



Visual J++ 参考 6.0 详解

王育新 张 勇 张俊波 编著



清华大学出版社



<http://www.tup.tsinghua.edu.cn>

Visual J++ 6.0

参考详解

王育新 张勇 张俊波 编著

清华大学出版社

(京)新登字 158 号

内 容 简 介

Visual J++是微软公司推出的 Java 语言开发工具,由于其功能强大,专业化程度高,受到了广大程序员的欢迎。本书以 Visual J++ 6.0 版为基础,详细地介绍了 Visual J++ 6.0 的 API,包括 API 库中的各个包,以及这些包中的类和接口,对使用这些类可能出现的情况也做了细致的介绍。通过阅读本书,有助于从整体上掌握 Visual J++提供的程序设计功能,从而提高软件开发的效率。本书适合所有 Visual J++的程序员阅读。

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,无标签者不得销售。

书 名: Visual J++ 6.0 参考详解

作 者: 王育新 张勇 张俊波

出版者: 清华大学出版社(北京清华大学校内,邮编 100084)

<http://www.tup.tsinghua.edu.cn>

印刷者: 昌平环球印刷厂

发行者: 新华书店总店北京发行所

开 本: 787×1092 1/16 **印张:** 25.5 **字数:** 997 千字

版 次: 1999 年 11 月第 1 版 1999 年 11 月第 1 次印刷

书 号: ISBN 7-302-03850-3/TP·2174

印 数: 0001~5000

定 价: 46.00 元

前 言

随着因特网(Internet)及万维网(World Wide Web)的迅速发展,Java 语言已迅速普及并成为网络语言的主流。为了满足 Java 语言开发人员的需求,微软公司开发的 Visual J++ 程序设计软件,将编程功能的高效性集成于一体,堪称是从初学者到专业程序开发人员都一直期盼的一个 Java 程序开发工具。目前,继 Visual J++ 1.1 之后,微软公司又推出了 Visual J++ 6.0,它比 Visual J++ 1.1 的开发效率更高、功能更强大、使用也更加方便,受到了 Java 程序员的广泛欢迎。

本书是 Visual J++ 6.0 API 的语言参考手册,其全面、系统、清晰地介绍了 Visual J++ 6.0 API 库的各个包及各包中的类、接口等内容,对使用 Visual J++ 6.0 开发 Java 程序的广大计算机专业人员有很大的参考价值。

本书在内容上共分 18 章,将 Java API 各类与接口按“包”分类,为用户提供了很大的方便,用户只要按照自己的需要查找相应的包,就能找到所需要的类、接口、例外以及可能产生的错误。

书中各章简要内容如下:

第 1 章介绍了 java.applet,它提供了编写 java.applet 所需的各种类和接口。

第 2 章介绍了 java.awt,它提供了图形用户界面的基本元素,用户利用此包可以实现友好界面的编辑。

第 3 章介绍了 java.awt.datatransfer,它提供了数据转移的机制。

第 4 章介绍了 java.awt.event,用户利用此包可以处理各种事件。

第 5 章介绍了 java.awt.image,它提供了处理图像的各种功能,用户利用此包可以使应用程序更加丰富多彩。

第 6 章介绍了 java.beans,它提供了处理 bean 的类和接口。

第 7 章介绍了 java.io,它提供了读写操作的基本功能。

第 8 章介绍了 java.lang,它是 java 的核心,提供了 java 的各种事务处理功能。

第 9 章介绍了 java.lang.reflect,利用此包可以实现信息的存取。

第 10 章介绍了 java.math,它提供了基本的运算操作功能。

第 11 章介绍了 java.net,通过此包可以实现各种网络的操作。

第 12 章介绍了 java.security,它提供了保证应用程序安全的基本功能。

第 13 章介绍了 java.security.acl,实现了对 ACL 的操作。

第 14 章介绍了 java.security.interfaces,它提供了管理密钥的一系列接口,

第 15 章介绍了 java.sql,它提供了对 SQL 操作的基本功能。

第 16 章介绍了 java.text,它提供了对数字,时间等文本的格式操作。

第 17 章介绍了 `java.util`, 它用于管理地区资源。

第 18 章介绍了 `java.util.zip`, 它提供了对数据流的处理功能。

在本书的完成过程中, 得到了许多同仁的大力协助。在此表示衷心感谢。由于时间仓促, 加之作者水平有限, 错误与不当之处在所难免, 希望广大读批评指正。

作 者

1999 年 8 月

目 录

第 1 章 包 java.applet	1	2.36 Insets 类	57
1.1 AppletContext 接口	1	2.37 Label 类	58
1.2 AppletStub 接口	1	2.38 List 类	58
1.3 AudioClip 接口	2	2.39 MediaTracker 类	62
1.4 Applet 类	2	2.40 Menu 类	64
第 2 章 包 java.awt	5	2.41 MenuBar 类	65
2.1 Adjustable 接口	5	2.42 MenuComponent 类	66
2.2 ItemSelectable 接口	5	2.43 MenuItem 类	67
2.3 LayoutManager 接口	6	2.44 MenuShortcut 类	68
2.4 LayoutManager2 接口	6	2.45 Panel 类	69
2.5 MenuContainer 接口	6	2.46 Point 类	69
2.6 PrintGraphics 接口	7	2.47 Polygon 类	70
2.7 Shape 接口	7	2.48 PopupMenu 类	71
2.8 AWTEvent 类	7	2.49 PrintJob 类	71
2.9 AWTEventMulticaster 类	9	2.50 Rectangle 类	71
2.10 BorderLayout 类	11	2.51 ScrollPane 类	74
2.11 Button 类	13	2.52 Scrollbar 类	75
2.12 Canvas 类	14	2.53 SystemColor 类	77
2.13 CardLayout 类	14	2.54 TextArea 类	79
2.14 Checkbox 类	15	2.55 TextComponent 类	81
2.15 CheckboxGroup 类	17	2.56 TextField 类	82
2.16 CheckboxMenuItem 类	17	2.57 Toolkit 类	84
2.17 Choice 类	18	2.58 Window 类	87
2.18 Color 类	19	2.59 AWTException 类	88
2.19 Component 类	22	2.60 IllegalComponentStateException 类	89
2.20 Container 类	30	2.61 AWTError 类	89
2.21 Cursor 类	34	第 3 章 包 java.awt.datatransfer	90
2.22 Dialog 类	34	3.1 ClipboardOwner 接口	90
2.23 Dimension 类	35	3.2 Transferable 接口	90
2.24 Event 类	36	3.3 Clipboard 类	90
2.25 EventQueue 类	39	3.4 DataFlavor 类	90
2.26 FileDialog 类	40	3.5 StringSelection 类	91
2.27 FlowLayout 类	41	3.6 UnsupportedFlavorException 类	92
2.28 Font 类	42	第 4 章 包 java.awt.event	93
2.29 FontMetrics 类	43	4.1 ActionListener 接口	93
2.30 Frame 类	45	4.2 AdjustmentListener 接口	93
2.31 Graphics 类	46	4.3 ComponentListener 接口	93
2.32 GridBagConstraints 类	51	4.4 ContainerListener 接口	93
2.33 GridBagLayout 类	53	4.5 FocusListener 接口	93
2.34 GridLayout 类	55	4.6 ItemListener 接口	93
2.35 Image 类	56	4.7 KeyListener 接口	94

4.8	MouseListener 接口	94	6.10	FeatureDescriptor 类	123
4.9	MouseMotionListener 接口	94	6.11	IndexedPropertyDescriptor 类	124
4.10	TextListener 接口	94	6.12	Introspector 类	125
4.11	WindowListener 接口	94	6.13	MethodDescriptor 类	126
4.12	ActionEvent 类	95	6.14	ParameterDescriptor 类	126
4.13	AdjustmentEvent 类	95	6.15	PropertyChangeEvent 类	126
4.14	ComponentAdapter 类	96	6.16	PropertyChangeSupport 类	127
4.15	ComponentEvent 类	96	6.17	PropertyDescriptor 类	127
4.16	ContainerAdapter 类	97	6.18	PropertyEditorManager 类	128
4.17	ContainerEvent 类	97	6.19	PropertyEditorSupport 类	128
4.18	FocusAdapter 类	97	6.20	SimpleBeanInfo 类	129
4.19	FocusEvent 类	97	6.21	VetoableChangeSupport 类	130
4.20	InputEvent 类	98	6.22	IntrospectionException 类	130
4.21	ItemEvent 类	99	6.23	PropertyVetoException 类	131
4.22	KeyAdapter 类	99			
4.23	KeyEvent 类	100	第 7 章 包 java.io	132	
4.24	MouseAdapter 类	101	7.1	DataInput 接口	132
4.25	MouseEvent 类	101	7.2	DataOutput 接口	133
4.26	MouseMotionAdapter 类	102	7.3	Externalizable 接口	134
4.27	PaintEvent 类	102	7.4	FilenameFilter 接口	134
4.28	TextEvent 类	102	7.5	ObjectInput 接口	134
4.29	WindowAdapter 类	103	7.6	ObjectInputValidation 接口	135
4.30	WindowEvent 类	103	7.7	ObjectOutput 接口	135
			7.8	Serializable 接口	135
第 5 章 包 java.awt.image	104		7.9	BufferedInputStream 类	136
5.1	ImageConsumer 接口	104	7.10	BufferedOutputStream 类	137
5.2	ImageProducer 接口	105	7.11	BufferedReader 类	138
5.3	ImageObserver 接口	105	7.12	BufferedWriter 类	139
5.4	AreaAveragingScaleFilter 类	106	7.13	ByteArrayInputStream 类	140
5.5	ColorModel 类	107	7.14	ByteArrayOutputStream 类	141
5.6	CropImageFilter 类	107	7.15	CharArrayReader 类	142
5.7	DirectColorModel 类	108	7.16	CharArrayWriter 类	142
5.8	FilteredImageSource 类	108	7.17	DataInputStream 类	143
5.9	ImageFilter 类	109	7.18	DataOutputStream 类	146
5.10	IndexColorModel 类	110	7.19	File 类	147
5.11	MemoryImageSource 类	111	7.20	FileDescriptor 类	150
5.12	PixelGrabber 类	113	7.21	FileInputStream 类	151
5.13	RGBImageFilter 类	116	7.22	FileOutputStream 类	152
5.14	ReplicateScaleFilter 类	117	7.23	FileReader 类	152
			7.24	FileWriter 类	153
第 6 章 包 java.beans	118		7.25	FilterInputStream 类	153
6.1	BeanInfo 接口	118	7.26	FilterOutputStream 类	154
6.2	Customizer 接口	119	7.27	FilterReader 类	155
6.3	PropertyChangeListener 接口	119	7.28	FilterWriter 类	155
6.4	PropertyEditor 接口	119	7.29	InputStream 类	156
6.5	VetoableChangeListener 接口	120	7.30	InputStreamReader 类	157
6.6	Visibility 接口	120	7.31	LineNumberInputStream 类	158
6.7	BeanDescriptor 类	121	7.32	LineNumberReader 类	159
6.8	Beans 类	121	7.33	ObjectInputStreams 类	159
6.9	EventSetDescriptor 类	122	7.34	ObjectOutputStream 类	162

7.35	ObjectStreamClass 类	165	8.14	Number 类	208
7.36	OutputStream 类	165	8.15	Object 类	209
7.37	OutputStreamWriter 类	166	8.16	Process 类	210
7.38	PipedInputStream 类	166	8.17	Runtime 类	211
7.39	PipedOutputStream 类	167	8.18	SecurityManager 类	212
7.40	PipedReader 类	168	8.19	Short 类	215
7.41	PipedWriter 类	168	8.20	String 类	216
7.42	PrintStream 类	168	8.21	StringBuffer 类	221
7.43	PrintWriter 类	170	8.22	System 类	223
7.44	PushbackInputStream 类	172	8.23	Thread 类	225
7.45	PushbackReader 类	173	8.24	ThreadGroup 类	228
7.46	RandomAccessFile 类	174	8.25	Throwable 类	230
7.47	Reader 类	177	8.26	Void 类	231
7.48	SequenceInputStream 类	178	8.27	ArithmeticException 类	231
7.49	StreamTokenizer 类	179	8.28	ArrayIndexOutOfBoundsException 类	231
7.50	StringBufferInputStream 类	181	8.29	ArrayStoreException 类	231
7.51	StringReader 类	182	8.30	ClassCastException 类	231
7.52	StringWriter 类	183	8.31	ClassNotFoundException 类	231
7.53	Writer 类	183	8.32	CloneNotSupportedException 类	232
7.54	CharConversionException 类	184	8.33	Exception 类	232
7.55	EOFException 类	184	8.34	IllegalAccessOperationException 类	232
7.56	FileNotFoundException 类	184	8.35	IllegalArgumentException 类	232
7.57	IOException 类	185	8.36	IllegalMonitorStateException 类	233
7.58	InterruptedIOException 类	185	8.37	IllegalStateException 类	233
7.59	InvalidClassException 类	185	8.38	IllegalThreadStateException 类	233
7.60	InvalidObjectException 类	185	8.39	IndexOutOfBoundsException 类	233
7.61	NotActiveException 类	185	8.40	InstantiationException 类	233
7.62	NotSerializableException 类	186	8.41	InterruptedException 类	233
7.63	ObjectStreamException 类	186	8.42	NegativeArraySizeException 类	234
7.64	OptionalDataException 类	186	8.43	NoSuchFieldException 类	234
7.65	StreamCorruptedException 类	186	8.44	NoSuchMethodException 类	234
7.66	SyncFailedException 类	186	8.45	NullPointerException 类	234
7.67	UTFDataFormatException 类	186	8.46	NumberFormatException 类	234
7.68	UnsupportedEncodingException 类	187	8.47	RuntimeException 类	235
7.69	WriteAbortedException 类	187	8.48	SecurityException 类	235
第 8 章 包 Java.lang		188	8.49	StringIndexOutOfBoundsException 类	235
8.1	Cloneable 接口	188	8.50	AbstractMethodError 类	235
8.2	Runnable 接口	188	8.51	CircularityError 类	235
8.3	Boolean 类	188	8.52	ClassFormatError 类	236
8.4	Byte 类	189	8.53	Error 类	236
8.5	Character 类	190	8.54	ExceptionInInitializerError 类	236
8.6	Class 类	194	8.55	IllegalAccessError 类	236
8.7	ClassLoader 类	196	8.56	IncompatibleClassChangeError 类	236
8.8	Compiler 类	198	8.57	InstantiationException 类	237
8.9	Double 类	198	8.58	InternalError 类	237
8.10	Float 类	200	8.59	LinkageError 类	237
8.11	Integer 类	202	8.60	NoClassDefFoundError 类	237
8.12	Long 类	204	8.61	NoSuchFieldError 类	237
8.13	Math 类	206	8.62	NoSuchMethodError 类	237
			8.63	OutOfMemoryError 类	238

8.64	StackOverflowError 类	238	12.2	Key 接口	276
8.65	ThreadDeath 类	238	12.3	Principal 接口	277
8.66	UnknownError 类	238	12.4	PrivateKey 接口	277
8.67	UnsatisfiedLinkError 类	238	12.5	PublicKey 接口	277
8.68	VerifyError 类	238	12.6	DigestInputStream 类	277
8.69	VirtualMachineError 类	239	12.7	DigestOutputStream 类	278
第9章	包 java.lang.reflect	240	12.8	Identity 类	279
9.1	Member 接口	240	12.9	IdentityScope 类	280
9.2	Array 类	240	12.10	KeyPair 类	281
9.3	Constructor 类	244	12.11	KeyPairGenerator 类	282
9.4	Field 类	245	12.12	MessageDigest 类	283
9.5	InvocationTargetException 类	248	12.13	Provider 类	284
9.6	Method 类	248	12.14	SecureRandom 类	285
9.7	Modifier 类	249	12.15	Security 类	285
第10章	包 java.math	251	12.16	Signature 类	286
10.1	BigDecimal 类	251	12.17	Signer 类	289
10.2	BigInteger 类	253	12.18	ProviderException 类	289
第11章	包 java.net	257	12.19	DigestException 类	289
11.1	ContentHandlerFactory 接口	257	12.20	InvalidKeyException 类	289
11.2	FileNameMap 接口	257	12.21	InvalidParameterException 类	290
11.3	SocketImplFactory 接口	257	12.22	KeyException 类	290
11.4	URLStreamHandlerFactory 接口	257	12.23	KeyManagementException 类	290
11.5	ContentHandler 类	257	12.24	NoSuchAlgorithmException 类	290
11.6	DatagramPacket 类	258	12.25	NoSuchProviderException 类	290
11.7	DatagramSocket 类	258	12.26	SignatureException 类	291
11.8	DatagramSocketImpl 类	259	第13章	包 java.security.acl	292
11.9	URLConnection 类	260	13.1	Acl 接口	292
11.10	InetAddress 类	260	13.2	AclEntry 接口	293
11.11	MulticastSocket 类	261	13.3	Group 接口	294
11.12	ServerSocket 类	262	13.4	Owner 接口	294
11.13	Socket 类	263	13.5	Permission 接口	295
11.14	SocketImpl 类	265	13.6	AclNotFoundException 类	295
11.15	URL 类	266	13.7	LastOwnerException 类	295
11.16	URLConnection 类	268	13.8	NotOwnerException 类	295
11.17	URLEncoder 类	273	第14章	包 java.security.interfaces	296
11.18	URLStreamHandler 类	273	14.1	DSAPublicKey 接口	296
11.19	BindException 类	273	14.2	DSAPrivateKey 接口	296
11.20	ConnectException 类	273	14.3	DSAParams 接口	296
11.21	MalformedURLException 类	274	14.4	DSAPrivateKey 接口	297
11.22	NoRouteToHostException 类	274	14.5	DSAPublicKey 接口	297
11.23	ProtocolException 类	274	第15章	包 java.sql	298
11.24	SocketException 类	274	15.1	CallableStatement 接口	298
11.25	UnknownHostException 类	274	15.2	Connection 接口	300
11.26	UnknownServiceException 类	275	15.3	DatabaseMetaData 接口	302
第12章	包 java.security	276	15.4	Driver 接口	317
12.1	Certificate 接口	276	15.5	PreparedStatement 接口	318
			15.6	ResultSet 接口	321

15.7	ResultSetMetaData 接口	325	17.7	Dictionary 类	368
15.8	Statement 接口	327	17.8	EventObject 类	369
15.9	DataTruncation 类	328	17.9	GregorianCalendar 类	369
15.10	Date 类	329	17.10	Hashtable 类	371
15.11	DriverManager 类	330	17.11	ListResourceBundle 类	373
15.12	DriverPropertyInfo 类	331	17.12	Locale 类	373
15.13	SQLException 类	331	17.13	Observable 类	375
15.14	SQLWarning 类	332	17.14	Properties 类	376
15.15	Time 类	332	17.15	PropertyResourceBundle 类	377
15.16	Timestamp 类	333	17.16	Random 类	377
15.17	Types 类	334	17.17	ResourceBundle 类	378
第 16 章	包 java. text	335	17.18	Stack 类	381
16.1	CharacterIterator 接口	335	17.19	StringTokenizer 类	381
16.2	BreakIterator 类	336	17.20	TimeZone 类	382
16.3	ChoiceFormat 类	337	17.21	Vector 类	383
16.4	CollationElementIterator 类	339	17.22	Empty StackException 类	385
16.5	CollationKey 类	339	17.23	MissingResourceException 类	385
16.6	Collator 类	340	17.24	NoSuchElementException 类	385
16.7	DateFormat 类	342	17.25	TooManyListenersException 类	386
16.8	DateFormatSymbols 类	346	第 18 章	包 java. util. zip	387
16.9	DecimalFormat 类	347	18.1	Checksum 接口	387
16.10	DecimalFormatSymbols 类	349	18.2	Adler32 类	387
16.11	FieldPosition 类	350	18.3	CRC32	387
16.12	Format 类	350	18.4	CheckedInputStream 类	387
16.13	MessageFormat 类	351	18.5	CheckedOutputStream 类	388
16.14	NumberFormat 类	352	18.6	Deflater 类	388
16.15	ParsePosition 类	355	18.7	DeflaterOutputStream 类	390
16.16	RuleBasedCollator 类	355	18.8	GZIPInputStream 类	391
16.17	SimpleDateFormat 类	355	18.9	GZIPOutputStream 类	391
16.18	StringCharacterIterator 类	357	18.10	Inflater 类	392
16.19	ParseException 类	358	18.11	InflaterInputStream 类	393
第 17 章	包 java. util	359	18.12	ZipEntry 类	394
17.1	Enumeration 接口	359	18.13	ZipFile 类	395
17.2	EventListener 接口	359	18.14	ZipInputStream 类	395
17.3	Observer 接口	359	18.15	ZipOutputStream 类	396
17.4	BitSet 类	359	18.16	DataFormatException 类	397
17.5	Calendar 类	360	18.17	ZipException 类	397
17.6	Date 类	365			

第 1 章 包 java.applet

1.1 AppletContext 接口

public interface AppletContext

通过该接口可以获得以下信息: Applet 的环境、包含 Applet 的文档、同一文档中的其他 Applets。具体可通过操作该接口中的方法获得。

方 法

getApplet

public abstract Applet getApplet(String name)

参 数 name Applet 名称。

该方法获得与参数 name 关联的 Applet。可以在 HTML 标签中设置以得到参数 name。

返回值 返回指定名称的 Applet, 否则返回 null。

getApplets

public abstract Enumeration getApplets()

获取当前 Applet 上下文所表示文档中的所有 Applet。

返回值 表示所有 applet 的枚举。

GetAudioClip

public abstract AudioClip getAudioClip(URL url)

参 数 url 声音片断所在的绝对 URL 地址。

获取指定 URL 地址处的声音片断。

返回值 返回指定 URL 地址处的声音片断。

GetImage

public abstract Image getImage(URL url)

参 数 url 图像的绝对 URL 地址。

获取可以被显示在屏幕上的图形对象。url 参数必须是一个绝对 URL 地址。

返回值 返回指定 URL 地址处的图形对象。无论该图像是否存在, 本方法总是立即返回。当 Applet 试图显示时, 数据将被加载, 并由图像的图形原语画出图像。

ShowDocument

public abstract void showDocument(URL url)

参 数 url 新主页的 URL 地址。

替换当前正在浏览的主页为指定 URL 地址处的主页。

ShowDocument

public abstract void showDocument(URL url, String target)

参 数 url 文档的 URL 地址。

target 指示主页的显示位置, 它可以取下列值:

“_ self” 在当前框架中显示。

“_ parent” 在父框架中显示。

“_ top” 在最顶层框架中显示。

“_ blank” 在新的最上层无名窗口中显示。

name 在新的最上层名为 name 的窗口中显示。

要求浏览器或 Applet 浏览器显示指定 URL 地址处的主页。

ShowStatus

public abstract void showStatus(String status)

参 数 status 在状态窗口中显示的字符串。

要求在状态窗口中显示参数所指定的字符串 status。许多浏览器或 Applet 浏览器都提供了状态窗口, 使用户可以及时获取当前状态信息。

1.2 AppletStub 接口

public interface AppletStub

当 applet 第一次创建时, 通过 setStub 方法将 applet 承接程序附加给它。承接程序作为 applet 和浏览器或 applet 浏览器环境的接口, 应用程序在该环境中运行。

方 法

appletResize

public abstract void appletResize(int width, int height)

参 数 width applet 的新宽度。

height applet 的新高度。

当 applet 要调整尺寸时调用此方法。

getAppletContext

public abstract AppletContext getAppletContext()

获得 applet 上下文的句柄。

返回值 applet 的上下文。

getCodeBase

public abstract URL getCodeBase()

获得 URL 的地址。

返回值 applet 的 URL 地址。

getDocumentBase

public abstract URL getDocumentBase()

获得文档的 URL 地址。

返回值 包含 applet 的文档的 URL 地址。

getParameter

public abstract String getParameter(String name)

参 数 name 参数名。

返回 HTML 标记中指定参数的值。

返回值 指定参数所对应的值。

isActive

public abstract boolean isActive()

判断当前 applet 是否处于活动状态,当调用 start 方法后,applet 处于活动状态;当调用 stop 方法后,applet 立即处于非活动状态。

返回值 applet 处于活动状态为 True;否则为 False。

1.3 AudioClip 接口

public interface AudioClip

该接口是一个播放声音片断的简单接口,多媒体声音片断可以同时播放,并得到混合声音。

方法

loop

public abstract void loop()

开始循环播放当前声音片断。

play

public abstract void play()

开始播放声音片断。每次调用该方法,声音片断从头开始播放。

stop

public abstract void stop()

停止播放声音片断。

1.4 Applet 类

applet 是一种不能自己单独运行的小程序,它只能被嵌入到其他应用程序中才能运行。

Applet 类是任何嵌入到 Web 页上的 applet 的基类,Applet 类提供了 applet 和环境之间的标准接口。

构造函数

Applet

public Applet()

applet 的默认构造函数。

方法

destroy

public void destroy()

由浏览器或 applet 浏览器调用,用来回收当前 applet 和所占用的资源。调用本方法前通常要先调用 stop 方法。Applet 类的子类如果在被回收之前要执行任何操作,可以对该方法进行重设。例如,有线程的 applet 可以通过 init 方法来创建线程或调用 destroy 方法来终止线程。

参见 init, start, stop。

getAppletContext

public AppletContext getAppletContext()

获得当前 applet 的上下文,它允许 applet 查询和影响其运行的环境。这个环境是指包含当前 applet 的文档。

返回值 当前 applet 的上下文。

getAudioClip

public AudioClip getAudioClip(URL url)

参数 url 指定声音片断的绝对 URL 地址。

返回指定 URL 地址的 AudioClip 对象。无论该声音片断是否存在,该方法都立即返回。当 applet 试图播放时,数据将被加载。

返回值 指定 URL 处的声音片断。

参见 AudioClip。

getAudioClip

public AudioClip getAudioClip(URL url, String name)

参数 url 指定声音片断的绝对 URL 地址。

name 指定声音片断的相对地址,和 url 参数有关。

返回指定 URL 地址的 AudioClip 对象。无论该声音片断是否存在,该方法都立即返回。当 applet 试图播放时,数据将被加载。

返回值 指定 URL 的声音片断。

参见 AudioClip。

getAppletInfo

public String getAppletInfo()

返回当前 applet 的信息。可以通过重设该方法来返回 applet 的作者、版本及版权等信息的字符串。

由 Applet 类提供的该方法的实现返回 null。

返回值 包含 applet 的作者、版本及版权等信息的字符串。

getCodeBase

public URL getCodeBase()

获得 applet 本身的 URL 地址

返回值 当前 applet 的 URL 地址。

参见 getDocumentBase。

getDocumentBase

public URL getDocumentBase()

获得当前 applet 包含的文档的 URL 地址。

返回值 包含当前 applet 文档的 URL 地址。

参见 getCodeBase。

getImage

public Image getImage(URL url)

参数 url 指定位置图像的绝对 URL 地址。

返回可以被显示在屏幕上的图像对象。指定的 url 参数必须是绝对 URL 地址。

无论图像存在与否,该方法都立即返回。当 applet 试图在屏幕上画图像时,数据将被加载,由图形原语在屏幕上画出图像。

返回值 指定 URL 地址处的图像。

参见 Image。

getImage

public Image getImage(URL url, String name)

参数 url 指定位置图像的绝对 URL 地址。

name 图像的相对地址, 相对于 url 参数提供的地址而言。

返回可以被显示在屏幕上的图像对象。指定的 url 参数必须是绝对 URL 地址。

无论图像存在与否, 该方法都立即返回。当 applet 试图在屏幕上画图像时, 数据将被加载, 由图形原语在屏幕上画出图像。

返回值 指定 URL 地址处的图像。

参见 Image。

getLocale

public Locale getLocale()

如果位置已设定, 则返回当前 applet 的位置; 否则, 返回默认位置。

返回值 位置对象。

getParameter

public String getParameter(String name)

参数 name 参数名。

返回 HTML 标记中指定参数的值。例如, 如果当前 applet 为:

```
<applet code="Clock" width=50 height=50>
<param name=Color value="blue">
</applet>
```

则调用 getParameter("Color") 返回值为"blue"。

该 name 参数与大小写无关。

返回值 指定参数的值。

getParameterInfo

public String[][] getParameterInfo()

返回当前参数信息。Applet 可以通过重设本方法来返回描述其参数信息的字符串。每个元素一组三个字符: 名称、类型和描述。

返回值 返回描述当前 applet 参数的字符串数组。

init

public void init()

被浏览器或 applet 浏览器调用, 用来通知当前 applet 已经被加载入系统。通常在 start 方法第一次调用之前调用。Applet 的子类可以重设本方法来实现初始化工作。例如, 使用线程的 applet 可以调用该方法来创建线程或调用 destroy 方法来取消线程。

参见 destroy, start, stop。

isActive

public boolean isActive()

判断当前 applet 是否处于活动状态, 当 start 方法调用后, applet 处于活动状态; 当调用 stop 方法后, applet 立即处于非活动状态。

返回值 applet 处于活动状态为 True; 否则为 False。

参见 start, stop。

play

public void play(URL url)

参数 url 指定声音片断的绝对 URL 地址。

在指定的绝对 URL 地址处播放声音片断。如果找不到该声音片断, 则不发生任何事情。

play

public void play(URL url, String name)

参数 url 声音片断的指定绝对 URL 地址。

name 声音片断的位置, 相对于 url 参数提供的地址而言。

在指定的绝对 URL 地址处播放声音片断。如果找不到该声音片断, 则不发生任何事情。

参见 image。

resize

public void resize(int width, int height)

参数 width 当前 applet 的新宽度。

Height 当前 applet 的新高度。

调整当前 applet 的大小。

该方法重设了类 Component 中的 resize 方法。

resize

public void resize(Dimension d)

参数 d 指定新宽度和高度的对象。

调整当前 applet 的大小。该方法重设了类 Component 中的 resize 方法。

setStub

public final void setStub(AppletStub stub)

参数 stub 新的承接程序。

由系统自动设置 applet 的承接程序。

showStatus

public void showStatus(String msg)

参数 msg 显示在状态窗口中的字符串。

在状态窗口中显示参数字符串。许多浏览器和 applet 浏览器都提供了这样一个窗口使用户知道当前的状态。

start

public void start()

由浏览器或 applet 浏览器调用, 用来通知 applet 开始运行。通常在 init 方法后及每次 applet 重新访问 Web 页时被调用。

Applet 的子类可以通过重设该方法使 applet 每次访问 Web 页时执行特定的操作。例如, 对一个动画 applet, 可以使用 start 方法来重新演示动画, 使用 stop 方法来挂起动画演示。

参见 destroy, init, stop。

stop

public void stop()

由浏览器或 applet 浏览器调用, 用来通知 applet 停止运

行。通常在包含当前 applet 的主页被另一主页替代时及调用 destroy 方法前被调用。Applet 的子类可以通过重设该方法,使每次包含当前 applet 的 Web 页不可见时执行特定的

操作。例如,对一个动画 applet 可以使用 start 方法来重新演示动画;使用 stop 方法来挂起动画演示。

参见 destroy, init。

第2章 包 java.awt

2.1 Adjustable 接口

public interface Adjustable

为在一定边界范围内数值可调整的对象提供的接口。

域

HORIZONTAL

public static final int HORIZONTAL

水平定位。

VERTICAL

public static final int VERTICAL

垂直定位。

方法

addAdjustmentListener

public abstract void addAdjustmentListener(AdjustmentListener l)

参数 l 接受事件的监听器。

当可调整对象的值改变时增加一个监听器来接收可调整事件。

参见 AdjustmentEvent。

getBlockIncrement

public abstract int getBlockIncrement()

获得可调整对象的块值。

getMaximum

public abstract int getMaximum()

获得可调整对象的最大值。

getMinimum

public abstract int getMinimum()

获得可调整对象的最小值。

getOrientation

public abstract int getOrientation()

获得可调整对象的位置。

getUnitIncrement

public abstract int getUnitIncrement()

获得可调整对象的单位增量值。

getValue

public abstract int getValue()

获得可调整对象当前值。

getVisibleAmount

public abstract int getVisibleAmount()

获得比例指示器的长度。

removeAdjustmentListener

public abstract void removeAdjustmentListener (AdjustmentListener l)

参数 l 正在删除的监听器。

删除可调整的监听器。

参见 AdjustmentEvent。

setBlockIncrement

public abstract void setBlockIncrement(int b)

参数 b 块增量。

为可调整对象设置块增量。

setMaximum

public abstract void setMaximum(int max)

参数 max 最大值。

设置可调整对象的最大值。

setMinimum

public abstract void setMinimum(int min)

参数 min 最小值。

设置可调整对象的最小值。

setUnitIncrement

public abstract void setUnitIncrement(int u)

参数 u 单位增量。

设置可调整对象的单位增量值。

setValue

public abstract void setValue(int v)

参数 v 当前值。

设置可调整对象的当前值。这个值必须是在可调整对象的最大和最小值范围之内。

setVisibleAmount

public abstract void setVisibleAmount(int v)

参数 v 指示器的长度。

设置可调整对象的比例指示器的长度。

2.2 ItemSelectable 接口

public interface ItemSelectable

该接口是包含一些项目的对象的接口。

方法

addItemListener

public abstract void addItemListener(ItemListener l)

参数 l 接收事件的监听器。

当一个选项发生改变时,增加一个监听器来接收选项事件。

参见 ItemEvent。

getSelectedObjects

public abstract Object[] getSelectedObjects()

返回选中的选项,没有选项被选中时返回空。

RemoveItemListener

public abstract void removeItemListener(ItemListener l)

参数 l 被删除的监听器。

删除一个监听器。

参见 ItemEvent。

2.3 LayoutManager 接口

public interface LayoutManager

该接口定义了所有布局管理器实现的方法,通过布局管理器可以对容器中的组件进行布局。

参见 Container。

方法

addLayoutComponent

public abstract void addLayoutComponent(String name, Component comp)

参数 name 组件名。

comp 添加的组件。

向布局管理器中添加指定的组件。

layoutContainer

public abstract void layoutContainer(Container parent)

参数 parent 需要进行布局的容器。

在指定的面板上给容器布局。

minimumLayoutSize

public abstract Dimension minimumLayoutSize(Container parent)

参数 parent 被布局的组件。

计算给定容器的布局管理器的最小尺寸。

参见 preferredLayoutSize。

preferredLayoutSize

public abstract Dimension preferredLayoutSize(Container parent)

参数 parent 被布局的组件。

计算给定容器的布局管理器的最佳尺寸。

removeLayoutComponent

public abstract void removeLayoutComponent(Component comp)

参数 comp 被删除的组件。

从布局管理器中删除指定的组件。

参见 minimumLayoutSize。

2.4 LayoutManager2 接口

public interface LayoutManager2 extends LayoutManager

该接口为基于约束性对象的容器布局管理器类的接口,它是 LayoutManager 接口的扩展,为约束性对象提供了怎样把组件添加到布局管理器中的方法。

参见 LayoutManager, Container。

方法

addLayoutComponent

public abstract void addLayoutComponent(Component comp, Object constraints)

参数 comp 添加的组件。

constraints 指明在何处或如何将组件添加到布局管理器中。

用指定的约束对象把指定组件添加到布局管理器中。

getLayoutAlignmentX

public abstract float getLayoutAlignmentX(Container target)

返回按 x 轴的对齐值,主要指定当前组件和其他组件的排列对齐关系。

返回值 是一个在 0 到 1 之间的值,0 表示按原来位置对齐,1 表示离原来位置最远,0.5 表示在中间。

GetLayoutAlignmentY

public abstract float getLayoutAlignmentY(Container target)

返回按 y 轴的对齐值,主要指定当前组件和其他组件的排列对齐关系。

返回值 是一个在 0 到 1 之间的值,0 表示按原来位置对齐,1 表示离原来位置最远,0.5 表示在中间。

invalidateLayout

public abstract void invalidateLayout(Container target)

使现有的布局无效。

maximumLayoutSize

public abstract Dimension maximumLayoutSize(Container target)

返回当前组件的最大值。

参见 getMinimumSize, getPreferredSize, LayoutManager。

2.5 MenuContainer 接口

public interface MenuContainer

该接口是所有和菜单相关的容器的基类,它指定了所有与菜单相关的容器都应实现的方法。

方法

getFont

public abstract Font getFont()

获得字体。

postEvent

public abstract boolean postEvent(Event evt)

参数 evt 被传递的事件。

向当前菜单容器传递一个事件。

remove

public abstract void remove(MenuComponent comp)

参数 comp 被删除的菜单组件。

从当前菜单容器中删除指定菜单组件。

2.6 PrintGraphics 接口

public interface PrintGraphics

为页面提供打印图形上下文的抽象类。

方法

getPrintJob

public abstract PrintJob getPrintJob()

返回生成当前 PrintGraphics 对象的 PrintJob 对象。

2.7 Shape 接口

public interface Shape

为表示某种形状的几何图形对象提供的接口。

该接口将在 Java2D 项目中进行修订。这意味着为各种现存的几何 AWT 类及其方法提供一种通用的接口。既然该接口将来要扩充, 研究人员应在该接口完成之前, 避免在自身的类中实现它。

方法

getBounds

public abstract Rectangle getBounds()

返回图形的边框。

2.8 AWTEvent 类

public abstract class AWTEvent extends EventObject

该类是所有 AWT 事件类的根事件类, 它的子类是在 java.awt.event 包的外部定义的, 定义的事件 ID 值能够比 RESERVED_ID_MAX 定义的值更大。该类中的事件屏蔽只为组件的子类所用, 通过 Component.enableEvents() 来选择那些没有被注册的监听器选中的事件类型。如果监听器已经对一个组件进行了注册, 那么事件屏蔽就由组件在内部自动进行了设定。

参见 enableEvents, ComponentEvent, FocusEvent, KeyEvent, MouseEvent, WindowEvent, ActionEvent, AdjustmentEvent, ItemEvent, TextEvent。

域

ACTION_EVENT_MASK

public static final long ACTION_EVENT_MASK

屏蔽选择的动作事件。

ADJUSTMENT_EVENT_MASK

public static final long ADJUSTMENT_EVENT_MASK

对可调整事件的屏蔽。

COMPONENT_EVENT_MASK

public static final long COMPONENT_EVENT_MASK

对选择组件事件的屏蔽。

CONTAINER_EVENT_MASK

public static final long CONTAINER_EVENT_MASK

对选择容器事件的屏蔽。

FOCUS_EVENT_MASK

public static final long FOCUS_EVENT_MASK

对焦点选择事件的屏蔽。

ITEM_EVENT_MASK

public static final long ITEM_EVENT_MASK

对选项选择事件的屏蔽。

KEY_EVENT_MASK

public static final long KEY_EVENT_MASK

对选择键事件的屏蔽。

MOUSE_EVENT_MASK

public static final long MOUSE_EVENT_MASK

对选择鼠标事件的屏蔽。

MOUSE_MOTION_EVENT_MASK

public static final long MOUSE_MOTION_EVENT_MASK

对选择鼠标移动事件屏蔽。

RESERVED_ID_MAX

public static final int RESERVED_ID_MAX

保留 AWT 事件 ID 值的最大值, 程序中用户自己定义的事件 ID 值应该大于这个值。

TEXT_EVENT_MASK

public static final long TEXT_EVENT_MASK

对选择文本事件的屏蔽。

WINDOW_EVENT_MASK

public static final long WINDOW_EVENT_MASK

对选择窗口事件的屏蔽。

构造函数

AWTEvent

public AWTEvent(Event event)

参数 event 事件。

1.0-style 事件的参数中构造一个 AWTEvent 对象。

AWTEvent

public AWTEvent(Object source, int id)

参数 source 生成事件的对象。

用指定源对象来构造 AWTEvent 对象。