import java.applet.Applet;

import java.awt.Graphics;

import java.awt.Graphics2D;

import java.awt.Image;

import java.awt.MediaTracker;

import java.awt.RenderingHints;

import java.awt.color.ColorSpace;

import java.awt.event.ActionEvent;

import java.awt.image.BufferedImage;

import java.awt.image.ColorConvertOp;

import java.awt.image.RenderedImage;

import java.io.File;

import java.io.IOException;

import java.net.URL;

import javax.imageio.ImageIO;

import javax.swing.ImageIcon;

import javax.swing.JComboBox;

import javax.swing.JFileChooser;

import javax.swing.JPanel;

import java.awt.Graphics;

import java.awt.Graphics2D;

import java.awt.Image;

import java.awt.Toolkit;

import java.awt.image.BufferedImage;

import java.io.File;

import java.io.IOException;

import java.net.URL;

import java.util.logging.Level;

import java.util.logging.Logger;

import javax.imageio.ImageIO;

public class Affichage extends Applet{

private Image image=null;

public BufferedImage toBufferedImage(Image image) {

/\*\* On test si l'image n'est pas déja une instance de BufferedImage \*/

if( image instanceof BufferedImage ) {

return( (BufferedImage)image );

} else {

/\*\* On s'assure que l'image est complètement chargée \*/

image = new ImageIcon(image).getImage();

/\*\* On crée la nouvelle image \*/

BufferedImage bufferedImage = new BufferedImage(

image.getWidth(null),

image.getHeight(null),

BufferedImage.TYPE\_INT\_RGB );

Graphics g = bufferedImage.createGraphics();

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

g.dispose();

return( bufferedImage );

}

}

public Image getImage(String path){

Image tempImage= null;

try{

URL imageURL = Affichage.class.getResource(path);

tempImage = Toolkit.getDefaultToolkit().getImage(imageURL);

}

catch(Exception e){

System.out.println("An error "+e.getMessage());

}

return tempImage;

}

public void paint(Graphics g) {

this.setSize(500, 400);

if(image ==null) image =getImage("bugatti.jpg");

ColorConvertOp op = new ColorConvertOp(

ColorSpace.getInstance(ColorSpace.CS\_GRAY),

null);

BufferedImage imageGrise = op.filter(toBufferedImage(image),null);

Graphics2D g2 =(Graphics2D)g;

g2.drawImage(imageGrise, 0, 0, 300, 300, this);

super.paint(g);

}

}