import java.awt.alphacomposite;  
import java.awt.color;  
import java.awt.font;  
import java.awt.graphics2d;  
import java.awt.image;  
import java.awt.geom.affinetransform;  
import java.awt.image.affinetransformop;  
import java.awt.image.bufferedimage;  
import java.io.file;  
import java.io.ioexception;  
import javax.imageio.imageio;  
/\*\*  
 \* @author eric xu  
 \*  
 \*/  
public final class imageutils {  
 /\*\*  
  \* 图片水印  
  \* @param pressimg 水印图片  
  \* @param targetimg 目标图片  
  \* @param x 修正值 默认在中间  
  \* @param y 修正值 默认在中间  
  \* @param alpha 透明度  
  \*/  
 public final static void pressimage(string pressimg, string targetimg, int x, int y, float alpha) {  
  try {  
   file img = new file(targetimg);  
   image src = imageio.read(img);  
   int wideth = src.getwidth(null);  
   int height = src.getheight(null);  
   bufferedimage image = new bufferedimage(wideth, height, bufferedimage.type\_int\_rgb);  
   graphics2d g = image.creategraphics();  
   g.drawimage(src, 0, 0, wideth, height, null);  
   //水印文件  
   image src\_biao = imageio.read(new file(pressimg));  
   int wideth\_biao = src\_biao.getwidth(null);  
   int height\_biao = src\_biao.getheight(null);  
   g.setcomposite(alphacomposite.getinstance(alphacomposite.src\_atop, alpha));  
   g.drawimage(src\_biao, (wideth - wideth\_biao) / 2, (height - height\_biao) / 2, wi             deth\_biao, height\_biao, null);  
   //水印文件结束  
   g.dispose();  
   imageio.write((bufferedimage) image, "jpg", img);  
  } catch (exception e) {  
   e.printstacktrace();  
  }  
 }  
 /\*\*  
  \* 文字水印  
  \* @param presstext 水印文字  
  \* @param targetimg 目标图片  
  \* @param fontname 字体名称  
  \* @param fontstyle 字体样式  
  \* @param color 字体颜色  
  \* @param fontsize 字体大小  
  \* @param x 修正值  
  \* @param y 修正值  
  \* @param alpha 透明度  
  \*/  
 public static void presstext(string presstext, string targetimg, string fontname, int fontstyle,    color color, int fontsize, int x, int y, float alpha) {  
  try {  
   file img = new file(targetimg);  
   image src = imageio.read(img);  
   int width = src.getwidth(null);  
   int height = src.getheight(null);  
   bufferedimage image = new bufferedimage(width, height, bufferedimage.type\_int\_rgb);  
   graphics2d g = image.creategraphics();  
   g.drawimage(src, 0, 0, width, height, null);  
   g.setcolor(color);  
   g.setfont(new font(fontname, fontstyle, fontsize));  
   g.setcomposite(alphacomposite.getinstance(alphacomposite.src\_atop, alpha));  
   g.drawstring(presstext, (width - (getlength(presstext) \* fontsize)) / 2 + x, (he             ight - fontsize) / 2 + y);  
   g.dispose();  
   imageio.write((bufferedimage) image, "jpg", img);  
  } catch (exception e) {  
   e.printstacktrace();  
  }  
 }  
 /\*\*  
  \* 缩放  
  \* @param filepath 图片路径  
  \* @param height 高度  
  \* @param width 宽度  
  \* @param bb 比例不对时是否需要补白  
  \*/  
 public static void resize(string filepath, int height, int width, boolean bb) {  
  try {  
   double ratio = 0.0; //缩放比例   
   file f = new file(filepath);  
   bufferedimage bi = imageio.read(f);  
   image itemp = bi.getscaledinstance(width, height, bi.scale\_smooth);  
   //计算比例  
   if ((bi.getheight() > height) || (bi.getwidth() > width)) {  
    if (bi.getheight() > bi.getwidth()) {  
     ratio = (new integer(height)).doublevalue() / bi.getheight();  
    } else {  
     ratio = (new integer(width)).doublevalue() / bi.getwidth();  
    }  
    affinetransformop op = new                affinetransformop(affinetransform.getscaleinstance(ratio, ratio), null);  
    itemp = op.filter(bi, null);  
   }  
   if (bb) {  
    bufferedimage image =   
                 new bufferedimage(width, height, bufferedimage.type\_int\_rgb);  
    graphics2d g = image.creategraphics();  
    g.setcolor(color.white);  
    g.fillrect(0, 0, width, height);  
    if (width == itemp.getwidth(null))  
     g.drawimage(itemp, 0, (height - itemp.getheight(null)) / 2,                   itemp.getwidth(null), itemp.getheight(null), color.white, null);  
    else  
  g.drawimage(itemp, (width - itemp.getwidth(null)) / 2, 0, itemp.getwidth(null), i                    temp.getheight(null), color.white, null);  
    g.dispose();  
    itemp = image;  
   }  
   imageio.write((bufferedimage) itemp, "jpg", f);  
  } catch (ioexception e) {  
   e.printstacktrace();  
  }  
 }  
 public static void main(string[] args) throws ioexception {  
  pressimage("g:imgtestsy.jpg", "g:imgtesttest1.jpg", 0, 0, 0.5f);  
  presstext("我是文字水印", "g:imgtesttest1.jpg", "黑体", 36, color.white, 80, 0, 0, 0.3f);  
  resize("g:imgtesttest1.jpg", 500, 500, true);  
 }  
 public static int getlength(string text) {  
  int length = 0;  
  for (int i = 0; i < text.length(); i++) {  
   if (new string(text.charat(i) + "").getbytes().length > 1) {  
    length += 2;  
   } else {  
    length += 1;  
   }  
  }  
  return length / 2;  
 }  
}

用过滤器,等我上线给你,不要结问题哦   
package com.jc.ts.services;   
  
import java.awt.Color;   
import java.awt.Graphics;   
import java.awt.Image;   
import java.awt.image.BufferedImage;   
import java.io.FileInputStream;   
import java.io.FileNotFoundException;   
import java.io.IOException;   
  
import javax.imageio.ImageIO;   
import javax.servlet.http.HttpServletRequest;   
import javax.servlet.http.HttpServletResponse;   
  
import com.sun.image.codec.jpeg.ImageFormatException;   
import com.sun.image.codec.jpeg.JPEGCodec;   
import com.sun.image.codec.jpeg.JPEGImageEncoder;   
  
/\*\*   
\*添加图片[水印](http://www.so.com/s?q=%E6%B0%B4%E5%8D%B0&ie=utf-8&src=internal_wenda_recommend_textn" \t "https://wenda.so.com/q/_blank)的服务类   
\* \*/   
public class WaterMark {   
/\*\*   
\* @param sizeContext添加水印[文字](http://www.so.com/s?q=%E6%96%87%E5%AD%97&ie=utf-8&src=internal_wenda_recommend_textn" \t "https://wenda.so.com/q/_blank)   
\* @param request 请求流对象   
\* @param request 响应流对象   
\* \*/   
@SuppressWarnings("deprecation")   
public static void createMarkSize(String sizeContext,HttpServletRequest request,HttpServletResponse response) {   
try {   
String path=request.getRealPath(request.getServletPath());   
FileInputStream in=new FileInputStream(path);   
Image src=ImageIO.read(in);   
int w=src.getWidth(null);   
int h=src.getHeight(null);   
BufferedImage img=new BufferedImage(w,h,BufferedImage.TYPE\_INT\_RGB);//构建画板   
Graphics g=img.getGraphics();//得到画笔   
g.drawImage(src,0,0,w,h,null);//把源图片写入画板   
g.setColor(Color.red);   
g.drawString(sizeContext,10,5); // 添加文字   
g.dispose();//生成图片   
JPEGImageEncoder e=JPEGCodec.createJPEGEncoder(response.getOutputStream());   
e.encode(img);   
response.getOutputStream().close();   
} catch (FileNotFoundException e) {   
// TODO Auto-generated catch block   
e.printStackTrace();   
} catch (ImageFormatException e) {   
// TODO Auto-generated catch block   
e.printStackTrace();   
} catch (IOException e) {   
// TODO Auto-generated catch block   
e.printStackTrace();   
}   
}   
/\*\*   
\* @param localPath 添加水印LOGO路径   
\* @param request 请求流对象   
\* @param request 响应流对象   
\*\*/   
@SuppressWarnings("deprecation")   
public static void createMarkLogo(String localPath,HttpServletRequest request,HttpServletResponse response) {   
try {   
FileInputStream file=new FileInputStream(localPath);   
Image fimg=ImageIO.read(file);   
int fw=fimg.getWidth(null);   
int fh=fimg.getHeight(null);   
String path=request.getRealPath(request.getServletPath());   
FileInputStream in=new FileInputStream(path);   
Image src=ImageIO.read(in);   
int w=src.getWidth(null);   
int h=src.getHeight(null);   
BufferedImage img=new BufferedImage(w,h,BufferedImage.TYPE\_INT\_RGB);//构建画板   
Graphics g=img.getGraphics();//得到画笔   
g.drawImage(src,0,0,w,h,null);//把原图片写入画板   
g.drawImage(fimg,w-20,h-15,fw,fh,null);//把水印图片写入画板   
g.dispose();//生成图片   
JPEGImageEncoder e=JPEGCodec.createJPEGEncoder(response.getOutputStream());   
e.encode(img);   
response.getOutputStream().close();   
} catch (FileNotFoundException e) {   
// TODO Auto-generated catch block   
e.printStackTrace();   
} catch (ImageFormatException e) {   
// TODO Auto-generated catch block   
e.printStackTrace();   
} catch (IOException e) {   
// TODO Auto-generated catch block   
e.printStackTrace();   
}   
}   
  
}   
  
/\*\*   
\* @param localPath 添加水印图片路径   
\* @param request 请求流对象   
\* @param request 响应流对象   
\* @param width 水印图片的宽度   
\* @param height 水印图片的长度   
\*\*/   
@SuppressWarnings("deprecation")   
public static void createMarkLogo(String localPath,HttpServletRequest request,HttpServletResponse response,int width,int height) {   
try {   
FileInputStream file=new FileInputStream(localPath);   
Image fimg=ImageIO.read(file);   
int fw=fimg.getWidth(null);   
int fh=fimg.getHeight(null);   
String path=request.getRealPath(request.getServletPath());   
FileInputStream in=new FileInputStream(path);   
Image src=ImageIO.read(in);   
int w=src.getWidth(null);//w为你过滤图片的宽度   
int h=src.getHeight(null);//h为你过滤图片的长度   
BufferedImage img=new BufferedImage(w+width,h+height,BufferedImage.TYPE\_INT\_RGB);//构建画板(画板的宽度为两个图片之和)   
Graphics g=img.getGraphics();//得到画笔   
g.drawImage(src,0,0,w,h,null);//把原图片写入画板   
g.drawImage(fimg,width,height,fw,fh,null);//把水印图片写入画板   
g.dispose();//生成图片   
JPEGImageEncoder e=JPEGCodec.createJPEGEncoder(response.getOutputStream());   
e.encode(img);   
response.getOutputStream().close();   
} catch (FileNotFoundException e) {   
// TODO Auto-generated catch block   
e.printStackTrace();   
} catch (ImageFormatException e) {   
// TODO Auto-generated catch block   
e.printStackTrace();   
} catch (IOException e) {   
// TODO Auto-generated catch block   
e.printStackTrace();   
}   
}   
  
}   
注意第三个方法的[注释](http://www.so.com/s?q=%E6%B3%A8%E9%87%8A&ie=utf-8&src=internal_wenda_recommend_textn" \t "https://wenda.so.com/q/_blank)地方g.drawImage(fimg,width,height,fw,fh,null);根据[参数](http://www.so.com/s?q=%E5%8F%82%E6%95%B0&ie=utf-8&src=internal_wenda_recommend_textn" \t "https://wenda.so.com/q/_blank)你在调调(放[原图](http://www.so.com/s?q=%E5%8E%9F%E5%9B%BE&ie=utf-8&src=internal_wenda_recommend_textn" \t "https://wenda.so.com/q/_blank)下面的)   
  
过滤器调用   
package com.jc.ts.filter;   
  
import java.io.IOException;   
  
import javax.servlet.Filter;   
import javax.servlet.FilterChain;   
import javax.servlet.FilterConfig;   
import javax.servlet.ServletException;   
import javax.servlet.ServletRequest;   
import javax.servlet.ServletResponse;   
import javax.servlet.http.HttpServletRequest;   
import javax.servlet.http.HttpServletResponse;   
  
import com.jc.ts.services.WaterMark;   
  
public class WaterFilter implements Filter {   
  
public void destroy() {   
// TODO Auto-generated method stub   
  
}   
public void doFilter(ServletRequest arg0, ServletResponse arg1,   
FilterChain arg2) throws IOException, ServletException {   
HttpServletRequest request=(HttpServletRequest)arg0;   
HttpServletResponse response=(HttpServletResponse)arg1;   
//WaterMark.createMarkSize("南京ts", request, response);   
//WaterMark.createMarkLogo("D:\\workspace\\mybook\\WebRoot\\images\\logo\\book.jpg", request, response);   
WaterMark.createMarkLogo("D:\\workspace\\mybook\\WebRoot\\images\\logo\\book.jpg", request, response,20,30);   
//注意路径为[绝对路径](http://www.so.com/s?q=%E7%BB%9D%E5%AF%B9%E8%B7%AF%E5%BE%84&ie=utf-8&src=internal_wenda_recommend_textn" \t "https://wenda.so.com/q/_blank)且三个效果不能同时执行   
  
}   
  
public void init(FilterConfig arg0) throws ServletException {   
// TODO Auto-generated method stub   
  
}   
  
}   
  
web.xml配置(写在servlet上面)   
  
<filter>   
<description>This is the description of my J2EE component</description>   
<display-name>This is the display name of my J2EE component</display-name>   
<filter-name>WaterFilter</filter-name>   
<filter-class>com.jc.ts.filter.WaterFilter</filter-class>   
</filter>   
希望你能满意。。。。。。

import javax.imageio.ImageIO;

import java.awt.Color;

import java.awt.Font;

import java.awt.Graphics2D;

import java.awt.Image;

import java.awt.image.BufferedImage;

import java.io.File;

import java.io.FileOutputStream;

**public** **class** WaterMark {

    /\*\*

     \* @param srcImgPath 源图片路径

     \* @param tarImgPath 保存的图片路径

     \* @param waterMarkContent 水印内容

     \* @param markContentColor 水印颜色

     \* @param font 水印字体

     \*/

**public** void addWaterMark(String srcImgPath, String tarImgPath, String waterMarkContent, Color markContentColor, Font font) {

**try** {

            // 读取原图片信息

            File srcImgFile = **new** File(srcImgPath);//得到文件

            Image srcImg = ImageIO.read(srcImgFile);//文件转化为图片

            int srcImgWidth = srcImg.getWidth(null);//获取图片的宽

            int srcImgHeight = srcImg.getHeight(null);//获取图片的高

            // 加水印

            BufferedImage bufImg = **new** BufferedImage(srcImgWidth, srcImgHeight, BufferedImage.TYPE\_INT\_RGB);

            Graphics2D g = bufImg.createGraphics();

            g.drawImage(srcImg, 0, 0, srcImgWidth, srcImgHeight, null);

            g.setColor(markContentColor); //根据图片的背景设置水印颜色

            g.setFont(font);              //设置字体

            //设置水印的坐标

            int x = srcImgWidth - getWatermarkLength(waterMarkContent, g) - 10;

            int y = srcImgHeight - 10;

            g.drawString(waterMarkContent, x, y);  //画出水印

            g.dispose();

            // 输出图片

            FileOutputStream outImgStream = **new** FileOutputStream(tarImgPath);

            ImageIO.write(bufImg, "jpg", outImgStream);

            System.out.println("添加水印完成");

            outImgStream.flush();

            outImgStream.close();

        } **catch** (Exception e) {

            // TODO: handle exception

        }

    }

**public** int getWatermarkLength(String waterMarkContent, Graphics2D g) {

**return** g.getFontMetrics(g.getFont()).charsWidth(waterMarkContent.toCharArray(), 0, waterMarkContent.length());

    }

**public** **static** void main(String[] args) {

        Font font = **new** Font("微软雅黑", Font.PLAIN, 60);                     //水印字体

        String srcImgPath="C://Users//pactera//Desktop//exe/333.jpg"; //源图片地址

        String tarImgPath="C://Users//pactera//Desktop//exe/444.jpg"; //待存储的地址

        String waterMarkContent="图片来源：随手而来";  //水印内容

        Color color=**new** Color(255,255,255,128);                               //水印图片色彩以及透明度

**new** WaterMark().addWaterMark(srcImgPath, tarImgPath,waterMarkContent, color ,font);

    }

}

import java.awt.Color;import java.awt.Graphics2D;import java.awt.Image;import java.awt.image.BufferedImage;import java.io.FileOutputStream;

import javax.swing.ImageIcon;

import com.sun.image.codec.jpeg.JPEGCodec;import com.sun.image.codec.jpeg.JPEGEncodeParam;import com.sun.image.codec.jpeg.JPEGImageEncoder;

public class WaterMark {

/\*\*

\* 给图片添加水印

\*

\* @param filePath

\* 需要添加水印的图片的路径

\* @param markContent

\* 水印的文字

\* @param markContentColor

\* 水印文字的颜色

\* @param qualNum

\* 图片质量

\* @return

\*/

public void createMark(String filePath, String markContent,

Color markContentColor, float qualNum) {

ImageIcon imgIcon = new ImageIcon(filePath);

Image theImg = imgIcon.getImage();

int width = theImg.getWidth(null);

int height = theImg.getHeight(null);

BufferedImage bimage = new BufferedImage(width, height,BufferedImage.TYPE\_INT\_RGB);

Graphics2D g = bimage.createGraphics();

g.setColor(markContentColor);

g.setBackground(Color.white);

g.drawImage(theImg, 0, 0, null);

g.drawString(markContent, width / 3, height / 2); // 添加水印的文字和设置水印文字出现的内容 g.dispose();

try {

FileOutputStream out = new FileOutputStream(filePath);

JPEGImageEncoder encoder = JPEGCodec.createJPEGEncoder(out);

JPEGEncodeParam param = encoder.getDefaultJPEGEncodeParam(bimage);

param.setQuality(qualNum, true);

encoder.encode(bimage, param);

out.close();

} catch (Exception e) {

return ;

}

}

public static void main(String[] args) {

WaterMark wm = new WaterMark();

wm.createMark("d://map.jpg","http://www.90che.com",Color.WHITE,70f);

}

}

我喜欢先有思路再说实现，大家都知道java可以画图的，但效率不是很高，我们又不用它写ps软件，哈哈啊哈。。。。。

转载留下出处哈：http://blog.csdn.net/zengshunyao/article/details/44758297

首先。涉及到文件就要用到文件操作，但我们这是操作图片

1、我们首先需要一个容器来存放图片ps：图片原来在磁盘（File）—-》内存（image）

2、我们要在原来的图片上加水印，就相当于要在纸上画画。。。所以我们的画纸应该是原图的一的副本，我们我们需要得到副本并用容器容乃他，

3、画画需要画笔对吧，我们要得到画副本的笔，当然比肯定有粗细。。。参数设置。。

4、如果就开始画了。。。首先的找位置。。。我们的图片就像桌面是二维图形，就用起始坐标，还是按部就班就从（0，0）开始呢

5、然后万一我们想歇着画了呢，java有旋转画布的方法

6、开始画画了。。。。给定位置。。。给定内容。。下笔了。。。

7、花完了。。是不是就要收拾现场了。。。。收拾好笔，将内存的image输出到磁盘文件。。。。

上代码

package com.funi.test;

import java.awt.AlphaComposite;

import java.awt.Color;

import java.awt.Font;

import java.awt.Graphics2D;

import java.awt.Image;

import java.awt.RenderingHints;

import java.awt.image.BufferedImage;

import java.io.File;

import java.io.FileOutputStream;

import java.io.InputStream;

import java.io.OutputStream;

import javax.imageio.ImageIO;

import javax.swing.ImageIcon;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Description: 图片水印工具类

\* @author zengshunyao

\* @version 1.0

\*/

public class ImageRemarkUtil {

// 水印透明度

private static float alpha = 0.5f;

// 水印横向位置

private static int positionWidth = 150;

// 水印纵向位置

private static int positionHeight = 300;

// 水印文字字体

private static Font font = new Font("宋体", Font.BOLD, 72);

// 水印文字颜色

private static Color color = Color.red;

/\*\*

\*

\* @param alpha

\* 水印透明度

\* @param positionWidth

\* 水印横向位置

\* @param positionHeight

\* 水印纵向位置

\* @param font

\* 水印文字字体

\* @param color

\* 水印文字颜色

\*/

public static void setImageMarkOptions(float alpha, int positionWidth,

int positionHeight, Font font, Color color) {

if (alpha != 0.0f)

ImageRemarkUtil.alpha = alpha;

if (positionWidth != 0)

ImageRemarkUtil.positionWidth = positionWidth;

if (positionHeight != 0)

ImageRemarkUtil.positionHeight = positionHeight;

if (font != null)

ImageRemarkUtil.font = font;

if (color != null)

ImageRemarkUtil.color = color;

}

/\*\*

\* 给图片添加水印图片

\*

\* @param iconPath

\* 水印图片路径

\* @param srcImgPath

\* 源图片路径

\* @param targerPath

\* 目标图片路径

\*/

public static void markImageByIcon(String iconPath, String srcImgPath,

String targerPath) {

markImageByIcon(iconPath, srcImgPath, targerPath, null);

}

/\*\*

\* 给图片添加水印图片、可设置水印图片旋转角度

\*

\* @param iconPath

\* 水印图片路径

\* @param srcImgPath

\* 源图片路径

\* @param targerPath

\* 目标图片路径

\* @param degree

\* 水印图片旋转角度

\*/

public static void markImageByIcon(String iconPath, String srcImgPath,

String targerPath, Integer degree) {

OutputStream os = null;

try {

Image srcImg = ImageIO.read(new File(srcImgPath));

BufferedImage buffImg = new BufferedImage(srcImg.getWidth(null),

srcImg.getHeight(null), BufferedImage.TYPE\_INT\_RGB);

// 1、得到画笔对象

Graphics2D g = buffImg.createGraphics();

// 2、设置对线段的锯齿状边缘处理

g.setRenderingHint(RenderingHints.KEY\_INTERPOLATION,

RenderingHints.VALUE\_INTERPOLATION\_BILINEAR);

g.drawImage(

srcImg.getScaledInstance(srcImg.getWidth(null),

srcImg.getHeight(null), Image.SCALE\_SMOOTH), 0, 0,

null);

// 3、设置水印旋转

if (null != degree) {

g.rotate(Math.toRadians(degree),

(double) buffImg.getWidth() / 2,

(double) buffImg.getHeight() / 2);

}

// 4、水印图片的路径 水印图片一般为gif或者png的，这样可设置透明度

ImageIcon imgIcon = new ImageIcon(iconPath);

// 5、得到Image对象。

Image img = imgIcon.getImage();

g.setComposite(AlphaComposite.getInstance(AlphaComposite.SRC\_ATOP,

alpha));

// 6、水印图片的位置

g.drawImage(img, positionWidth, positionHeight, null);

g.setComposite(AlphaComposite.getInstance(AlphaComposite.SRC\_OVER));

// 7、释放资源

g.dispose();

// 8、生成图片

os = new FileOutputStream(targerPath);

ImageIO.write(buffImg, "JPG", os);

System.out.println("图片完成添加水印图片");

} catch (Exception e) {

e.printStackTrace();

} finally {

try {

if (null != os)

os.close();

} catch (Exception e) {

e.printStackTrace();

}

}

}

/\*\*

\* 给图片添加水印文字

\*

\* @param logoText

\* 水印文字

\* @param srcImgPath

\* 源图片路径

\* @param targerPath

\* 目标图片路径

\*/

public static void markImageByText(String logoText, String srcImgPath,

String targerPath) {

markImageByText(logoText, srcImgPath, targerPath, null);

}

/\*\*

\* 给图片添加水印文字、可设置水印文字的旋转角度

\*

\* @param logoText

\* @param srcImgPath

\* @param targerPath

\* @param degree

\*/

public static void markImageByText(String logoText, String srcImgPath,

String targerPath, Integer degree) {

InputStream is = null;

OutputStream os = null;

try {

// 1、源图片

Image srcImg = ImageIO.read(new File(srcImgPath));

BufferedImage buffImg = new BufferedImage(srcImg.getWidth(null),

srcImg.getHeight(null), BufferedImage.TYPE\_INT\_RGB);

// 2、得到画笔对象

Graphics2D g = buffImg.createGraphics();

// 3、设置对线段的锯齿状边缘处理

g.setRenderingHint(RenderingHints.KEY\_INTERPOLATION,

RenderingHints.VALUE\_INTERPOLATION\_BILINEAR);

g.drawImage(

srcImg.getScaledInstance(srcImg.getWidth(null),

srcImg.getHeight(null), Image.SCALE\_SMOOTH), 0, 0,

null);

// 4、设置水印旋转

if (null != degree) {

g.rotate(Math.toRadians(degree),

(double) buffImg.getWidth() / 2,

(double) buffImg.getHeight() / 2);

}

// 5、设置水印文字颜色

g.setColor(color);

// 6、设置水印文字Font

g.setFont(font);

// 7、设置水印文字透明度

g.setComposite(AlphaComposite.getInstance(AlphaComposite.SRC\_ATOP,

alpha));

// 8、第一参数->设置的内容，后面两个参数->文字在图片上的坐标位置(x,y)

g.drawString(logoText, positionWidth, positionHeight);

// 9、释放资源

g.dispose();

// 10、生成图片

os = new FileOutputStream(targerPath);

ImageIO.write(buffImg, "JPG", os);

System.out.println("图片完成添加水印文字");

} catch (Exception e) {

e.printStackTrace();

} finally {

try {

if (null != is)

is.close();

} catch (Exception e) {

e.printStackTrace();

}

try {

if (null != os)

os.close();

} catch (Exception e) {

e.printStackTrace();

}

}

}

public static void main(String[] args) {

String srcImgPath = "d:/1.jpg";

String logoText = "复 印 无 效";

String iconPath = "d:/2.jpg";

String targerTextPath = "d:/qie\_text.jpg";

String targerTextPath2 = "d:/qie\_text\_rotate.jpg";

String targerIconPath = "d:/qie\_icon.jpg";

String targerIconPath2 = "d:/qie\_icon\_rotate.jpg";

System.out.println("给图片添加水印文字开始...");

// 给图片添加水印文字

markImageByText(logoText, srcImgPath, targerTextPath);

// 给图片添加水印文字,水印文字旋转-45

markImageByText(logoText, srcImgPath, targerTextPath2, -45);

System.out.println("给图片添加水印文字结束...");

System.out.println("给图片添加水印图片开始...");

setImageMarkOptions(0.3f, 1, 1, null, null);

// 给图片添加水印图片

markImageByIcon(iconPath, srcImgPath, targerIconPath);

// 给图片添加水印图片,水印图片旋转-45

markImageByIcon(iconPath, srcImgPath, targerIconPath2, -45);

System.out.println("给图片添加水印图片结束...");

}

}

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

43

44

45

46

47

48

49

50

51

52

53

54

55

56

57

58

59

60

61

62

63

64

65

66

67

68

69

70

71

72

73

74

75

76

77

78

79

80

81

82

83

84

85

86

87

88

89

90

91

92

93

94

95

96

97

98

99

100

101

102

103

104

105

106

107

108

109

110

111

112

113

114

115

116

117

118

119

120

121

122

123

124

125

126

127

128

129

130

131

132

133

134

135

136

137

138

139

140

141

142

143

144

145

146

147

148

149

150

151

152

153

154

155

156

157

158

159

160

161

162

163

164

165

166

167

168

169

170

171

172

173

174

175

176

177

178

179

180

181

182

183

184

185

186

187

188

189

190

191

192

193

194

195

196

197

198

199

200

201

202

203

204

205

206

207

208

209

210

211

212

213

214

215

216

217

218

219

220

221

222

223

224

225

226

227

228

229

230

231

232

233

234

235

236

237

238

239

240

241

242

243

244

245

246

247

248

249

250

251

252

253

254

255

256

257

258

259

260

261

262

263

请对别人知识的尊重，http://blog.csdn.net/zengshunyao/article/details/44758297

---------------------

作者：贪玩好耍调皮捣蛋

来源：CSDN

原文：https://blog.csdn.net/zengshunyao/article/details/44758297

版权声明：本文为博主原创文章，转载请附上博文链接！

周末在家上围脖，看到sina围脖上图片都有水印，想到Java有专门Image的处理包，同样应该可以实现水印功能，查了资料小试下来发现java实现水印还是非常方便的，水印可以是图片或者文字，而且还可以选择不同的角度旋转呢，以后有需要可以写个代码批量处理自己的图片了。

本门目录：

图片水印

文字水印

[一]、图片水印

Java代码

package michael.io.image;

import java.awt.AlphaComposite;

import java.awt.Graphics2D;

import java.awt.Image;

import java.awt.RenderingHints;

import java.awt.image.BufferedImage;

import java.io.File;

import java.io.FileInputStream;

import java.io.FileOutputStream;

import java.io.InputStream;

import java.io.OutputStream;

import javax.imageio.ImageIO;

import javax.swing.ImageIcon;

import com.sun.image.codec.jpeg.JPEGCodec;

import com.sun.image.codec.jpeg.JPEGImageDecoder;

import com.sun.image.codec.jpeg.JPEGImageEncoder;

/\*\*

 \* 图片水印

 \* @blog http://sjsky.iteye.com

 \* @author Michael

 \*/

public class ImageMarkLogoByIcon {

    /\*\*

     \* @param args

     \*/

    public static void main(String[] args) {

        String srcImgPath = "d:/test/michael/myblog\_01.png";

        String iconPath = "d:/test/michael/blog\_logo.png";

        String targerPath = "d:/test/michael/img\_mark\_icon.jpg";

        String targerPath2 = "d:/test/michael/img\_mark\_icon\_rotate.jpg";

        // 给图片添加水印

        ImageMarkLogoByIcon.markImageByIcon(iconPath, srcImgPath, targerPath);

        // 给图片添加水印,水印旋转-45

        ImageMarkLogoByIcon.markImageByIcon(iconPath, srcImgPath, targerPath2,

                -45);

    }

    /\*\*

     \* 给图片添加水印

     \* @param iconPath 水印图片路径

     \* @param srcImgPath 源图片路径

     \* @param targerPath 目标图片路径

     \*/

    public static void markImageByIcon(String iconPath, String srcImgPath,

            String targerPath) {

        markImageByIcon(iconPath, srcImgPath, targerPath, null);

    }

    /\*\*

     \* 给图片添加水印、可设置水印图片旋转角度

     \* @param iconPath 水印图片路径

     \* @param srcImgPath 源图片路径

     \* @param targerPath 目标图片路径

     \* @param degree 水印图片旋转角度

     \*/

    public static void markImageByIcon(String iconPath, String srcImgPath,

            String targerPath, Integer degree) {

        OutputStream os = null;

        try {

            Image srcImg = ImageIO.read(new File(srcImgPath));

            BufferedImage buffImg = new BufferedImage(srcImg.getWidth(null),

                    srcImg.getHeight(null), BufferedImage.TYPE\_INT\_RGB);

            // 得到画笔对象

            // Graphics g= buffImg.getGraphics();

            Graphics2D g = buffImg.createGraphics();

            // 设置对线段的锯齿状边缘处理

            g.setRenderingHint(RenderingHints.KEY\_INTERPOLATION,

                    RenderingHints.VALUE\_INTERPOLATION\_BILINEAR);

            g.drawImage(srcImg.getScaledInstance(srcImg.getWidth(null), srcImg

                    .getHeight(null), Image.SCALE\_SMOOTH), 0, 0, null);

            if (null != degree) {

                // 设置水印旋转

                g.rotate(Math.toRadians(degree),

                        (double) buffImg.getWidth() / 2, (double) buffImg

                                .getHeight() / 2);

            }

            // 水印图象的路径 水印一般为gif或者png的，这样可设置透明度

            ImageIcon imgIcon = new ImageIcon(iconPath);

            // 得到Image对象。

            Image img = imgIcon.getImage();

            float alpha = 0.5f; // 透明度

            g.setComposite(AlphaComposite.getInstance(AlphaComposite.SRC\_ATOP,

                    alpha));

            // 表示水印图片的位置

            g.drawImage(img, 150, 300, null);

            g.setComposite(AlphaComposite.getInstance(AlphaComposite.SRC\_OVER));

            g.dispose();

            os = new FileOutputStream(targerPath);

            // 生成图片

            ImageIO.write(buffImg, "JPG", os);

            System.out.println("图片完成添加Icon印章。。。。。。");

        } catch (Exception e) {

            e.printStackTrace();

        } finally {

            try {

                if (null != os)

                    os.close();

            } catch (Exception e) {

                e.printStackTrace();

            }

        }

    }

}

---------------------

作者：dy\_smile

来源：CSDN

原文：https://blog.csdn.net/dy\_smile/article/details/6722907

版权声明：本文为博主原创文章，转载请附上博文链接！

java中调整图片的大小并给图片加水印（或是文字）

前几天给图片加水印一张一张的处理，相当的麻烦，相当的累，最近有时间写了一个动态实现这种效果的java类

package ouc.sei.test.servlet;

import java.io.\*;  
import java.awt.\*;  
import java.awt.image.\*;  
import java.util.\*;  
import org.apache.log4j.Logger;

import javax.imageio.ImageIO;

import com.sun.image.codec.jpeg.JPEGCodec;  
import com.sun.image.codec.jpeg.JPEGImageEncoder;

public   final   class  ImageUtils  {  
  public static Logger logger = Logger.getLogger(ImageUtils.class);  
       
    public   static   void  main(String[] args)  {  
     File sourceimage=new File("c:/image/") ;  
     File flist[]=sourceimage.listFiles();  
     for(int i=0;i<flist.length;i++){  
      System.out.println("flist[i] is "+flist[i].getName());  
      adjustpic("c:/image/"+flist[i].getName(),"c:/adjimage/ddddd"+i+".jpg",500,333);  
      pressImage("C:/33.PNG","c:/adjimage/ddddd"+i+".jpg",20,20);  
      //pressText("ybc","c:/adjimage/ddddd"+i+".jpg","宋体",800,234,12,20,20);  
     }  
     }  
    public final static void adjustpic(String originalpic,String adjustedpic,int imgwidth,int imgheight){  
     String filecreate=adjustedpic.substring(0,adjustedpic.lastIndexOf("/")+1);//获取最后一个“/”的位置，并把取得它之前的文件路径  
     try{  
      File adjfile=new File(filecreate);  
      if(!adjfile.exists()){  
       adjfile.mkdir();  
      }  
     }catch(Exception e){  
      System.out.println("新建文件夹操作出错");    
      e.printStackTrace();      
     }  
     try   {  
            File \_file  =   new  File(originalpic);  
            Image src  =  ImageIO.read(\_file);  
             int  wideth  =  src.getWidth( null );  
             int  height  =  src.getHeight( null );  
             boolean flag=true;  
             if (wideth >= height)  
             {  
                 flag = true;  
                 logger.info("图片长度Width大于或者等于高度Height");  
                 System.out.println("图片长度Width大于或者等于高度Height");  
             }  
             else  
             {  
                 flag = false;  
                 logger.info("图片长度Width小于高度Height");  
                 System.out.println("图片长度Width小于高度Height");  
             }  
             Float f = new Float(0);  
             if(flag){  
              f = ((new Float(height) \* imgwidth) / new Float(wideth));  
                 logger.info("新生成的图片的高度height是:" + f);  
                 System.out.println("新的页面的高度是："+f);

                 int newHeight = f.intValue() + 1;  
             
            BufferedImage image  =   new  BufferedImage(imgwidth, newHeight,  
                    BufferedImage.TYPE\_INT\_RGB);  
            Graphics g  =  image.createGraphics();  
            g.drawImage(src,  0 ,  0 , imgwidth, newHeight,  null );  
            g.dispose();  
            FileOutputStream out  =   new  FileOutputStream(adjustedpic);  
            JPEGImageEncoder encoder  =  JPEGCodec.createJPEGEncoder(out);  
            encoder.encode(image);  
            out.close();  
             }  
             else{  
              f = ((new Float(wideth) \* imgheight) / new Float(height));  
                 logger.info("新生成的图片的宽度width是:" + f);  
                 System.out.println("新的图片的宽度："+f);

                 int newWidth = f.intValue() + 1;  
                 BufferedImage image  =   new  BufferedImage(newWidth, imgheight,  
                         BufferedImage.TYPE\_INT\_RGB);  
                 Graphics g  =  image.createGraphics();  
                 g.drawImage(src,  0 ,  0 , newWidth, imgheight,  null );  
            g.dispose();  
            FileOutputStream out  =   new  FileOutputStream(adjustedpic);  
            JPEGImageEncoder encoder  =  JPEGCodec.createJPEGEncoder(out);  
            encoder.encode(image);  
            out.close();  
             }  
        }   catch  (Exception e)  {  
            e.printStackTrace();  
        }  
    }  
    
    public   final   static   void  pressImage(String pressImg, String targetImg,  int  x,  int y)  {  
        try   {  
           File \_file  =   new  File(targetImg);  
           Image src  =  ImageIO.read(\_file);  
            int  wideth  =  src.getWidth( null );  
            int  height  =  src.getHeight( null );  
           BufferedImage image  =   new  BufferedImage(wideth, height,  
                   BufferedImage.TYPE\_INT\_RGB);  
           Graphics g  =  image.createGraphics();  
           g.drawImage(src,  0 ,  0 , wideth, height,  null );

            //  水印文件  
           File \_filebiao  =   new  File(pressImg);  
           Image src\_biao  =  ImageIO.read(\_filebiao);  
            int  wideth\_biao  =  src\_biao.getWidth( null );  
            int  height\_biao  =  src\_biao.getHeight( null );  
           g.drawImage(src\_biao, wideth  -  wideth\_biao  -  x, height  -  height\_biao  - y, wideth\_biao,  
                   height\_biao,  null );  
           g.dispose();  
           FileOutputStream out  =   new  FileOutputStream(targetImg);  
           JPEGImageEncoder encoder  =  JPEGCodec.createJPEGEncoder(out);  
           encoder.encode(image);  
           out.close();  
       }   catch  (Exception e)  {  
           e.printStackTrace();  
       }  
   }  
       
       
      public   static   void  pressText(String pressText, String targetImg, String fontName, int fontStyle,  int  color,  int  fontSize,  int  x,  int  y)  {  
          try   {  
             File \_file  =   new  File(targetImg);  
             Image src  =  ImageIO.read(\_file);  
              int  wideth  =  src.getWidth( null );  
              int  height  =  src.getHeight( null );  
             BufferedImage image  =   new  BufferedImage(wideth, height,  
                     BufferedImage.TYPE\_INT\_RGB);  
             Graphics g  =  image.createGraphics();  
             g.drawImage(src,  0 ,  0 , wideth, height,  null );  
                         g.setColor(Color.RED);  
             g.setFont( new  Font(fontName, fontStyle, fontSize));  
           
   
             g.drawString(pressText, wideth  -  fontSize  -  x, height  -  fontSize / 2   -  y);  
             g.dispose();  
             FileOutputStream out  =   new  FileOutputStream(targetImg);  
             JPEGImageEncoder encoder  =  JPEGCodec.createJPEGEncoder(out);  
             encoder.encode(image);  
             out.close();  
         }   catch  (Exception e)  {  
             System.out.println(e);  
         }  
     }

 }

# 用java实现给图片增加图片水印或者文字水印（也支持视频图像帧添加水印）

2016年10月30日 19:36:38 [本博客已停止维护！-eguid](https://me.csdn.net/eguid_1" \t "https://blog.csdn.net/eguid_1/article/details/_blank) 阅读数 15639 标签： [java](https://so.csdn.net/so/search/s.do?q=java&t=blog" \t "https://blog.csdn.net/eguid_1/article/details/_blank)[图片](https://so.csdn.net/so/search/s.do?q=%E5%9B%BE%E7%89%87&t=blog" \t "https://blog.csdn.net/eguid_1/article/details/_blank)[图片水印](https://so.csdn.net/so/search/s.do?q=%E5%9B%BE%E7%89%87%E6%B0%B4%E5%8D%B0&t=blog" \t "https://blog.csdn.net/eguid_1/article/details/_blank)[文字水印](https://so.csdn.net/so/search/s.do?q=%E6%96%87%E5%AD%97%E6%B0%B4%E5%8D%B0&t=blog" \t "https://blog.csdn.net/eguid_1/article/details/_blank) 更多

个人分类： [java](https://blog.csdn.net/eguid_1/article/category/6251299" \t "https://blog.csdn.net/eguid_1/article/details/_blank)

版权声明：eguid温馨提示：本博客所有原创文章均采用知识共享署名-相同方式共享 3.0 中国大陆许可协议进行许可。如有转载请注明出处和eguid作者名，侵权必究！ https://blog.csdn.net/eguid\_1/article/details/52973508

javaCV图像处理系列：

[javaCV图像处理之1：实时视频添加文字水印并截取视频图像保存成图片，实现文字水印的字体、位置、大小、粗度、翻转、平滑等操作](http://blog.csdn.net/eguid_1/article/details/53236923" \t "https://blog.csdn.net/eguid_1/article/details/_blank)

# **[javaCV图像处理之2：实时视频添加图片水印，实现不同大小图片叠加，图像透明度控制](http://blog.csdn.net/eguid_1/article/details/53259649" \t "https://blog.csdn.net/eguid_1/article/details/_blank)**

# **前言：**

这是个很常用的操作，一般我们的网站在用户上传图片时都会给图片添加一个水印以防止其他站点盗图的行为

实现功能：①给图片增加文字水印②给图片增加图片水印

# **一、核心功能实现：**

## **1、添加文字水印**

// 加文字水印

public void mark(BufferedImage bufImg, Image img, String text, Font font, Color color, int x, int y) {

Graphics2D g = bufImg.createGraphics();

g.drawImage(img, 0, 0, bufImg.getWidth(), bufImg.getHeight(), null);

g.setColor(color);

g.setFont(font);

g.drawString(text, x, y);

g.dispose();

}

## **2、添加图片水印**

// 加图片水印

public void mark(BufferedImage bufImg, Image img, Image markImg, int width, int height, int x, int y) {

Graphics2D g = bufImg.createGraphics();

g.drawImage(img, 0, 0, bufImg.getWidth(), bufImg.getHeight(), null);

g.drawImage(markImg, x, y, width, height, null);

g.dispose();

}

以上两个方法实现了最核心的功能：给图片添加水印。

****补充：****

为什么要这么写？

考虑到该方法不仅可以用于给图片文件增加水印，而且还可以给视频的每一帧图像也添加，所以为了方便不同场合重复使用，去除了不必要的依赖关系。

有了核心的两个方法就我们可以给图片文件增加水印了

# **二、功能性实现**

## **1、给图片增加文字水印**

/\*\*

\* 给图片增加文字水印

\*

\* @param imgPath

\* -要添加水印的图片路径

\* @param outImgPath

\* -输出路径

\* @param text-文字

\* @param font

\* -字体

\* @param color

\* -颜色

\* @param x

\* -文字位于当前图片的横坐标

\* @param y

\* -文字位于当前图片的竖坐标

\*/

public void mark(String imgPath, String outImgPath, String text, Font font, Color color, int x, int y) {

try {

// 读取原图片信息

File imgFile = null;

Image img = null;

if (imgPath != null) {

imgFile = new File(imgPath);

}

if (imgFile != null && imgFile.exists() && imgFile.isFile() && imgFile.canRead()) {

img = ImageIO.read(imgFile);

}

int imgWidth = img.getWidth(null);

int imgHeight = img.getHeight(null);

// 加水印

BufferedImage bufImg = new BufferedImage(imgWidth, imgHeight, BufferedImage.TYPE\_INT\_RGB);

mark(bufImg, img, text, font, color, x, y);

// 输出图片

FileOutputStream outImgStream = new FileOutputStream(outImgPath);

ImageIO.write(bufImg, "jpg", outImgStream);

outImgStream.flush();

outImgStream.close();

} catch (Exception e) {

e.printStackTrace();

}

}

## **2、给图片增加图片水印**

/\*\*

\* 给图片增加图片水印

\*

\* @param inputImg

\* -源图片，要添加水印的图片

\* @param markImg

\* - 水印图片

\* @param outputImg

\* -输出图片(可以是源图片)

\* @param width

\* - 水印图片宽度

\* @param height

\* -水印图片高度

\* @param x

\* -横坐标，相对于源图片

\* @param y

\* -纵坐标，同上

\*/

public void mark(String inputImg, String markImg, String outputImg, int width, int height, int x, int y) {

// 读取原图片信息

File inputImgFile = null;

File markImgFile = null;

Image img = null;

Image mark = null;

try {

if (inputImg != null && markImg != null) {

inputImgFile = new File(inputImg);

markImgFile = new File(markImg);

}

if (inputImgFile != null && inputImgFile.exists() && inputImgFile.isFile() && inputImgFile.canRead()) {

img = ImageIO.read(inputImgFile);

}

if (markImgFile != null && markImgFile.exists() && markImgFile.isFile() && markImgFile.canRead()) {

mark = ImageIO.read(markImgFile);

}

int imgWidth = img.getWidth(null);

int imgHeight = img.getHeight(null);

BufferedImage bufImg = new BufferedImage(imgWidth, imgHeight, BufferedImage.TYPE\_INT\_RGB);

mark(bufImg, img, mark, width, height, x, y);

FileOutputStream outImgStream = new FileOutputStream(outputImg);

ImageIO.write(bufImg, "jpg", outImgStream);

outImgStream.flush();

outImgStream.close();

} catch (IOException e) {

e.printStackTrace();

}

}

3、测试一下效果

文字水印的字体和颜色需要自行定义，这里我们使用宋体，14号字体，颜色选择橙色 - -！，坐标是x轴0，y轴等于字体的大小，也就是图片的左上角。

public static void main(String[] args) {

Font font = new Font("宋体", Font.PLAIN, 14);

// 原图位置, 输出图片位置, 水印文字颜色, 水印文字

// new MarkText4J().mark("eguidMarkText2.jpg", "eguidMarkText2.jpg", "水印效果测试", font, Color.ORANGE, 0, 14);

// 增加图片水印

new MarkText4J().mark("eguidMarkText2.jpg", "eguid.jpg", "eguidMarkText3.jpg", 40, 20, 0, 14);

}

简直棒极了，至此我们可以随意的给图片任意位置添加任意文字或者图片水印了。

最后，额。。。导出的图片比原本图片还小是什么鬼？欢迎加入讨论群一起讨论：371249677。- -！

# [用java实现给图片增加图片水印或者文字水印（也支持视频图像帧添加水印）](https://www.cnblogs.com/fengwenzhee/p/9124794.html)

**阅读目录**

* [1、添加文字水印](https://www.cnblogs.com/fengwenzhee/p/9124794.html" \l "_label0)
* [2、添加图片水印](https://www.cnblogs.com/fengwenzhee/p/9124794.html" \l "_label1)
* [1、给图片增加文字水印](https://www.cnblogs.com/fengwenzhee/p/9124794.html" \l "_label2)
* [2、给图片增加图片水印](https://www.cnblogs.com/fengwenzhee/p/9124794.html" \l "_label3)

这是个很常用的操作，一般我们的网站在用户上传图片时都会给图片添加一个水印以防止其他站点盗图的行为

实现功能：①给图片增加文字水印②给图片增加图片水印

# **一、核心功能实现：**

[回到顶部](https://www.cnblogs.com/fengwenzhee/p/9124794.html" \l "_labelTop)

## **1、添加文字水印**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9 | // 加文字水印      public void mark(BufferedImage bufImg, Image img, String text, Font font, Color color, int x, int y) {          Graphics2D g = bufImg.createGraphics();          g.drawImage(img, 0, 0, bufImg.getWidth(), bufImg.getHeight(), null);          g.setColor(color);          g.setFont(font);          g.drawString(text, x, y);          g.dispose();      } |

[回到顶部](https://www.cnblogs.com/fengwenzhee/p/9124794.html" \l "_labelTop)

## **2、添加图片水印**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | // 加图片水印      public void mark(BufferedImage bufImg, Image img, Image markImg, int width, int height, int x, int y) {          Graphics2D g = bufImg.createGraphics();          g.drawImage(img, 0, 0, bufImg.getWidth(), bufImg.getHeight(), null);          g.drawImage(markImg, x, y, width, height, null);          g.dispose();      } |

　　以上两个方法实现了最核心的功能：给图片添加水印。

补充：

为什么要这么写？

考虑到该方法不仅可以用于给图片文件增加水印，而且还可以给视频的每一帧图像也添加，所以为了方便不同场合重复使用，去除了不必要的依赖关系。

有了核心的两个方法就我们可以给图片文件增加水印了

# **二、功能性实现**

[回到顶部](https://www.cnblogs.com/fengwenzhee/p/9124794.html" \l "_labelTop)

## **1、给图片增加文字水印**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42 | /\*\*       \* 给图片增加文字水印       \*       \* @param imgPath       \*            -要添加水印的图片路径       \* @param outImgPath       \*            -输出路径       \* @param text-文字       \* @param font       \*            -字体       \* @param color       \*            -颜色       \* @param x       \*            -文字位于当前图片的横坐标       \* @param y       \*            -文字位于当前图片的竖坐标       \*/      public void mark(String imgPath, String outImgPath, String text, Font font, Color color, int x, int y) {          try {              // 读取原图片信息              File imgFile = null;              Image img = null;              if (imgPath != null) {                  imgFile = new File(imgPath);              }              if (imgFile != null && imgFile.exists() && imgFile.isFile() && imgFile.canRead()) {                  img = ImageIO.read(imgFile);              }              int imgWidth = img.getWidth(null);              int imgHeight = img.getHeight(null);              // 加水印              BufferedImage bufImg = new BufferedImage(imgWidth, imgHeight, BufferedImage.TYPE\_INT\_RGB);              mark(bufImg, img, text, font, color, x, y);              // 输出图片              FileOutputStream outImgStream = new FileOutputStream(outImgPath);              ImageIO.write(bufImg, "jpg", outImgStream);              outImgStream.flush();              outImgStream.close();          } catch (Exception e) {              e.printStackTrace();          }      } |

[回到顶部](https://www.cnblogs.com/fengwenzhee/p/9124794.html" \l "_labelTop)

## **2、给图片增加图片水印**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51 | /\*\*       \* 给图片增加图片水印       \*       \* @param inputImg       \*            -源图片，要添加水印的图片       \* @param markImg       \*            - 水印图片       \* @param outputImg       \*            -输出图片(可以是源图片)       \* @param width       \*            - 水印图片宽度       \* @param height       \*            -水印图片高度       \* @param x       \*            -横坐标，相对于源图片       \* @param y       \*            -纵坐标，同上       \*/      public void mark(String inputImg, String markImg, String outputImg, int width, int height, int x, int y) {          // 读取原图片信息          File inputImgFile = null;          File markImgFile = null;          Image img = null;          Image mark = null;          try {              if (inputImg != null && markImg != null) {                  inputImgFile = new File(inputImg);                  markImgFile = new File(markImg);              }              if (inputImgFile != null && inputImgFile.exists() && inputImgFile.isFile() && inputImgFile.canRead()) {                    img = ImageIO.read(inputImgFile);                }              if (markImgFile != null && markImgFile.exists() && markImgFile.isFile() && markImgFile.canRead()) {                    mark = ImageIO.read(markImgFile);                }              int imgWidth = img.getWidth(null);              int imgHeight = img.getHeight(null);              BufferedImage bufImg = new BufferedImage(imgWidth, imgHeight, BufferedImage.TYPE\_INT\_RGB);              mark(bufImg, img, mark, width, height, x, y);              FileOutputStream outImgStream = new FileOutputStream(outputImg);              ImageIO.write(bufImg, "jpg", outImgStream);              outImgStream.flush();              outImgStream.close();          } catch (IOException e) {              e.printStackTrace();          }      } |

　　3、测试一下效果

文字水印的字体和颜色需要自行定义，这里我们使用宋体，14号字体，颜色选择橙色 - -！，坐标是x轴0，y轴等于字体的大小，也就是图片的左上角。

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | public static void main(String[] args) {          Font font = new Font("宋体", Font.PLAIN, 14);          // 原图位置, 输出图片位置, 水印文字颜色, 水印文字          // new MarkText4J().mark("eguidMarkText2.jpg", "eguidMarkText2.jpg", "水印效果测试", font, Color.ORANGE, 0, 14);          // 增加图片水印          new MarkText4J().mark("eguidMarkText2.jpg", "eguid.jpg", "eguidMarkText3.jpg", 40, 20, 0, 14);      } |

本文转载自：[https://blog.csdn.net/eguid\_1/article/details/52973508](https://blog.csdn.net/eguid_1/article/details/52973508" \t "https://www.cnblogs.com/fengwenzhee/p/_blank)

package com.sm.utils;

import java.awt.AlphaComposite;

import java.awt.Graphics2D;

import java.awt.Image;

import java.awt.RenderingHints;

import java.awt.image.BufferedImage;

import java.io.File;

import java.io.FileOutputStream;

import java.io.OutputStream;

import javax.imageio.ImageIO;

import javax.swing.ImageIcon;

/\*\*

\*

\* 生成水印

\*

\*/public class ImageMarkUtils {

/\*\* 水印透明度 \*/

private static float alpha = 0.5f;

/\*\* 水印图片旋转角度 \*/

private static double degree = 0f;

private static int interval = 0;

/\*\*

\* 设置水印参数，不设置就使用默认值

\*

\* @param alpha

\* 水印透明度

\* @param degree

\* 水印图片旋转角度 \*

\* @param interval

\* 水印图片间隔

\*/

public static void setImageMarkOptions(float alpha, int degree, int interval) {

if (alpha != 0.0f) {

ImageMarkUtils.alpha = alpha;

}

if (degree != 0f) {

ImageMarkUtils.degree = degree;

}

if (interval != 0f) {

ImageMarkUtils.interval = interval;

}

}

/\*\*

\* 给图片添加水印图片

\*

\* @param waterImgPath

\* 水印图片路径

\* @param srcImgPath

\* 源图片路径

\* @param targerPath

\* 目标图片路径

\*/

public static void waterMarkByImg(String waterImgPath, String srcImgPath, String targerPath) throws Exception {

waterMarkByImg(waterImgPath, srcImgPath, targerPath, 0);

}

/\*\*

\* 给图片添加水印图片

\*

\* @param waterImgPath

\* 水印图片路径

\* @param srcImgPath

\* 源图片路径

\* @param targerPath

\* 目标图片路径

\*/

public static void waterMarkByImg(String waterImgPath, String srcImgPath) {

try {

waterMarkByImg(waterImgPath, srcImgPath, srcImgPath, 0);

} catch (Exception e) {

// TODO Auto-generated catch block e.printStackTrace();

}

}

/\*\*

\* 给图片添加水印图片、可设置水印图片旋转角度

\*

\* @param waterImgPath

\* 水印图片路径

\* @param srcImgPath

\* 源图片路径

\* @param targerPath

\* 目标图片路径

\* @param degree

\* 水印图片旋转角度

\*/

public static void waterMarkByImg(String waterImgPath, String srcImgPath, String targerPath, double degree)

throws Exception {

OutputStream os = null;

try {

Image srcImg = ImageIO.read(new File(srcImgPath));

BufferedImage buffImg = new BufferedImage(srcImg.getWidth(null), srcImg.getHeight(null),

BufferedImage.TYPE\_INT\_RGB);

// 1、得到画笔对象

Graphics2D g = buffImg.createGraphics();

// 2、设置对线段的锯齿状边缘处理 g.setRenderingHint(RenderingHints.KEY\_INTERPOLATION, RenderingHints.VALUE\_INTERPOLATION\_BILINEAR);

g.drawImage(srcImg.getScaledInstance(srcImg.getWidth(null), srcImg.getHeight(null), Image.SCALE\_SMOOTH), 0,

0, null);

// 3、设置水印旋转

if (0 != degree) {

g.rotate(Math.toRadians(degree), (double) buffImg.getWidth() / 2, (double) buffImg.getHeight() / 2);

}

// 4、水印图片的路径 水印图片一般为gif或者png的，这样可设置透明度

ImageIcon imgIcon = new ImageIcon(waterImgPath);

// 5、得到Image对象。

Image img = imgIcon.getImage();

g.setComposite(AlphaComposite.getInstance(AlphaComposite.SRC\_ATOP, alpha));

// 6、水印图片的位置

for (int height = interval + imgIcon.getIconHeight(); height < buffImg.getHeight(); height = height

+ interval + imgIcon.getIconHeight()) {

for (int weight = interval + imgIcon.getIconWidth(); weight < buffImg.getWidth(); weight = weight

+ interval + imgIcon.getIconWidth()) {

g.drawImage(img, weight - imgIcon.getIconWidth(), height - imgIcon.getIconHeight(), null);

}

}

g.setComposite(AlphaComposite.getInstance(AlphaComposite.SRC\_OVER));

// 7、释放资源 g.dispose();

// 8、生成图片

os = new FileOutputStream(targerPath);

ImageIO.write(buffImg, "JPG", os);

System.out.println("图片完成添加水印图片");

} catch (Exception e) {

e.printStackTrace();

} finally {

try {

if (null != os)

os.close();

} catch (Exception e) {

e.printStackTrace();

}

}

}

public static void main(String[] args) {

System.out.println("..添加水印图片开始...");

/\*\*

\* watermarkPath 水印图片地址 加水印图片地址 上传成功后文件地址

\*/

// 修改默认参数

String imgPath = "C:/xym/abc.jpg";// 测试需加水印图片

String watermarkPath = "C:/xym/xinyimei.png"; // 测试水印图片

ImageMarkUtils.setImageMarkOptions(0.0f, 0, 20);

ImageMarkUtils.waterMarkByImg(watermarkPath, imgPath);

System.out.println("..添加水印图片结束...");

}

}

# [用java实现给图片增加图片水印或者文字水印（也支持视频图像帧添加水印）](https://www.cnblogs.com/fengwenzhee/p/9124794.html)

**阅读目录**

* [1、添加文字水印](https://www.cnblogs.com/fengwenzhee/p/9124794.html" \l "_label0)
* [2、添加图片水印](https://www.cnblogs.com/fengwenzhee/p/9124794.html" \l "_label1)
* [1、给图片增加文字水印](https://www.cnblogs.com/fengwenzhee/p/9124794.html" \l "_label2)
* [2、给图片增加图片水印](https://www.cnblogs.com/fengwenzhee/p/9124794.html" \l "_label3)

这是个很常用的操作，一般我们的网站在用户上传图片时都会给图片添加一个水印以防止其他站点盗图的行为

实现功能：①给图片增加文字水印②给图片增加图片水印

# **一、核心功能实现：**

[回到顶部](https://www.cnblogs.com/fengwenzhee/p/9124794.html" \l "_labelTop)

## **1、添加文字水印**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9 | // 加文字水印      public void mark(BufferedImage bufImg, Image img, String text, Font font, Color color, int x, int y) {          Graphics2D g = bufImg.createGraphics();          g.drawImage(img, 0, 0, bufImg.getWidth(), bufImg.getHeight(), null);          g.setColor(color);          g.setFont(font);          g.drawString(text, x, y);          g.dispose();      } |

[回到顶部](https://www.cnblogs.com/fengwenzhee/p/9124794.html" \l "_labelTop)

## **2、添加图片水印**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | // 加图片水印      public void mark(BufferedImage bufImg, Image img, Image markImg, int width, int height, int x, int y) {          Graphics2D g = bufImg.createGraphics();          g.drawImage(img, 0, 0, bufImg.getWidth(), bufImg.getHeight(), null);          g.drawImage(markImg, x, y, width, height, null);          g.dispose();      } |

　　以上两个方法实现了最核心的功能：给图片添加水印。

补充：

为什么要这么写？

考虑到该方法不仅可以用于给图片文件增加水印，而且还可以给视频的每一帧图像也添加，所以为了方便不同场合重复使用，去除了不必要的依赖关系。

有了核心的两个方法就我们可以给图片文件增加水印了

# **二、功能性实现**

[回到顶部](https://www.cnblogs.com/fengwenzhee/p/9124794.html" \l "_labelTop)

## **1、给图片增加文字水印**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42 | /\*\*       \* 给图片增加文字水印       \*       \* @param imgPath       \*            -要添加水印的图片路径       \* @param outImgPath       \*            -输出路径       \* @param text-文字       \* @param font       \*            -字体       \* @param color       \*            -颜色       \* @param x       \*            -文字位于当前图片的横坐标       \* @param y       \*            -文字位于当前图片的竖坐标       \*/      public void mark(String imgPath, String outImgPath, String text, Font font, Color color, int x, int y) {          try {              // 读取原图片信息              File imgFile = null;              Image img = null;              if (imgPath != null) {                  imgFile = new File(imgPath);              }              if (imgFile != null && imgFile.exists() && imgFile.isFile() && imgFile.canRead()) {                  img = ImageIO.read(imgFile);              }              int imgWidth = img.getWidth(null);              int imgHeight = img.getHeight(null);              // 加水印              BufferedImage bufImg = new BufferedImage(imgWidth, imgHeight, BufferedImage.TYPE\_INT\_RGB);              mark(bufImg, img, text, font, color, x, y);              // 输出图片              FileOutputStream outImgStream = new FileOutputStream(outImgPath);              ImageIO.write(bufImg, "jpg", outImgStream);              outImgStream.flush();              outImgStream.close();          } catch (Exception e) {              e.printStackTrace();          }      } |

[回到顶部](https://www.cnblogs.com/fengwenzhee/p/9124794.html" \l "_labelTop)

## **2、给图片增加图片水印**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51 | /\*\*       \* 给图片增加图片水印       \*       \* @param inputImg       \*            -源图片，要添加水印的图片       \* @param markImg       \*            - 水印图片       \* @param outputImg       \*            -输出图片(可以是源图片)       \* @param width       \*            - 水印图片宽度       \* @param height       \*            -水印图片高度       \* @param x       \*            -横坐标，相对于源图片       \* @param y       \*            -纵坐标，同上       \*/      public void mark(String inputImg, String markImg, String outputImg, int width, int height, int x, int y) {          // 读取原图片信息          File inputImgFile = null;          File markImgFile = null;          Image img = null;          Image mark = null;          try {              if (inputImg != null && markImg != null) {                  inputImgFile = new File(inputImg);                  markImgFile = new File(markImg);              }              if (inputImgFile != null && inputImgFile.exists() && inputImgFile.isFile() && inputImgFile.canRead()) {                    img = ImageIO.read(inputImgFile);                }              if (markImgFile != null && markImgFile.exists() && markImgFile.isFile() && markImgFile.canRead()) {                    mark = ImageIO.read(markImgFile);                }              int imgWidth = img.getWidth(null);              int imgHeight = img.getHeight(null);              BufferedImage bufImg = new BufferedImage(imgWidth, imgHeight, BufferedImage.TYPE\_INT\_RGB);              mark(bufImg, img, mark, width, height, x, y);              FileOutputStream outImgStream = new FileOutputStream(outputImg);              ImageIO.write(bufImg, "jpg", outImgStream);              outImgStream.flush();              outImgStream.close();          } catch (IOException e) {              e.printStackTrace();          }      } |

　　3、测试一下效果

文字水印的字体和颜色需要自行定义，这里我们使用宋体，14号字体，颜色选择橙色 - -！，坐标是x轴0，y轴等于字体的大小，也就是图片的左上角。

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | public static void main(String[] args) {          Font font = new Font("宋体", Font.PLAIN, 14);          // 原图位置, 输出图片位置, 水印文字颜色, 水印文字          // new MarkText4J().mark("eguidMarkText2.jpg", "eguidMarkText2.jpg", "水印效果测试", font, Color.ORANGE, 0, 14);          // 增加图片水印          new MarkText4J().mark("eguidMarkText2.jpg", "eguid.jpg", "eguidMarkText3.jpg", 40, 20, 0, 14);      } |

本文转载自：[https://blog.csdn.net/eguid\_1/article/details/52973508](https://blog.csdn.net/eguid_1/article/details/52973508" \t "https://www.cnblogs.com/fengwenzhee/p/_blank)

package com.sm.utils;

import java.awt.AlphaComposite;

import java.awt.Graphics2D;

import java.awt.Image;

import java.awt.RenderingHints;

import java.awt.image.BufferedImage;

import java.io.File;

import java.io.FileOutputStream;

import java.io.OutputStream;

import javax.imageio.ImageIO;

import javax.swing.ImageIcon;

/\*\*

\*

\* 生成水印

\*

\*/public class ImageMarkUtils {

/\*\* 水印透明度 \*/

private static float alpha = 0.5f;

/\*\* 水印图片旋转角度 \*/

private static double degree = 0f;

private static int interval = 0;

/\*\*

\* 设置水印参数，不设置就使用默认值

\*

\* @param alpha

\* 水印透明度

\* @param degree

\* 水印图片旋转角度 \*

\* @param interval

\* 水印图片间隔

\*/

public static void setImageMarkOptions(float alpha, int degree, int interval) {

if (alpha != 0.0f) {

ImageMarkUtils.alpha = alpha;

}

if (degree != 0f) {

ImageMarkUtils.degree = degree;

}

if (interval != 0f) {

ImageMarkUtils.interval = interval;

}

}

/\*\*

\* 给图片添加水印图片

\*

\* @param waterImgPath

\* 水印图片路径

\* @param srcImgPath

\* 源图片路径

\* @param targerPath

\* 目标图片路径

\*/

public static void waterMarkByImg(String waterImgPath, String srcImgPath, String targerPath) throws Exception {

waterMarkByImg(waterImgPath, srcImgPath, targerPath, 0);

}

/\*\*

\* 给图片添加水印图片

\*

\* @param waterImgPath

\* 水印图片路径

\* @param srcImgPath

\* 源图片路径

\* @param targerPath

\* 目标图片路径

\*/

public static void waterMarkByImg(String waterImgPath, String srcImgPath) {

try {

waterMarkByImg(waterImgPath, srcImgPath, srcImgPath, 0);

} catch (Exception e) {

// TODO Auto-generated catch block e.printStackTrace();

}

}

/\*\*

\* 给图片添加水印图片、可设置水印图片旋转角度

\*

\* @param waterImgPath

\* 水印图片路径

\* @param srcImgPath

\* 源图片路径

\* @param targerPath

\* 目标图片路径

\* @param degree

\* 水印图片旋转角度

\*/

public static void waterMarkByImg(String waterImgPath, String srcImgPath, String targerPath, double degree)

throws Exception {

OutputStream os = null;

try {

Image srcImg = ImageIO.read(new File(srcImgPath));

BufferedImage buffImg = new BufferedImage(srcImg.getWidth(null), srcImg.getHeight(null),

BufferedImage.TYPE\_INT\_RGB);

// 1、得到画笔对象

Graphics2D g = buffImg.createGraphics();

// 2、设置对线段的锯齿状边缘处理 g.setRenderingHint(RenderingHints.KEY\_INTERPOLATION, RenderingHints.VALUE\_INTERPOLATION\_BILINEAR);

g.drawImage(srcImg.getScaledInstance(srcImg.getWidth(null), srcImg.getHeight(null), Image.SCALE\_SMOOTH), 0,

0, null);

// 3、设置水印旋转

if (0 != degree) {

g.rotate(Math.toRadians(degree), (double) buffImg.getWidth() / 2, (double) buffImg.getHeight() / 2);

}

// 4、水印图片的路径 水印图片一般为gif或者png的，这样可设置透明度

ImageIcon imgIcon = new ImageIcon(waterImgPath);

// 5、得到Image对象。

Image img = imgIcon.getImage();

g.setComposite(AlphaComposite.getInstance(AlphaComposite.SRC\_ATOP, alpha));

// 6、水印图片的位置

for (int height = interval + imgIcon.getIconHeight(); height < buffImg.getHeight(); height = height

+ interval + imgIcon.getIconHeight()) {

for (int weight = interval + imgIcon.getIconWidth(); weight < buffImg.getWidth(); weight = weight

+ interval + imgIcon.getIconWidth()) {

g.drawImage(img, weight - imgIcon.getIconWidth(), height - imgIcon.getIconHeight(), null);

}

}

g.setComposite(AlphaComposite.getInstance(AlphaComposite.SRC\_OVER));

// 7、释放资源 g.dispose();

// 8、生成图片

os = new FileOutputStream(targerPath);

ImageIO.write(buffImg, "JPG", os);

System.out.println("图片完成添加水印图片");

} catch (Exception e) {

e.printStackTrace();

} finally {

try {

if (null != os)

os.close();

} catch (Exception e) {

e.printStackTrace();

}

}

}

public static void main(String[] args) {

System.out.println("..添加水印图片开始...");

/\*\*

\* watermarkPath 水印图片地址 加水印图片地址 上传成功后文件地址

\*/

// 修改默认参数

String imgPath = "C:/xym/abc.jpg";// 测试需加水印图片

String watermarkPath = "C:/xym/xinyimei.png"; // 测试水印图片

ImageMarkUtils.setImageMarkOptions(0.0f, 0, 20);

ImageMarkUtils.waterMarkByImg(watermarkPath, imgPath);

System.out.println("..添加水印图片结束...");

}

}

# [用java实现给图片增加图片水印或者文字水印（也支持视频图像帧添加水印）](https://www.cnblogs.com/fengwenzhee/p/9124794.html)

**阅读目录**

* [1、添加文字水印](https://www.cnblogs.com/fengwenzhee/p/9124794.html" \l "_label0)
* [2、添加图片水印](https://www.cnblogs.com/fengwenzhee/p/9124794.html" \l "_label1)
* [1、给图片增加文字水印](https://www.cnblogs.com/fengwenzhee/p/9124794.html" \l "_label2)
* [2、给图片增加图片水印](https://www.cnblogs.com/fengwenzhee/p/9124794.html" \l "_label3)

这是个很常用的操作，一般我们的网站在用户上传图片时都会给图片添加一个水印以防止其他站点盗图的行为

实现功能：①给图片增加文字水印②给图片增加图片水印

# **一、核心功能实现：**

[回到顶部](https://www.cnblogs.com/fengwenzhee/p/9124794.html" \l "_labelTop)

## **1、添加文字水印**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9 | // 加文字水印      public void mark(BufferedImage bufImg, Image img, String text, Font font, Color color, int x, int y) {          Graphics2D g = bufImg.createGraphics();          g.drawImage(img, 0, 0, bufImg.getWidth(), bufImg.getHeight(), null);          g.setColor(color);          g.setFont(font);          g.drawString(text, x, y);          g.dispose();      } |

[回到顶部](https://www.cnblogs.com/fengwenzhee/p/9124794.html" \l "_labelTop)

## **2、添加图片水印**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | // 加图片水印      public void mark(BufferedImage bufImg, Image img, Image markImg, int width, int height, int x, int y) {          Graphics2D g = bufImg.createGraphics();          g.drawImage(img, 0, 0, bufImg.getWidth(), bufImg.getHeight(), null);          g.drawImage(markImg, x, y, width, height, null);          g.dispose();      } |

　　以上两个方法实现了最核心的功能：给图片添加水印。

补充：

为什么要这么写？

考虑到该方法不仅可以用于给图片文件增加水印，而且还可以给视频的每一帧图像也添加，所以为了方便不同场合重复使用，去除了不必要的依赖关系。

有了核心的两个方法就我们可以给图片文件增加水印了

# **二、功能性实现**

[回到顶部](https://www.cnblogs.com/fengwenzhee/p/9124794.html" \l "_labelTop)

## **1、给图片增加文字水印**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42 | /\*\*       \* 给图片增加文字水印       \*       \* @param imgPath       \*            -要添加水印的图片路径       \* @param outImgPath       \*            -输出路径       \* @param text-文字       \* @param font       \*            -字体       \* @param color       \*            -颜色       \* @param x       \*            -文字位于当前图片的横坐标       \* @param y       \*            -文字位于当前图片的竖坐标       \*/      public void mark(String imgPath, String outImgPath, String text, Font font, Color color, int x, int y) {          try {              // 读取原图片信息              File imgFile = null;              Image img = null;              if (imgPath != null) {                  imgFile = new File(imgPath);              }              if (imgFile != null && imgFile.exists() && imgFile.isFile() && imgFile.canRead()) {                  img = ImageIO.read(imgFile);              }              int imgWidth = img.getWidth(null);              int imgHeight = img.getHeight(null);              // 加水印              BufferedImage bufImg = new BufferedImage(imgWidth, imgHeight, BufferedImage.TYPE\_INT\_RGB);              mark(bufImg, img, text, font, color, x, y);              // 输出图片              FileOutputStream outImgStream = new FileOutputStream(outImgPath);              ImageIO.write(bufImg, "jpg", outImgStream);              outImgStream.flush();              outImgStream.close();          } catch (Exception e) {              e.printStackTrace();          }      } |

[回到顶部](https://www.cnblogs.com/fengwenzhee/p/9124794.html" \l "_labelTop)

## **2、给图片增加图片水印**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51 | /\*\*       \* 给图片增加图片水印       \*       \* @param inputImg       \*            -源图片，要添加水印的图片       \* @param markImg       \*            - 水印图片       \* @param outputImg       \*            -输出图片(可以是源图片)       \* @param width       \*            - 水印图片宽度       \* @param height       \*            -水印图片高度       \* @param x       \*            -横坐标，相对于源图片       \* @param y       \*            -纵坐标，同上       \*/      public void mark(String inputImg, String markImg, String outputImg, int width, int height, int x, int y) {          // 读取原图片信息          File inputImgFile = null;          File markImgFile = null;          Image img = null;          Image mark = null;          try {              if (inputImg != null && markImg != null) {                  inputImgFile = new File(inputImg);                  markImgFile = new File(markImg);              }              if (inputImgFile != null && inputImgFile.exists() && inputImgFile.isFile() && inputImgFile.canRead()) {                    img = ImageIO.read(inputImgFile);                }              if (markImgFile != null && markImgFile.exists() && markImgFile.isFile() && markImgFile.canRead()) {                    mark = ImageIO.read(markImgFile);                }              int imgWidth = img.getWidth(null);              int imgHeight = img.getHeight(null);              BufferedImage bufImg = new BufferedImage(imgWidth, imgHeight, BufferedImage.TYPE\_INT\_RGB);              mark(bufImg, img, mark, width, height, x, y);              FileOutputStream outImgStream = new FileOutputStream(outputImg);              ImageIO.write(bufImg, "jpg", outImgStream);              outImgStream.flush();              outImgStream.close();          } catch (IOException e) {              e.printStackTrace();          }      } |

　　3、测试一下效果

文字水印的字体和颜色需要自行定义，这里我们使用宋体，14号字体，颜色选择橙色 - -！，坐标是x轴0，y轴等于字体的大小，也就是图片的左上角。

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | public static void main(String[] args) {          Font font = new Font("宋体", Font.PLAIN, 14);          // 原图位置, 输出图片位置, 水印文字颜色, 水印文字          // new MarkText4J().mark("eguidMarkText2.jpg", "eguidMarkText2.jpg", "水印效果测试", font, Color.ORANGE, 0, 14);          // 增加图片水印          new MarkText4J().mark("eguidMarkText2.jpg", "eguid.jpg", "eguidMarkText3.jpg", 40, 20, 0, 14);      } |

本文转载自：[https://blog.csdn.net/eguid\_1/article/details/52973508](https://blog.csdn.net/eguid_1/article/details/52973508" \t "https://www.cnblogs.com/fengwenzhee/p/_blank)

package com.sm.utils;

import java.awt.AlphaComposite;

import java.awt.Graphics2D;

import java.awt.Image;

import java.awt.RenderingHints;

import java.awt.image.BufferedImage;

import java.io.File;

import java.io.FileOutputStream;

import java.io.OutputStream;

import javax.imageio.ImageIO;

import javax.swing.ImageIcon;

/\*\*

\*

\* 生成水印

\*

\*/public class ImageMarkUtils {

/\*\* 水印透明度 \*/

private static float alpha = 0.5f;

/\*\* 水印图片旋转角度 \*/

private static double degree = 0f;

private static int interval = 0;

/\*\*

\* 设置水印参数，不设置就使用默认值

\*

\* @param alpha

\* 水印透明度

\* @param degree

\* 水印图片旋转角度 \*

\* @param interval

\* 水印图片间隔

\*/

public static void setImageMarkOptions(float alpha, int degree, int interval) {

if (alpha != 0.0f) {

ImageMarkUtils.alpha = alpha;

}

if (degree != 0f) {

ImageMarkUtils.degree = degree;

}

if (interval != 0f) {

ImageMarkUtils.interval = interval;

}

}

/\*\*

\* 给图片添加水印图片

\*

\* @param waterImgPath

\* 水印图片路径

\* @param srcImgPath

\* 源图片路径

\* @param targerPath

\* 目标图片路径

\*/

public static void waterMarkByImg(String waterImgPath, String srcImgPath, String targerPath) throws Exception {

waterMarkByImg(waterImgPath, srcImgPath, targerPath, 0);

}

/\*\*

\* 给图片添加水印图片

\*

\* @param waterImgPath

\* 水印图片路径

\* @param srcImgPath

\* 源图片路径

\* @param targerPath

\* 目标图片路径

\*/

public static void waterMarkByImg(String waterImgPath, String srcImgPath) {

try {

waterMarkByImg(waterImgPath, srcImgPath, srcImgPath, 0);

} catch (Exception e) {

// TODO Auto-generated catch block e.printStackTrace();

}

}

/\*\*

\* 给图片添加水印图片、可设置水印图片旋转角度

\*

\* @param waterImgPath

\* 水印图片路径

\* @param srcImgPath

\* 源图片路径

\* @param targerPath

\* 目标图片路径

\* @param degree

\* 水印图片旋转角度

\*/

public static void waterMarkByImg(String waterImgPath, String srcImgPath, String targerPath, double degree)

throws Exception {

OutputStream os = null;

try {

Image srcImg = ImageIO.read(new File(srcImgPath));

BufferedImage buffImg = new BufferedImage(srcImg.getWidth(null), srcImg.getHeight(null),

BufferedImage.TYPE\_INT\_RGB);

// 1、得到画笔对象

Graphics2D g = buffImg.createGraphics();

// 2、设置对线段的锯齿状边缘处理 g.setRenderingHint(RenderingHints.KEY\_INTERPOLATION, RenderingHints.VALUE\_INTERPOLATION\_BILINEAR);

g.drawImage(srcImg.getScaledInstance(srcImg.getWidth(null), srcImg.getHeight(null), Image.SCALE\_SMOOTH), 0,

0, null);

// 3、设置水印旋转

if (0 != degree) {

g.rotate(Math.toRadians(degree), (double) buffImg.getWidth() / 2, (double) buffImg.getHeight() / 2);

}

// 4、水印图片的路径 水印图片一般为gif或者png的，这样可设置透明度

ImageIcon imgIcon = new ImageIcon(waterImgPath);

// 5、得到Image对象。

Image img = imgIcon.getImage();

g.setComposite(AlphaComposite.getInstance(AlphaComposite.SRC\_ATOP, alpha));

// 6、水印图片的位置

for (int height = interval + imgIcon.getIconHeight(); height < buffImg.getHeight(); height = height

+ interval + imgIcon.getIconHeight()) {

for (int weight = interval + imgIcon.getIconWidth(); weight < buffImg.getWidth(); weight = weight

+ interval + imgIcon.getIconWidth()) {

g.drawImage(img, weight - imgIcon.getIconWidth(), height - imgIcon.getIconHeight(), null);

}

}

g.setComposite(AlphaComposite.getInstance(AlphaComposite.SRC\_OVER));

// 7、释放资源 g.dispose();

// 8、生成图片

os = new FileOutputStream(targerPath);

ImageIO.write(buffImg, "JPG", os);

System.out.println("图片完成添加水印图片");

} catch (Exception e) {

e.printStackTrace();

} finally {

try {

if (null != os)

os.close();

} catch (Exception e) {

e.printStackTrace();

}

}

}

public static void main(String[] args) {

System.out.println("..添加水印图片开始...");

/\*\*

\* watermarkPath 水印图片地址 加水印图片地址 上传成功后文件地址

\*/

// 修改默认参数

String imgPath = "C:/xym/abc.jpg";// 测试需加水印图片

String watermarkPath = "C:/xym/xinyimei.png"; // 测试水印图片

ImageMarkUtils.setImageMarkOptions(0.0f, 0, 20);

ImageMarkUtils.waterMarkByImg(watermarkPath, imgPath);

System.out.println("..添加水印图片结束...");

}

}

## 使用java或jsp给图片添加水印代码

 (2006-08-24 14:11:36)

[IMG_256转载▼](http://blog.sina.com.cn/s/javascript:;)

|  |  |
| --- | --- |
|  |  |

/\*  
 \* 创建日期 2006-8-24  
 \*  
 \* TODO 要更改此生成的文件的模板，请转至  
 \* 窗口 － 首选项 － Java － 代码样式 － 代码模板  
 \*/  
package com.et.yizh.zhp;

import java.awt.Color;  
import java.awt.Graphics;  
import java.awt.Graphics2D;  
import java.awt.Image;  
import java.awt.image.BufferedImage;  
import java.io.File;  
import java.io.FileOutputStream;

import javax.swing.\*;  
import com.sun.image.codec.jpeg.\*;

import com.sun.image.codec.jpeg.JPEGImageEncoder;

/\*\*  
 \* @author Administrator  
 \*  
 \* TODO 要更改此生成的类型注释的模板，请转至  
 \* 窗口 － 首选项 － Java － 代码样式 － 代码模板  
 \*  
 \* 添加水印,  
 \* filePath 源图片路径，  
 \* watermark 水印图片路径  
 \* savePath 为你添加水印后的图片保存路径文件夹  
 \*/  
//添加水印,filePath 源图片路径， watermark 水印图片路径  
public class Mark {

    public static boolean createMark(String filePath,String watermark,String savePath) {

        ImageIcon imgIcon=new ImageIcon(filePath);

        Image theImg =imgIcon.getImage();

        ImageIcon waterIcon=new ImageIcon(watermark);

        Image waterImg =waterIcon.getImage();  
         
        ///////////////////////////////////////////////////////////////////////  
        File f = new File(filePath);  
         
        String picname = f.getName();//取得图片名  
         
        ImageIcon markIcon=new ImageIcon(watermark); //要添加的水印图标

        Image markImg =markIcon.getImage();  
         
        int wid=markImg.getWidth(null); //水印图标宽度

        int het= markImg.getHeight(null); //水印图标高度  
         
        //////////////////////////////////////////////////////////////////////

        int width=theImg.getWidth(null); //源图片宽度

        int height= theImg.getHeight(null); //源图片高度

        BufferedImage bimage = new BufferedImage(width,height, BufferedImage.TYPE\_INT\_RGB);

        Graphics2D g=bimage.createGraphics( );

        g.setColor(Color.red); //设置颜色

        g.setBackground(Color.white);

        g.drawImage(theImg, 0, 0, null );

        g.drawImage(waterImg, width-wid+5, height-het+5, null ); //添加图标中间两个数字参数 是设定位置

        g.drawString("12233",100,10); //添加文字

        try{

        FileOutputStream out=new FileOutputStream(savePath+"[\\"+picname](http://blog.sina.com.cn/s/));

        JPEGImageEncoder encoder =JPEGCodec.createJPEGEncoder(out);

        JPEGEncodeParam param = encoder.getDefaultJPEGEncodeParam(bimage);

        param.setQuality(50f, true); //图片质量

        encoder.encode(bimage, param);

        out.close();

        }catch(Exception e){  
             
                e.printStackTrace();  
                System.out.println("===========失败");  
                return false;  
              
            }  
         
            System.out.println("===========成功");  
            return true;  
          }  
    ///测试主程序  
        public static void main(String[] args){  
              createMark("E:\\photo\\image3.jpg","E:\\photo\\image2.jpg","E:\\dabase");

          }

}

## 关于改良JAVA程序实现给图片加水印效果的问题！！

[y\_\_smiling](https://wenda.so.com/u/1902866105" \t "https://wenda.so.com/q/_blank)11级分类：[编程开发](https://wenda.so.com/c/20)被浏览303次2013.06.12

下面是个 给图片加载水印效果的JAVA程序 。 但是输入和输出都是常量形式，我想将它作成个button，让用户自己添加水印效果，添加字。 希望JAVA高手指点…… import java.awt.\*;  
import java.awt.image.\*;  
import java.io.\*;  
import javax.swing.\*;  
import com.sun.image.codec.jpeg.\*; //此包水印，要求输出JPG格式  
public class ImageMark  
{  
private static final String SOURCE\_JPG="source.jpg";  
private static final String MARK\_JPG="mark.jpg";  
private static final String DEST1="文字水印.jpg";  
private static final String DEST2="图片水印.jpg";  
public boolean createMark(String sImgFile,String sMarkContent,Color sMarkContentColor,float fQual)  
{  
ImageIcon imgIcon=new ImageIcon(sImgFile);  
Image theImg=imgIcon.getImage();  
int width=theImg.getWidth(null);  
int height=theImg.getHeight(null);  
BufferedImage bimage=new BufferedImage(width,height,BufferedImage.TYPE\_INT\_RGB);  
Graphics2D g=bimage.createGraphics();  
g.setColor(sMarkContentColor);  
g.setBackground(Color.white);  
g.drawImage(theImg,0,0,null);  
g.setFont(new Font(null,Font.BOLD,32)); //字体字形字号  
g.drawString(sMarkContent,width/10,height/10); //画文字  
g.dispose();  
try  
{  
FileOutputStream out=new FileOutputStream(DEST1);  
JPEGImageEncoder encoder=JPEGCodec.createJPEGEncoder(out);  
JPEGEncodeParam param=encoder.getDefaultJPEGEncodeParam(bimage);  
param.setQuality(fQual,true);  
encoder.encode(bimage,param);  
out.close();  
}  
catch(Exception e)  
{return false;}  
return true;  
  
}  
//重载形式2：添加图片水印  
public boolean createMark(String sImgFile,float fQual)  
{  
//要处理的原始图片  
ImageIcon icoInput=new ImageIcon(sImgFile);  
Image imgInput=icoInput.getImage();  
int width=imgInput.getWidth(null);  
int height=imgInput.getHeight(null);  
BufferedImage buffInput=new BufferedImage(width,height,BufferedImage.TYPE\_INT\_RGB);  
//要添加上来的水印  
ImageIcon icoADD=new ImageIcon(MARK\_JPG);  
Image imgADD=icoADD.getImage();  
int w=imgADD.getWidth(null);  
int h=imgADD.getHeight(null);  
//绘图  
Graphics2D g=buffInput.createGraphics();  
g.drawImage(imgInput,0,0,null);  
g.drawImage(imgADD,10,10,w,h,null);  
g.dispose();  
try  
{  
FileOutputStream out=new FileOutputStream(DEST2);  
JPEGImageEncoder encoder=JPEGCodec.createJPEGEncoder(out);  
JPEGEncodeParam param=encoder.getDefaultJPEGEncodeParam(buffInput);  
param.setQuality(fQual,true);  
encoder.encode(buffInput,param);  
out.close();  
  
}  
catch (Exception e)  
{return false;}  
return true;  
}  
public static void main(String arg[])  
{  
ImageMark wmObj=new ImageMark();  
//文字水印  
if (wmObj.createMark(SOURCE\_JPG,"茜茜水印",Color.red,5)==true)  
System.out.println("succ!查看生成的"+DEST1+"!");  
else  
System.out.println("文字水印执行失败!");  
//图片水印  
if (wmObj.createMark(SOURCE\_JPG,3)==true)  
System.out.println("succ!查看生成的"+DEST1+"!");  
else  
System.out.println("图片水印执行失败!");  
System.exit(0);  
}  
}

## 满意答案

[IMG_256](https://wenda.so.com/u/1904523158)

[iloveyouka41](https://wenda.so.com/u/1904523158" \t "https://wenda.so.com/q/_blank)

采纳率：58%13级 2013.06.12

以下[文字](http://www.so.com/s?q=%E4%BB%A3%E7%A0%81&ie=utf-8&src=internal_wenda_recommend_textn" \t "https://wenda.so.com/q/_blank)内链\_点击实体词" target="\_blank">代码实现了自由选择源图片并输入[水印](http://www.so.com/s?q=%E6%B0%B4%E5%8D%B0&ie=utf-8&src=internal_wenda_recommend_textn" \t "https://wenda.so.com/q/_blank)文字的功能  
其他[参数](http://www.so.com/s?q=%E5%8F%82%E6%95%B0&ie=utf-8&src=internal_wenda_recommend_textn" \t "https://wenda.so.com/q/_blank)的输入请参照下面的类进行修改import java.awt.Color;  
import java.awt.Container;  
import java.awt.event.ActionEvent;  
import java.awt.event.ActionListener;import javax.swing.JButton;  
import javax.swing.JFileChooser;  
import javax.swing.JFrame;  
import javax.swing.JLabel;  
import javax.swing.JOptionPane;  
import javax.swing.JPanel;  
import javax.swing.JTextField;import com.test.ImageMark;//注意需要引入 ImageMark 类public class TestButton {  
public static void main(String[] args) {  
ButtonFrame frame = new ButtonFrame();  
frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);  
frame.setVisible(true);  
}  
}class ButtonFrame extends JFrame {public static final int DEFAULT\_WIDTH = 400;  
public static final int DEFAULT\_HEIGHT = 150;public ButtonFrame() {  
setTitle("文字水印");  
setSize(DEFAULT\_WIDTH, DEFAULT\_HEIGHT);// add panel to frame  
ButtonPanel panel = new ButtonPanel();  
Container contentPane = getContentPane();  
contentPane.add(panel);  
}  
}class ButtonPanel extends JPanel {JLabel lable1 = null;  
JLabel lable2 = null;  
JTextField file = null;  
JTextField text = null;  
JButton button1 = null;  
JButton button2 = null;  
JFileChooser fDialog = null;public ButtonPanel() {  
fDialog = new JFileChooser();  
button1 = new JButton("选择图片");  
lable1 = new JLabel("图片路径：");  
lable2 = new JLabel("水印文字：");  
file = new JTextField(20);  
text = new JTextField(20);  
button2 = new JButton("生成水印");  
add(lable1);  
add(file);  
add(button1);  
add(lable2);  
add(text);  
add(button2);ButtonAction button1Action = new ButtonAction(1);  
ButtonAction button2Action = new ButtonAction(2);button1.addActionListener(button1Action);  
button2.addActionListener(button2Action);}private class ButtonAction implements ActionListener {int Type = 0;public ButtonAction(int type) {  
Type = type;  
}public void actionPerformed(ActionEvent event) {if (Type == 2) {if (null == text.getText() || "".equals(text.getText())) {  
JOptionPane.showMessageDialog(null, "请输入水印文字", "错误提示",  
JOptionPane.ERROR\_MESSAGE);  
return;  
}if (null == file.getText() || "".equals(file.getText())) {  
JOptionPane.showMessageDialog(null, "请选择源图片", "错误提示",  
JOptionPane.ERROR\_MESSAGE);  
return;  
}  
ImageMark wmObj = new ImageMark();  
String SOURCE\_JPG = "source.jpg";  
// 文字水印  
if (wmObj.createMark(file.getText(), text.getText(), Color.red,  
5) == true) {  
JOptionPane.showMessageDialog(null, "文字水印执行成功!", "成功提示",  
JOptionPane.INFORMATION\_MESSAGE);  
text.setText("");  
file.setText("");  
} else {  
JOptionPane.showMessageDialog(null, "文字水印执行失败!", "错误提示",  
JOptionPane.ERROR\_MESSAGE);  
}} else {JFileChooser fileChooser = new JFileChooser();int result = fileChooser.showOpenDialog(null);  
if (result == JFileChooser.APPROVE\_OPTION) {  
String filePath = fileChooser.getSelectedFile().getPath();  
file.setText(filePath);  
}  
}  
}  
}  
}