2.2.5 运算符的优先级与结合性 2.3 类型转换 2.4 语句 2.4.1 表达式语句 2.4.2 复合语句 2.4.3 选择语句 2.4.4 switch开关语句 2.4.5 循环语句 2.4.6 跳转语句 2.5 数组 2.5.1 数组声明 2.5.2 创建数组空间 2.5.3 数组元素的引用及初始化 2.6 字符和字符串的处理 2.6.1 字符 2.6.2 字符串 2.7 小结 习题2 第3章 面向对象的Java 3.1 介绍类 3.1.1 类基础 3.1.2 声明对象 3.1.3 给对象引用变量赋值 3.1.4 方法 3.1.5 构造方法 3.1.6 this关键字 3.1.7 垃圾回收 3.1.8 finalize () 方法 3.1.9 一个堆栈类 3.2 类的高级话题 3.2.1 方法重载 3.2.2 对象作为参数 3.2.3 参数是如何传递的 3.2.4 返回对象 3.2.5 递归 3.2.6 访问控制 3.2.7 理解static 3.2.8 介绍final 3.2.9 重新温习数组 3.2.10 介绍嵌套类和内部类 3.2.11 探索String类 3.2.12 命令行参数 3.3 类间继承 3.3.1 继承的基础 3.3.2 使用super 3.3.3 创建多级类层次 3.3.4 何时调用构造方法 3.3.5 方法重载 3.3.6 动态方法调度 3.3.7 抽象类 3.3.8 继承中使用final 3.3.9 Object类 3.4 包和接口 3.4.1 包 3.4.2 访问保护 3.4.3 引入包 3.4.4 关键字interface 3.4.5 接口可以扩展 3.5 小结 习题3 第4章 常用工具类 4.1 Data类 4.2 Calendar类 4.3 Math类 4.4 向量类Vector 4.5 栈类Stack 4.6 哈希表类Hashtable 4.7 小结 习题4 第5章 图形用户界面 5.1 AWT实现用户界面 5.1.1 java.awt包结构 5.1.2 容器类和组件类 5.1.3 常用的容器组件类 5.1.4 布局管理类 5.1.5 AWT事件处理 5.1.6 常用AWT组件类 5.2 Swing实现用户界面 5.2.1 Swing简介 5.2.2 Swing组件和容器 5.2.3 选择界面风格 5.3 用户界面在Applet中的实现 5.3.1 Applet的基本工作原理 5.3.2 Applet小程序中实现用户界面 5.4 小结 习题5 第6章 异常处理 第7章 Java多线程机制 第8章 流式输入输出与文件处理 第9章 Java网络编程 第10章 Java数据库操作 参考文献

<<Java程序设计教程>>

章节摘录

版权页： 插图： 5.1.2 容器类和组件类 Java的图形用户界面由组件构成，例如按钮（Button）、文本输入框（TextField）、标签（Label）等都是组件，其中有一类特殊的组件称为“容器”（Container）。

所谓“容器组件”，是指能够包含其他组件的组件，各种组件（包括大部分容器）可以通过add（）方法添加到容器中。

一个设计良好的用户界面应该是高度结构化的，而不是一些简单GUI组件的无序组合。

而容器组件恰恰有助于实现GUI的结构化。

AWT中提供的容器组件类有面板（Panel）、窗口（Window）、框架（Frame）和对话框（Dialog）等。

其中Panel没有具体的图形表示，但它却是实现GUI结构化的一个至关重要的组件，我们通常用它来实现GUI基本组件的组合；Frame则定义了一个包含标题条、系统菜单栏、最大化/最小化按钮及可选菜单条的完整的窗口；而Dialog主要用于实现应用程序与用户的信息交流。

上述的容器组件虽然功能各异，但是它们都有一个共同的特点：即能够包含其他GUI组件。

然而容器组件是如何管理所包含的GUI组件，如何确定这些组件的位置和大小呢？

这就涉及AWT中一个很重要的概念——“布局管理”。

采用布局管理与Java语言的跨平台特性密切相关。

因为不同平台显示文字、图形的方式并不相同，如果显式地指定一个组件的绝对位置和大小，那么它在某一个平台上能够正常显示，而在其他平台上的显示效果却可能不尽如人意。

因此，AWT中定义了布局管理器接口LayoutManager，LayoutManager的实现类可以自动根据运行平台的不同来布置所有可视化GUI组件。

容器组件既可以使用缺省的布局管理器，也可以通过setLayout方法来设置所需要的布局管理器。

一旦确定了布局管理方式，容器组件就可以使用相应的add（）方法向其中加入其他GUI组件。

布局管理器将在后边5.1.4小结中介绍。

读者可以通过第一章介绍过的帮助文档查找Component类中的成员方法和成员变量。

5.1.3 常用的容器组件类 容器java.awt.Container是Component的子类，一个容器可以容纳多个组件，并使它们成为一个整体。

容器可以简化图形界面的设计，以整体结构来布置界面。

所有的容器都可以通过add（）方法向容器中添加组件。

AWT中提供的容器组件类有面板（Panel）、窗口（Window）、框架（Frame）和对话框（Dialog）等。

1.Frame类 一般我们要生成一个窗口，通常是用Window的子类Frame来进行实例化，而不是直接用Window类。

Frame的外观就像我们平常在windows系统下见到的窗口，有标题、边框、菜单、大小等。

每个Frame的对象实例化以后，都是没有大小和不可见的，因此必须调用setSize（）来设置大小，调用setVisible（true）来设置该窗口为可见的。

窗口是一个容器，可以向窗口添加组件。

需要注意的是，窗口默认地被系统添加到显示器屏幕上，因此，不允许将一个窗口添加到另一个容器中。

Frame类常用方法： Frame（）：该构造方法可以创建一个无标题的窗口，窗口的默认布局为BorderLayout布局。

Frame（String s）：该构造方法可以创建一个标题s的窗口，窗口的默认布局为BorderLayout布局。

<<Java程序设计教程>>

编辑推荐

《Java程序设计教程》从Java语言的易懂易学特点出发，介绍Java面向对象的程序设计要点、难点和应用实例，力求结构严谨，由浅入深、循序渐进。
在介绍基本概念和方法的同时，配有精心设计的实例，使读者能够更快地理解、应用该语言的各个功能，培养读者运用Java语言进行编程的实践能力。

<<Java程序设计教程>>

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:<http://www.tushu007.com>