

跟我学Java之JavaGUI的发展和演化 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/144/2021_2022__E8_B7_9F_E6_88_91_E5_AD_A6J_c104_144602.htm 概览 当Java在1995年的春天第一次发布的时候，它包含了一个叫AWT(Abstract Windowing Toolkit)的库，用来构建图形用户界面应用程序。Java很有雄心的宣言 - - "write once, run anywhere"许诺：一个具有下拉菜单，命令按钮，滚动条以及其他常见的GUI控件的应用程序将能够在各种操作系统上运行而不必重新编译成针对某一平台的二进制代码，包括Microsoft Windows, Sun 's own Solaris, Apple 's Mac OS以及Linux。数码精品，购物零风险 建站_花小钱做大事业 新版Google工具栏 视频电话免费通全球 虽然最初Java是支持操作系统独立的应用程序开发，在Java提出"write once, run anywhere"宣言那段时间的革命导致了Java applet和主导桌面应用程序计划的产生。尽管从那以后大部分构建桌面应用程序的成就都慢慢衰退，Java构建图形用户界面的能力反倒增强了。跟踪Java GUI的发展和演化，我们将发现3个主要的构建窗口程序库:AWT,Swing和SWT(Standard Widget Toolkit)。在这一章里我们将查看和分析这3个库中的每一个库，并且我们将一起来看一看第4个库 - - JFace。其实JFace不算一个真正的构建窗口程序库，而是在基于SWT之上的一个抽象层。AWT 很多围绕着介绍Java技术的令人激动的地方都基于applets - - 一个可以让程序通过Internet发布并在浏览器内执行的新技术。用户和开发人员都热衷于斯，因为applets许诺将简化跨平台应用程序的开发，维护和发布，而这是商业软件开发中几个最富挑战性的话

题。为了方便用Java构建图形用户界面，Sun最初提供了一个在所有平台下具有的独特Java外观的图形界面库。Sun在applet技术策略方面的首要伙伴Netscape提出applets应该维持和运行时平台一样的外观。他们希望applets在某一平台下在显示和行为上能够像其他应用程序一样。为了实现Netscape的"本地外观"的目标，在JDK的第一个发布版中包含了AWT这个库。AWT的缺省实现使用了"对等"机制，即每一个Java GUI窗口部件都在底层的窗口系统中有一个对应的组件。例如，每一个java.awt.Button对象将在底层窗口系统中创建一个唯一对应的button。当用户点击那个按钮的时候，事件将从本地实现库传送到Java虚拟机里，并且最终传送到与java.awt.Button对象相关联的逻辑。对等系统的实现以及Java组件与对等组件之间的交流的实现都隐藏在底层JVM实现中，Java语言级的代码仍然跨平台。尽管如此，为了保持"write once, run anywhere"的承诺，Java不得不妥协和折衷。特别的，Java采用了"最小公分母"的方法，即AWT仅仅提供所有本地窗口系统都提供的特性。这就需要开发人员为更多高级特性开发他们自己的高级窗口部件，然后提供给用户不同的使用体验。其他的问题也减缓了人们对applets的接受和承认。Applets运行在一个安全的"沙箱"里面并且能够阻止恶意的applets对文件系统，网络连接等资源的滥用。尽管沙箱提供了安全性，但它"阉割"了应用程序。毕竟一个应用程序不能运用一个网络连接来保存一个文件是不好的。Java GUI应用程序也不能像本地程序一样响应灵敏。这是在当前硬件平台和Java的解释性天性下预期结果。所以，用AWT开发的应用程序既缺少流行GUI程序的许多特性，又不能达到在显示和行为上像用本地窗口构建库开发的

程序一样的目标。应该有一个更好的库来让Java GUI取得成功。 Swing 于1997年JavaOne大会上提出并在1998年5月发布的JFC(Java Foundation Classes)包含了一个新的使用Java窗口开发包。这个新的GUI组件叫做Swing，感觉到它是对AWT的升级，并且看起来对Java占据计算机世界很有帮助。对Java来说已经万事具备了:可下载的applets将是未来的软件，人们将从其他操作系统转向JavaOS，从传统的计算机转向叫做JavaStation的瘦客户端网络计算机，Microsoft将最终因为不能在桌面程序领域与之想抗衡而被废黜。虽然这些景象从来没有实现，Swing作为Java applets和applications的GUI库倒确实十分繁荣。 Swing架构 尽管"Swing"仅仅是这个新组件的指代名称，它一直持续使用到今天。可能是因为这个名称太贴切了，Swing尝试着以以下几种方式改变公认的观点: AWT依赖对等架构，用Java代码包装本地窗口部件，Swing却根本不使用本地代码和本地窗口部件 AWT把绘制屏幕交给本地窗口部件，Swing自己的组件绘制自己 因为Swing不依赖本地窗口部件，它可以抛弃AWT的最小公分母的方法并在每个平台下实现每个窗口部件，从而创建一个比AWT更强大的开发工具包Swing缺省情况下采用本地平台的显示外观。然而，它不仅限于此，而是还可以采用插件式的显示外观。因此Swing应用程序可以看起来想Windows应用程序，Motif应用程序，Mac应用程序甚至它自己的显示外观 - - "金属"。所以，Swing应用程序可以完全忽略它运行时所在的操作系统环境并且仅仅看起来像自己。这是单调一致的桌面应用程序外观的一大挑衅。想象一下Swing有多傲慢! 尽管如此，Swing组件超越了简单的窗口部件，它体现了正不断出现的设计模式以

及一些最佳实践。采用Swing，你不仅仅得到GUI窗口部件的句柄和它所包含的数据，而是定义一个模型去保存数据，定义一个视图去显示数据，定义一个控制器去响应用户输入。事实上，大部分Swing组件的构建是基于MVC(model-view-controller)模式的。MVC使应用程序开发变得更清晰，更易维护和管理。100Test 下载频道开通，各类考试题目直接下载。详细请访问 www.100test.com