1. **package** com.yang.rm.system;
3. **import** java.awt.AWTException;
4. **import** java.awt.Dimension;
5. **import** java.awt.Rectangle;
6. **import** java.awt.Robot;
7. **import** java.awt.Toolkit;
8. **import** java.awt.image.BufferedImage;
9. **import** java.io.File;
11. **import** javax.imageio.ImageIO;
13. **public** **class** ReadColorTest {
14. /\*\*
15. \* 读取一张图片的RGB值
16. \*
17. \* @throws Exception
18. \*/
19. **public** **void** getImagePixel(String image) **throws** Exception {
20. **int**[] rgb = **new** **int**[3];
21. File file = **new** File(image);
22. BufferedImage bi = **null**;
23. **try** {
24. bi = ImageIO.read(file);
25. } **catch** (Exception e) {
26. e.printStackTrace();
27. }
28. **int** width = bi.getWidth();
29. **int** height = bi.getHeight();
30. **int** minx = bi.getMinX();
31. **int** miny = bi.getMinY();
32. System.out.println("width=" + width + ",height=" + height + ".");
33. System.out.println("minx=" + minx + ",miniy=" + miny + ".");
34. **for** (**int** i = minx; i < width; i++) {
35. **for** (**int** j = miny; j < height; j++) {
36. **int** pixel = bi.getRGB(i, j); // 下面三行代码将一个数字转换为RGB数字
37. rgb[0] = (pixel & 0xff0000) >> 16;
38. rgb[1] = (pixel & 0xff00) >> 8;
39. rgb[2] = (pixel & 0xff);
40. System.out.println("i=" + i + ",j=" + j + ":(" + rgb[0] + ","
41. + rgb[1] + "," + rgb[2] + ")");
42. }
43. }
44. }
46. /\*\*
47. \* 返回屏幕色彩值
48. \*
49. \* @param x
50. \* @param y
51. \* @return
52. \* @throws AWTException
53. \*/
54. **public** **int** getScreenPixel(**int** x, **int** y) **throws** AWTException { // 函数返回值为颜色的RGB值。
55. Robot rb = **null**; // java.awt.image包中的类，可以用来抓取屏幕，即截屏。
56. rb = **new** Robot();
57. Toolkit tk = Toolkit.getDefaultToolkit(); // 获取缺省工具包
58. Dimension di = tk.getScreenSize(); // 屏幕尺寸规格
59. System.out.println(di.width);
60. System.out.println(di.height);
61. Rectangle rec = **new** Rectangle(0, 0, di.width, di.height);
62. BufferedImage bi = rb.createScreenCapture(rec);
63. **int** pixelColor = bi.getRGB(x, y);
65. **return** 16777216 + pixelColor; // pixelColor的值为负，经过实践得出：加上颜色最大值就是实际颜色值。
66. }
68. /\*\*
69. \* @param args
70. \*/
71. **public** **static** **void** main(String[] args) **throws** Exception {
72. **int** x = 0;
73. ReadColorTest rc = **new** ReadColorTest();
74. x = rc.getScreenPixel(100, 345);
75. System.out.println(x + " - ");
76. rc.getImagePixel("D:\\car.jpg");
77. }
79. }

**[javascript]** [view plain](http://blog.csdn.net/ubuntu_yanglei/article/details/46443929" \o "view plain) [copy](http://blog.csdn.net/ubuntu_yanglei/article/details/46443929" \o "copy) [IMG_256](https://code.csdn.net/snippets/1613828)[IMG_257](https://code.csdn.net/snippets/1613828/fork)

1. i=76,j=554:(189,157,118)
2. i=76,j=555:(171,139,100)
3. i=76,j=556:(176,144,105)
4. i=76,j=557:(183,153,117)
5. i=76,j=558:(172,145,115)
6. i=76,j=559:(114,87,57)
7. i=76,j=560:(156,127,93)
8. i=76,j=561:(173,143,105)
9. i=76,j=562:(164,132,94)
10. i=76,j=563:(170,138,99)
11. i=76,j=564:(170,136,99)
12. i=76,j=565:(167,135,97)
13. i=76,j=566:(174,144,108)
14. i=76,j=567:(158,128,90)
15. i=76,j=568:(183,151,112)
16. i=76,j=569:(185,153,115)
17. i=76,j=570:(206,177,145)
18. i=76,j=571:(134,107,78)
19. i=76,j=572:(113,88,58)
20. i=76,j=573:(190,165,135)
21. i=76,j=574:(154,130,104)
22. i=76,j=575:(94,68,41)
23. i=76,j=576:(182,151,120)
24. i=76,j=577:(190,157,122)
25. i=76,j=578:(170,137,102)
26. i=76,j=579:(167,136,105)
27. i=76,j=580:(149,123,98)
28. i=76,j=581:(91,69,46)
29. i=76,j=582:(134,109,89)
30. i=76,j=583:(139,114,94)
31. i=76,j=584:(127,102,80)
32. i=76,j=585:(126,101,81)
33. i=76,j=586:(111,85,68)
34. i=76,j=587:(111,85,70)
35. i=76,j=588:(113,88,68)
36. i=76,j=589:(119,94,72)
37. i=76,j=590:(142,118,92)
38. i=76,j=591:(140,116,88)
39. i=76,j=592:(151,127,101)
40. i=76,j=593:(86,64,41)
41. i=76,j=594:(103,82,65)
42. i=76,j=595:(114,93,76)
43. i=76,j=596:(110,85,63)
44. i=76,j=597:(166,139,110)
45. i=76,j=598:(172,142,108)
46. i=76,j=599:(139,107,69)
47. i=76,j=600:(173,141,100)
48. i=76,j=601:(205,172,131)
49. i=76,j=602:(142,108,71)
50. i=76,j=603:(201,168,133)
51. i=76,j=604:(165,134,103)
52. i=76,j=605:(136,109,79)
53. i=76,j=606:(147,122,92)
54. i=76,j=607:(144,118,91)
55. i=76,j=608:(142,116,93)
56. i=76,j=609:(140,114,91)
57. i=76,j=610:(121,94,67)
58. i=76,j=611:(168,139,109)
59. i=76,j=612:(159,126,91)
60. i=76,j=613:(167,133,95)
61. i=76,j=614:(191,154,110)
62. i=76,j=615:(197,159,120)
63. i=76,j=616:(198,167,136)
64. i=76,j=617:(102,76,51)
65. i=76,j=618:(110,89,72)
66. i=76,j=619:(116,97,83)
67. i=76,j=620:(127,109,95)
68. i=76,j=621:(92,74,62)
69. i=76,j=622:(102,85,75)
70. i=76,j=623:(94,80,71)
71. i=76,j=624:(96,81,74)
72. i=76,j=625:(95,80,73)
73. i=76,j=626:(107,90,82)
74. i=76,j=627:(99,82,74)
75. i=76,j=628:(100,83,73)
76. i=76,j=629:(110,93,83)
77. i=76,j=630:(105,88,78)
78. i=76,j=631:(104,90,81)