<< به نام او >>

ثمین انصاری

پروژه مبانی کامپیوتر : نهان سازی متن در تصویر

استاد :میلاد مظفری

با تشکر از استاد برای ترم خیلی خوب D:

<<محتوای ایمیل>>

این برنامه شامل یک برنامه Main است و دو تابع TextHider و Textextracter که تابع اول هدفش مخفی کردن عکس و خروجی دیگری به عنوان key.bmp میدهد و کلاس دوم عکس و کلید را دریافت و متن مورد نظر را به صورت خروجی بیرون میدهد .

<<TextHider>>

این کلاس یک constructer دارد به صورتی که ادرس فایل متن را دریافت میکند .

و یک تابع به صورتی که ، عکس ورودی و عکس خروجی را دریافت میکند و در انتها عکسی هم به عنوان key.bmp ایجاد میکند !

در اول inputImage را بارگذاری کردیم و دو تصویر خام outputImage و keyImage تولید کردیم .

***در ابتدا باید حروف متن ورودی را به صورت کد اسکی و کد اسکی را در مبنای ۲ بنویسیم (خط ۳۰ تا ۵۰) : ابتدا از فایل مورد نظر متن را میخوانیم و بدین صورت که اسپیس را با \r\n نشان میدهیم (یعنی اسپیس بدین صورت نوشته میشود-> به صورت دو حرف)
string b را به عنوان خروجی این قسمت تعریف میکنیم !

chara کد اسکیه هریک از حروف متن را نمایش میدهد .

Exception گفته شده در که اگر اندازه مبنای ۲ کد اسکی متن بیشتر از سه چهارمه پیکسل های موجود باشد !! (خط ۵۳ تا ۵۴) ***

*** (خط ۵۷ تا ۶۹)انتخاب پیکسل های رندوم : یک ArrayList با نام rand تعریف میکنیم تا عدد های رندوم را داخل ان نگه داری میکنیم !

بدین صورت عمل کرده ایم که به هر پیکسله تصویر شماره ای اختصاص دادیم از صفر تا (1-(x*y)) مانند مثال زیر :

۰	۱	۲

۵	۴	۳
۸	۷	۶
۱۱	۱۰	۹

که از این ها به اندازه بیت های استرینگ b رندوم انتخاب میکنیم و در ArrayList rand قرار میدهیم ، حال ۲ نکته وجود دارد : ۱. ممکن است عدد های رندوم تکراری شوند
۲.مرتب نیستند

برای رفع گزینه ۱ ، Boolean tekrar تعریف کردیم به صورتی که اگر عدد رندوم جدید انتخاب شده قبلا در ArrayLlst rand وجود داشت Boolean tekrar برابر با false میشود و ان عدد رندوم در ArrayList rand اضافه نمیگردد .

برای رفع گزینه ۲ ، از Libary خود جاوا برای مرتب کردن خانه های ArrayLIst مورد نظر استفاده کردیم .

***(خط ۷۳ تا ۱۰۲) تشکیل مختصات پیکسل های رندوم در outputImage و keyImage :

برای هر پیکسل :اول k را برای انتخاب رندوم بین رنگ های قرمز(۰) ، سبز (۱) ، ابی(۲) انتخاب میکنیم به صورت رندوم

بر اساس k ی انتخاب شده سه تا ifداریم که rgb ان پیکسل را تشکیل میدهیم بدین صورت ک : به طور مثال :

```
int green = (rgboutput.getGreen() & ~1) | Character.getNumericValue(b.charAt(i));
            rgbkey = new Color(0, 255, 0);
            rgboutput = new Color(rgboutput.getRed(), green, rgboutput.getBlue());
```

rgb سبز را نگه میدارد جز اخرین بیت ان که با توجه به ۰ یا ۱ متن که باید جاگذاری شود انتخاب میشود .
و rgboutput را مانند تصویر انتخاب میکند به جز قسمت سبز !!
همچنین برای key هم فقط قسمت سبز ان را ۲۵۶ میگذارد سایر قسمت ها صفر میشود .
و این rgb های تولید شده را در مختصات در مختصات ان پیکسل جای گذاری میکند .
و بدین صورت همین عمل به اندازه بیت های مورد نیاز ادامه میابد .

*** (خط های اخر) تشکیل مختصات پیکسل های غیر رندوم در outputImage و keyImage:

در keyImage که مختصات سایره نقاط توسط کامپیوتر به صورت فرضی صفر در نظر گرفته شده است .
و برای outputImage همان rgb عکس اصلی استفاده میشود .

*** و در اخر عکس ذخیره میشود.

<<TextHider>>

*** این کلاس یک constructer دارد به صورتی که ادرس عکس input را دریافت میکند .

*** و یک تابع به صورتی که ادرس متن و عکس کلید را دریافت میکند و در انتها فایل متنی ایجاد میکند !

در اول inputImage و keyImageرا بارگذاری کردیم .

PrintWriter out = new PrintWriter(textAddress); ***
این دستور برای ایجاد نتیجه به صورت فایل جدید است .

*** String result تعریف شده برای نتیجه انتهایی
binary result برای نتیجه های ۸ تایی که مربوط به یک حرف میشوند

پس ان بیت های تغیر یافته را که با توجه به دستور هایی که داریم از قبیلcolor میتوان با سه ایف بدست اورد

و ۸ تا از این بیت هارا در باید جدا کرده و با توجه به کد اسکی ان حرف مورد نظر را به String result اضافه کرد

####توضیحات بیشتر از دریافت کد اسکی : با توجه به keyImage ای که داریم با سه نوع if میتوانیم بفهمیم که کدام یک از رنگ های قرمز یا سبز یا ابی تغییر یافته است و با توجه با ان قسمت مورد نظر (قرمز یا سبز یا ابی) از inputImage را گرفته و & اش را با یک اگر محاسبه کنیم اخرین بیت ان درمیاید !!!