Java 常用方法大全

文章整理: www.diybl.com 文章来源: 网络 去论坛 建我的blog

字符串

1、获取字符串的长度

length()

2 、判断字符串的前缀或后缀与已知字符串是否相同

前缀 startsWith(String s)

后缀 endsWith(String s)

3、比较两个字符串

equals(String s)

4、把字符串转化为相应的数值

int 型 Integer.parseInt(字符串)

long 型 Long.parseLong(字符串)

float 型 Folat.valueOf(字符串).floatValue()

double 型 Double.valueOf(字符串).doubleValue()

4、将数值转化为字符串

valueOf(数值)

5、字符串检索

indexOf(Srting s) 从头开始检索

indexOf(String s ,int startpoint) 从 startpoint 处开始检索

如果没有检索到，将返回-1

6、得到字符串的子字符串

substring(int startpoint) 从 startpoint 处开始获取

substring(int start,int end) 从 start 到 end 中间的字符

7、替换字符串中的字符,去掉字符串前后空格

replace(char old,char new) 用 new 替换 old

trim()

8、分析字符串

StringTokenizer(String s) 构造一个分析器，使用默认分隔字符（空格，换行，回车，Tab，进纸符）

StringTokenizer(String s,String delim) delim 是自己定义的分隔符

nextToken() 逐个获取字符串中的语言符号

boolean hasMoreTokens() 只要字符串还有语言符号将返回 true，否则返回 false

countTokens() 得到一共有多少个语言符号

文本框和文本区

1、文本框

TextField() 构造文本框，一个字符长

TextField(int x) 构造文本框，x 个字符长

TextField(String s) 构造文本框，显示 s

setText(String s) 设置文本为 s

getText() 获取文本

setEchoChar(char c) 设置显示字符为 c

setEditable(boolean) 设置文本框是否可以被修改

addActionListener() 添加监视器

removeActionListener() 移去监视器

2、文本区
TextArea() 构造文本区
TextArea(String s) 构造文本区，显示 s
TextArea(String s,int x,int y) 构造文本区，x 行，y 列，显示 s
TextArea(int x,int y) 构造文本区，x 行，y 列
TextArea(String s,int x,ing y,int scrollbar)
scrollbar 的值是：
TextArea.SCROLLBARS_BOTH
TextArea.SCROLLBARS_VERTICAL_ONLY
TextArea.SCROLLBARS_HORIZONTAL_ONLY
TextArea.SCROLLBARS_NONE
setText(String s) 设置文本为 s
getText() 获取文本
addTextListener() 添加监视器
removeTextListener() 移去监视器
insert(String s,int x) 在 x 处插入文本 s
replaceRange(String s,int x,int y) 用 s 替换从 x 到 y 处的文本
append(String s) 在文本的最后追加文本 s
Int getCaretPosition(int n) 获取文本区中光标的位置

按钮

1、按钮
Button() 构造按钮
Button(String s) 构造按钮，标签是 s
setLabel(String s) 设置按钮标签是 s
getLabel() 获取按钮标签
addActionListener() 添加监视器
removeActionListener() 移去监视器

标签

1、标签
Label() 构造标签
Label(String s) 构造标签，显示 s
Label(String s,int x)
x 是对齐方式，取值：
Label.LEFT
Label.RIGHT
Label.CENTER
setText(String s) 设置文本 s
getText() 获取文本
setBackground(Color c) 设置标签背景颜色
setForeground(Color c) 设置字体颜色

选择框

1、选择框
Checkbox() 构造选择框
Checkbox(String s) 构造选择框，给定标题 s
Checkbox(String s,boolean b) b 设定初始状态
Checkbox(String s,boolean b,CheckboxGroup g) g 设定了所属的组（有了组就成为单选框）
addItemListener() 添加监视器
removeItemListener() 移去监视器
getState() 返回选择框的是否选中状态
setState(boolean b) 设置选择框的状态
getLabel() 获取选择框的标题
setLabel(String s) 设置选择框的标题为 s

选择控件和滚动列表

1、选择控件
Choice() 构造选择控件
add(String s) 向选择控件增加一个选项
addItemListener() 添加监视器
removeItemListener() 移去监视器
getSelectedIndex() 返回当前选项的索引
getSelectedItem() 返回当前选项的字符串代表
insert(String s,int n) 在 n 处插入选项 s
remove(int n)
removeAll()
2、滚动列表
List() 构造滚动列表
List(int n) 参数 n 是可见行数
List(int n,boolean b) 参数 b 是设置是否可以多项选择
add(String s) 向列表的结尾增加一个选项
add(String s,int n) 在 n 处增加一个选项
AddActionListener() 滚动列表添加监视器
addItemListener() 滚动列表上的选项添加监视器
remove(int n) 删除 n 初的选项
remnoveAll() 删除全部选项
getSelectedIndex() 返回当前选项的索引
getSelectedItem() 返回当前选项的字符串代表

3、组件类的一些常用方法
void setBackground(Color c) 设置组件背景颜色
void setForeground(Color c) 设置组件前景颜色
void setFonts(Font f) 设置组件字体
void setBounds(int x,int y,int w,int h) 设置坐标，x，y 表示在容器中坐标，w,h 表示宽和高
void setLocation(int x,int y) 移动到 x，y 处
void setSize(int w,int h) 设置宽和高
void setVisible(boolean b) 设置组建是否可见
int getBounds().wigth 获取宽
int getBounds().height 获取高

int getBounds().x 获取 x 坐标
int getBounds().y 获取 y 坐标
Toolkit getToolkit() 获取工具包对
void setEnabled(boolean b) 设置是否可以使用（默认可以）

窗口和菜单
1、窗口
Frame() 构造窗口
Frame(String s) 窗口标题是 s
setBounds(int x,int y,int w,int h) 窗口位置 x，y，宽 w,高 y
setSize(int w,int h) 设置窗口位置（单位是像素）
setBackground(Color c) 设置背景颜色
setVisible(boolean b) 设置窗口是否可见
pack() 窗口出现时紧凑
setTitle(String s) 设置标题为 s
getTitle() 获取标题
setResizable(boolean b) 设置窗口大小是否可以调整
2、菜单条
Menubar() 构造菜单条
setMenubar() 窗口添加菜单条
3、菜单
Menu() 构造菜单
Menu(String s) 构造菜单，标题 s
add
add(MenuItem item) 菜单增加菜单选项 item
add(String s) 向菜单增加选项 s
getItem(int n) 获取 n 处的选项
getItemCount() 获取选项数目
insert(MenuItem item,int n) 在 n 处插入菜单选项 item
insert(String s,int n) 在 n 处插入菜单选项
remove(int n) 删除菜单的 n 处的菜单选项
removeAll() 删除全部
4、菜单项
MenuItem() 构造菜单项
MenuItem(String s) 构造标题是 s 的菜单项
setEnabled(boolean b) 设置是否可以被选择
getLabel() 得到菜单选项名
addActionListener() 添加监视器
5、有关菜单的技巧
addSeparator() 增加菜单分割线
CheckboxMenuItem() 复选框菜单项
setShortcut(MenuShortcut k) 设置快捷键(k 取值 KeyEvent.VK_A----KeyEvent.VK_Z)

建立对话框
1、Dialog 类
Dialog(Frame f,String s) 构造对话框，初始不可见，s 是标题，f 是对话框所依赖的窗口
Dialog(Frame f,String s,boolean b) b 设置初始是否可见

getTitle() 获取对话框标题
setTitle(String s) 设置对话框标题
setModal(boolean b) 设置对话框模式
setSize(int w,int h) 设置对话框大小
setVisible(boolean b) 显示或隐藏对话框
2、FileDialog 类
Filedialog(Frame f,String s,int mode) mode 的值是 fileDialog.LOAD 或者 fileDialog.SAVE
public String getDirectory() 获取当前文件对话框中显示的文件所属目录
public String getFile() 获取当前文件对话框中文件的字符串表示，不存在返回 null

Java 中的鼠标和键盘事件
1、使用 MouseListener 借口处理鼠标事件
鼠标事件有 5 种：按下鼠标键，释放鼠标键，点击鼠标键，鼠标进入和鼠标退出
鼠标事件类型是 MouseEvent，主要方法有：
getX(),getY() 获取鼠标位置
getModifiers() 获取鼠标左键或者右键
getClickCount() 获取鼠标被点击的次数
getSource() 获取鼠标发生的事件源
事件源获得监视器的方法是 addMouseListener()，移去监视器的方法是 removeMouseListener()
处理事件源发生的时间的事件的接口是 MouseListener 接口中有如下的方法
mousePressed(MouseEvent) 负责处理鼠标按下事件
mouseReleased(MouseEvent) 负责处理鼠标释放事件
mouseEntered(MouseEvent) 负责处理鼠标进入容器事件
mouseExited(MouseEvent) 负责处理鼠标离开事件
mouseClicked(MouseEvent) 负责处理点击事件
2、使用 MouseMotionListener 接口处理鼠标事件
事件源发生的鼠标事件有 2 种：拖动鼠标和鼠标移动
鼠标事件的类型是 MouseEvent
事件源获得监视器的方法是 addMouseMotionListener()
处理事件源发生的事件的接口是 MouseMotionListener 接口中有如下的方法
mouseDragged() 负责处理鼠标拖动事件
mouseMoved() 负责处理鼠标移动事件
3、控制鼠标的指针形状
setCursor(Cursor.getPreddfinedCursor(Cursor.鼠标形状定义)) 鼠标形状定义见（书 P 210）
4、键盘事件
键盘事件源使用 addKeyListener 方法获得监视器
键盘事件的接口是 KeyListener 接口中有 3 个方法
public void keyPressed(KeyEvent e) 按下键盘按键
public void keyReleased(KeyEvent e) 释放键盘按键
public void keyTypde(KeyEvent e) 按下又释放键盘按键

Java 多线程机制
1、Java 的线程类与 Runnable 接口
Thread 类
public Thread() 创建线程对象
public Thread(Runnable target) target 称为被创建线程的目标对象，负责实现 Runnable 接口
线程优先级
Thread 类有三个有关线程优先级的静态常量：MIN_PRIORITY,MAX_PRIORITY,NORM_PRIORITY

新建线程将继承创建它的副相承的优先级，用户可以调用 Thread 类的 setPriority(int a)来修改
a 的取值:
Thread.MIN_PRIORITY,Thread.MAX_PRIORITY,Thread.NORM_PRIORITY
主要方法
启动线程 start()
定义线程操作 run()
使线程休眠 sleep()
sleep(int millsecond) 以毫秒为单位的休眠时间
sleep(int millsecond,int nanosecond) 以纳秒为单位的休眠时间
currentThread() 判断谁在占用 CPU 的线程
第二十章 输入输出流
1、FileInputStream 类
FileInputStream(String name) 使用给定的文件名 name 创建一个 FileInputStream 对象
FileInputStream(File file) 使用 File 对象创建 FileInpuStream 对象
File 类有两个常用方法：
File(String s) s 确定文件名字
File(String directory,String s) directory 是文件目录
例如：

```
File f=new File("Myfile.dat");
FileInputStream istream=new FileInputStream(f);
```

处理 I/O 异常
当出现 I/O 错误的时候，Java 生成一个 IOException(I/O 异常)对象来表示这个错误的信号。
程序必须使用一个 catch 检测这个异常
例如：

```
try{
FileInputStream ins= new FileInputStream("Myfile.dat");
}
catch(IOException e){
System.out.println("File read Error:"+e);
}
```

从输入流中读取字节
int read() 返回 0~255 之间一个整数，如果到输入流末尾，则返回-1
int read(byte b[]) 读取字节数组
int read(byte b[],int off,int len) off 指定把数据存放在 b 中什么地方，len 指定读取的最大字节数
关闭流
close()
2、FileOutputStream 类
FileOutputStream(String name) 使用指定的文件名 name 创建 FileOutputStream 对象
FileOutputStream（File file） 使用 file 对象创建 FileOutputStream 对象
FileOutputStream（FileDescriptor fdobj） 使用 FileDescriptor 对象创建 FileOutputStream 对象
3、FileReader 类和 FileWriter 类
FileReader(String filename)
FileWriter(String filename)
处理时需要 FileNotFoundException 异常
4、RandomAccessFile 类
RandomAccessFile 不同于 FileInputStream 和 FileOutputStream,不是他们的子类
当我们想对一个文件进行读写操作的时候，创建一个指向该文件的 RandomAccessFile 流就可以了
RandomAccessFile 类有两个构造方法：

RandomAccessFile（String name, String mode） name 是文件名，mode 取 r(只读)或 rw(读写)
RandomAccessFile（File file,String mode） file 给出创建流的源
seek(long a) 移动 RandomAccessFile 流指向文件的指针,a 确定指针距文件开头的位置
getFilePointer() 获取当前文件的指针位置
close() 关闭文件
getFD() 获取文件的 FileDescriptor
length() 获取文件长度
read() 读取一个字节数据
readBoolean() 读取一个布尔值
readByte() 读取一个字节
readChar()
readFloat()
readFully(byte b[])
readInt()
readLine()
readLong()
readUnsignedShort()
readUTF() 读取一个 UTF 字符串
setLength(long newLength) 设置文件长度
skipByte(int n) 在文件中跳过给定数量的字节
write(byte b[]) 写 b.length 个字节到文件
writeBoolean(bolean b)
writeByte(int v)
writeChar(char c)
writeChars(String s)
writeDouble(double d)
writeFloat(float v)
writeInt(int i)
writeLong(long l)
writeShort(int i)
writeUTF(String s)
5、管道流
PipedInputStream 类
PipedInputStream() 创建一个管道输入流
PipedInputStream（PipedOutputStream a） 连接到输出流 a 的输入流
read() 从输入流中读取一个字节
read(byte b[],int off,int len) off 是在 b 中的开始位置，len 是字节长度
PipedOutputStream 类
PipedOutputStream() 创建一个输出流
PipedOutputStream(PipedInputStream a) 连接到输入流 a 的输出流
write(int b)
write(byte b[],int off,int len)
counnect() 连接输入输出流
close() 关闭流
在使用的时候要捕获 IOException 异常。
6、数据流
DataInputStream 类(数据输入流)
DataInputStream(InputStream in) 将数据输入流指向一个由 in 指定的输入流

DataOutputStream 类(数据输出流)
DataOutputStream(OutputStream out) 将数据输出流指向一个由 out 指定的输出流
主要方法：
close()
read() 读取一个字节数据
readBoolean() 读取一个布尔值
readByte() 读取一个字节
readChar()
readFloat()
readFully(byte b[])
readInt()
readLine()
readLong()
readUnsignedShort()
readUTF() 读取一个 UTF 字符串
skipByte(int n) 在文件中跳过给定数量的字节
write(byte b[]) 写 b.length 个字节到文件
writeBoolean(bolean b)
writeByte(int v)
writeChar(char c)
writeChars(String s)
writeDouble(double d)
writeFloat(float v)
writeInt(int i)
writeLong(long l)
writeShort(int i)
writeUTF(String s)
7、对象流
ObjectInputStream 类和 ObjectOutputStream 类分别是 DataInputStream 类和 DataOutputStream 类的子类
8、回压输入流
PushbackInputStream 类
PushbackInputStream(InputStream in)
PushbackReader 类
PushbackReader(Reader in)
unread(char c) 回压一个字符
unread(char c[]) 回压数组 c 中全部字符
unread(char c[],offset,int n) 回压 c 中从 offset 开始的 n 个字符

java 网络的基本知识
1、使用 URL（统一资源定位）
例如：

```
try
{
url=new URL("http://www.sina.com.cn";
}
catch(MalformedURLException e)
{
System.out.println("Bad URL:"+url);
```

}
在 Applet 中链接向另外的 Web 页面，使用代码：
getAppletContext().showDocument(url);
2、套接字
客户建立到服务器的套接字（Socket）
Socket(String host,int port) host 是服务器的 IP 地址，port 是端口号
建立了套接字后可以使用 getInputStream()获得输入流
还可以使用 getOutputStream()获得一个输出流
服务器建立接受客户套接字的服务器套接字（ServerSocket）
ServerSocket(int port) port 是端口号
建立了套接字连接后可以使用 accept()接收客户的套接字
可以使用 getOutputStream()获得一个输出流
还可以使用 getInputStream()获得一个输入流
3、InetAddress 类
getByName(String)
3、InetAddress 类
getByName(String s) 获取 Internet 上主机的地址
getHostName() 获取 InetAddress 对象所包含的域名
getHostAddress() 获取 InetAddress 对象所包含的 IP 地址
getLocalHost() 获取本地地址
4、UDP 数据报
发送数据包，即使用 DatagramPacket 类将数据打包,有两种构造方法
DatagramPacket(byte data[],int length,InetAddress address,int port)
?含有 data 数组的数据
?该数据包将发送到地质是 address，端口是 port 的主机上
DatagramPacket(byte data[],int offset,int length,InetAddress address,int port)
?含有 data 数组的从 offset 开始，length 长度的数据
?该数据包将发送到地质是 address，端口是 port 的主机上
接收数据包，即使用 DatagramSocket（int port）创建一个对象，port 必须和待接收的数据包的端口相同
例如：
如果发送方的数据包端口是 5566
DatagramSocket mail=new DatagramSocket(5566);
然后对象 mail 可以使用方法 receive（Datagrampacket pack）接收数据包
在使用参数 pack 接收数据包前，必须创建 pack
byte data[]=new byte[100];
int length=90;
DatagramPacket pack=new DatagramPataet(data,length);
mail.receive(pack);
该数据包 pack 将接收长度是 length 的数据放入 data,注意数据长度不要超过 8192KB
pack 还可以调用方法 getPort()获取所接受数据包是从什么端口发出的
调用方法 InetAddress getAddress()可以获知数据包来自哪个主机

Java 与图像
1、java 支持的图像类型：GIF，JPEG，BMP
2、Image 类
首先申请一个 Image 对象
Image img =getImage(URL url,String name) url 是图像地址，name 是图片名称
通常使用：

Image img =getImage(getCodBase(),String name) getCodBase()获取当前小应用程序的 URL，也就是在同一目录下
图像被加载后，就可以在 paint()中绘制了
drawImage(Image img,int x,int y,ImageObserver observer)
img 是上面获取的图像， x,y 是指定图像左上角的位置，observer 是加载图像时的图像观察器
Applet 类已经实现了 ImageObserver 接口，所以可以直接使用 this 作为最后一个参数
drawImage(Image img,int x,int y,int width,int height,ImageObserver observer)
width 和 height 是要绘制的图像的宽和高
可以使用 img.getHeight(this)和 img.getWidth(this)来获取被加载的图像的宽和高
3、设置 Java 窗口图标
Frame 对象可以使用 setIconImage(Image img)方法设置左上角图标，默认图标是咖啡杯

Java 数据库连接(JDBC)
1、JDBC-ODBC 桥接器
建立 JDBC-ODBC 桥接器
try
{
Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
}
catch(ClassNotFoundException e){}
连接到数据库
try
{
Connection con=DriverManager.getConnection("jdbcdbc:数据源名称","数据源的 login name",
"数据源的 password");
}
catch(SQLException e){}
向数据库发送 SQL 语句
try
{
Statement sql=con.createStatement();
}
catch(SQLException e){}
处理查询结果
ResultSet rs=sql.executeQuery("SQL 语句");
第二十四章 Java 与多媒体
1、在小程序中播放声音
java 可以播放 au,aiff,wav,midi,rfm 格式的音频
可以调用 Applet 的一个静态方法：
newAudioClip(URL url,String name) url 是地址，name 是音频名称
也可以用 Applet 类的实例方法：
getAudioClip(URL url,String name)
根据 url 地址和声音文件 name，获得一个用于播放的音频对象，这对象可以使用下面的方法来处理声音：
play() 播放声音文件 name
loop() 循环播放 name
stop() 停止播放 name
2、Java 媒体框架(JMF)
创建播放器
try

```
{
URL url=new URL(getDocumenBase(),视频文件名称);
player player=Manager.createPlayer(url);
}
catch(IOException e){}
```

向播放器注册控制监视器

```
player.addControllerListener(监视器);
```

创建监视器必须使用接口 ControllerListener ， 该接口中的方法是

```
public void controllerUpdate(ControllerEvent e)
```

让播放器对媒体进行预提取

```
player.prefetch()
```

启动播放器

```
player.start();
```

停止播放器

```
player.stop();
```

停止播放器后必须释放内存中的资源

```
player.deallocate();
```

Java Swing 基础

1、Jcomponent 类

Jcomponent 类 是所有轻量组件的父类，主要的子类有：

JButton 创建按钮对象，而且可以创建在图标的按钮

JComboBox 创建组合框对象，和 Choice 相似

JCheckBox 创建复选框对象

JFileChooser 创建文件选择器

JInternalFrame 创建内部窗体

JLabel 创建标签

JMenu 创建菜单对象

JMenuBar 创建菜单条对象

JMenuItem 创建菜单项对象

JPanel 创建面板对象

JPasswordField 创建口令文本对象

JPopupMenu 创建弹出式菜单

JProgressBar 创建进程条

JRadioButton 创建单选按钮

JScrollBar 创建滚动条

JScrollPane 创建滚动窗格

JSlider 创建滚动条

JSplitPane 创建拆分窗格

JTable 创建表格

JTextArea 创建文本区

JTexPane 创建文本窗格

JToolBar 创建工具条

JToolTip 创建工具提示对象

JTree 创建树对象

2、JFrame 类

JFrame 类及其子类创建的对象是窗体

（1）JFrame 类及其子类创建的窗体是 swing 窗体

（2）不可以把组件直接加到 swing 窗体中，应该把组件加到内容面板中
（3）不能为 swing 窗体设置布局，而应当为内容面板设置布局
（4）swing 窗体通过调用 getContentPane()方法得到它的内容面板
3、JApplet 类
（1）不可以把组件直接添加到小程序容器中，也应该添加到内容面板中
（2）不能为小程序设置布局
（3）小程序容器通过调用 getContenPane()方法得到内容面板
4、JDialog 类
（1）不可以把组件直接添加到对话框容器中，也应该添加到内容面板中
（2）不能为对话框设置布局
（3）对话框容器通过调用 getContenPane()方法得到内容面板
5、JPanel 面板
JPanel()
JPanel(布局对象)
6、滚动窗口 JScrollPane
JScrollPane()
JScrollPane(component c)
7、拆分窗口 JSplitPane
JSplitPane(int a,Component b,Component c)
a 的取值是 HORIZONTAL_SPLIT 或者 VERTICAL_SPLIT 决定水平拆分还是垂直拆分
JSplitPane(int a,boolean b ,Component b,Component c) b 的取值决定拆分线移动的时候组件是否连续变化
8、内部窗体 JInternalFrame
JInternalFrame(String title,boolean resizable,booleam closable,boolean max,boolean min)
参数的意义分别是窗口名称，是否能调整大小，是否有关闭按钮，最大化按钮，最小化按钮
（1）不能把组件直接加到窗体中，而是加到内容面板中
（2）必须先把内部窗体加到一个容器中（JDesktopPane），该容器是专门为内部窗体服务的
9、按钮（JButton）
JButton() 创建按钮
JButton(String s) s 是按钮的名字
JButton(Icon icon) icon 是按钮上的图标
JButton(String s,Icon icon)
getText() 获取按钮名字
getIcon() 获取按钮图标
setIcon(Icon icon) 设置按钮图标
setHorizontalTextposition(int a) a 确定按钮上图标的位置，取值：
AbstractButton_CENTR,AbstractButton_LEFT,AbstractButton_RIGHT
setVerticalTextposition(int a) a 确定按钮上名字相对图标的位置，取值：
AbstractButton.TOP，AbstractButton.BOTTOM，AbstractButton.CENTR
setMnemonic(char c) 设置按钮的键盘操作方式是字符 c（Alt+c）
setEnabled(boolean b) b 决定按钮是否可以被单击