/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Ã€ RETIRER DANS VOTRE PROJET \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\* Code de dÃ©monstration de l'usage de cette classe \*\*/

public static void main( String[] args )

{

try {

if (args.length!=1) throw new IllegalArgumentException("PrÃ©ciser le nom (et chemin) d'un unique fichier png en paramÃ¨tre");

ImagePNG png = new ImagePNG(args[0]);

ImagePNG pngNB = new ImagePNG(png); // une copie

ImagePNG pngR = new ImagePNG(png); // une autre copie

// 1. MODIFICATION DES COULEURS

Color col,nb,red;

int r,g,b,l;

for (int x=0 ; x<png.width(); x++)

{

for (int y=0 ; y<png.height() ; y++)

{

// extraction de la couleur du pixel (x,y)

col = png.getPixel(x,y);

// modification de la couleur

r = col.getRed();

g = col.getGreen();

b = col.getBlue();

l = (r+g+b)/3;

nb = new Color(l,l,l);

red = new Color(l,0,0);

// inscription de la nouvelle couleur des pixels (x,y)

pngNB.setPixel(x,y,nb);

pngR.setPixel(x,y,red);

}

}

png.save(args[0]);

pngNB.save(args[0]+"NB");

pngR.save(args[0]+"R");

// 2. COMPARAISON DES RÃ‰SULTATS

// calcul des indices de similaritÃ©

double siNB = ImagePNG.computeEQM(png,pngNB);

double siR = ImagePNG.computeEQM(png,pngR);

// chargement des fichiers

File fic = new File(args[0]);

File ficNB = new File(args[0]+"NB");

File ficR = new File(args[0]+"NB");

// rapport des tailles

double wNB = Math.ceil(10000.0\*ficNB.length() / fic.length())/100.0;

double wR = Math.ceil(10000.0\*ficR.length() / fic.length())/100.0;

// affichage des critÃ¨res

System.out.println("NB: taille="+wNB+"% / qualitÃ©="+siNB+"%");

System.out.println("R: taille="+wR+"% / qualitÃ©="+siR+"%");

} catch (Exception e) {

e.printStackTrace();

}

}