Lecture3 ADT、颜色、图片和图形

1. 抽象数据类型 ADT

数据类型介绍

数据类型是一组值和对这些值的一组操作

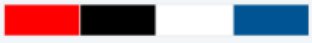
基本数据类型 Primitive types

- 值直接被映射到机器码表示
- 操作符直接被映射到机器语言

type	*set of values*	*examples of values*	*examples of operations*
`char`	characters	'A' '@'	compare
`String`	sequences of characters	"Hello World" "CS is fun"	concatenate
`int`	integers	17 12345	add, subtract, multiply, divide
`double`	floating-point numbers	3.1415 6.022e23	add, subtract, multiply, divide
`boolean`	truth values	true false	and, or, not

抽象数据类型 Abstract data types

- 抽象数据类型是对客户端隐藏其表示的数据类型
- 颜色、图片、字符串、复杂的数字、向量、矩阵等
- 客户端可以再不知道实现细节的情况下使用 ADT
    - 如何为几个有用的 ADT 编写客户端程序
    - 如何实现自己的 ADT

data type	*set of values*	*examples of operations*	
`Color`	three 8-bit integers	get red component, brighten	
`Picture`	2D array of colors	get/set color of pixel (i, j)	
`String`	sequence of characters	length, substring, compare	C A T A G C G C

String ADT

- 字符串是以 Unicode 编码的字符序列
- Java 的 ADT 允许我们编写操作字符串的 Java 程序

public class String	
String(String s)	*create a string with the same value*
int length()	*string length*
char charAt(int i)	*ith character*
String substring(int i, int j)	*i th through (j-1)st characters*
boolean contains(String sub)	*does string contain sub?*
boolean startsWith(String pre)	*does string start with pre?*
boolean endsWith(String post)	*does string end with post?*
int indexOf(String p)	*index of first occurrence of p*
int indexOf(String p, int i)	*index of first occurrence of p after i*
String concat(String t)	*this string with t appended*
int compareTo(String t)	*string comparison*
String replaceAll(String a, String b)	*result of changing a s to b s*
String[] split(String delim)	*strings between occurrences of delim*
boolean equals(Object t)	*is this string's value the same as t's?*

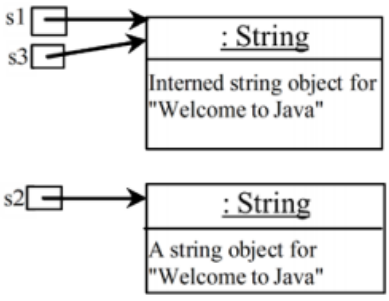
封装的 String

```
String s1 = "Welcome to Java";

String s2 = new String("Welcome to Java");

String s3 = "Welcome to Java";

System.out.println("s1 == s2 is " + (s1 == s2));
System.out.println("s1 == s3 is " + (s1 == s3));
```



- s1 == s2 is false
- s1 == s3 is true
- 如果使用 new 操作符，将创建一个新对象
- 如果使用 String 初始化器，则如果已经创建了内部对象，则不会创建新对象

Color ADT

- 颜色是电磁辐射在眼睛中的一种感觉
- ADT 允许我们编写操作颜色的 Java 程序

Values

		examples							
R (8 bits)	red intensity	255	0	0	0	255	0	119	105
G (8 bits)	green intensity	0	255	0	0	255	64	33	105
B (8 bits)	blue intensity	0	0	255	0	255	128	27	105
color									

API (operations)

Public class java.awt.Color	
Color(int r, int g, int b)	
int getRed()	*red intensity*
int getGreen()	*green intensity*
int getBlue()	*blue intensity*
Color brighter()	*brighter version of this color*
Color darker()	*darker version of this color*
String toString()	*string representation of this color*
boolean equals(Color c)	*is this color the same as c's ?*

单色亮度 monochrome luminance

一种颜色的单色亮度量化了它的有效亮度

	examples							
red intensity	255	0	0	0	255	0	119	105
green intensity	0	255	0	0	255	64	33	105
blue intensity	0	0	255	0	255	128	27	105
color								
luminance	76	150	29	0	255	52	58	105

示例：在屏幕上，字体和背景色以什么样的搭配最易阅读

- 单色亮度的绝对值 > 128 的时候

```
public static boolean compatible(Color a, Color b)
{
  return Math.abs(lum(a) - lum(b)) > 128.0;
}
```

		76	0	255	52
	76	255	76	179	24
	0	76		255	52
	255	179	255		203
	52	24	52	203	

灰度 garyscale

目的：将有颜色的图片转换为灰白的值

- 当所有三个 R、G 和 B 值相同时，生成的颜色灰度从 0 (黑色)到 255 (白色)
- 其实，给予颜色的就是单色亮度

```
public static Color toGray(Color c)
{
   int y = (int) Math.round(lum(c));
   Color gray = new Color(y, y, y);
   return gray;
}
```

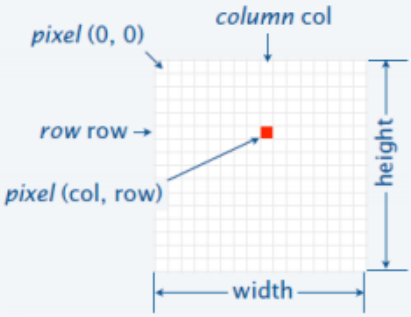
method for Luminance library

	examples							
red intensity	255	0	0	0	255	0	119	105
green intensity	0	255	0	0	255	64	33	105
blue intensity	0	0	255	0	255	128	27	105
color								
luminance	76	150	29	0	255	52	58	105
grayscale								

Picture ADT

- 图片是由像素组成的二维数组
- ADT 允许我们编写处理图片的 Java 程序

Values (2D arrays of Colors)

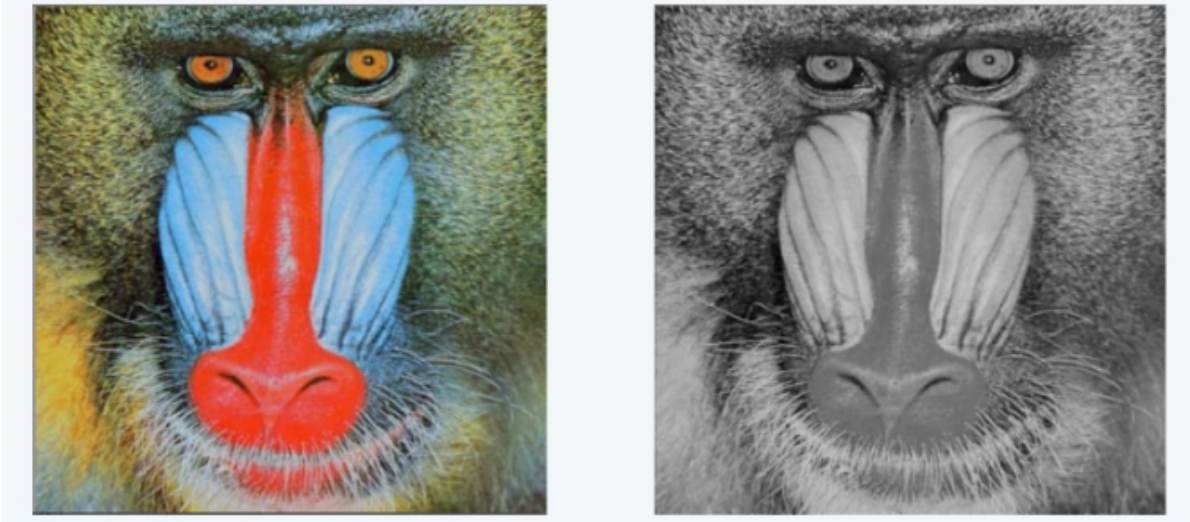


API (operations)

public class Picture	
Picture(String filename)	*create a picture from a file*
Picture(int w, int h)	*create a blank w-by-h picture*
int width()	*width of the picture*
int height()	*height of the picture*
Color get(int col, int row)	*the color of pixel (col, row)*
void set(int col, int row, Color c)	*set the color of pixel (col, row) to c*
void show()	*display the image in a window*
void save(String filename)	*save the picture to a file*

图片灰度转换

目的：将彩色图片转换成灰色图片

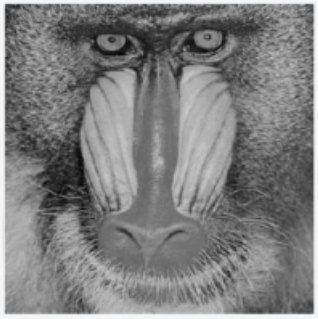


```
import java.awt.Color;
public class Grayscale
{
   public static void main(String[] args)
   {
      Picture pic = new Picture(args[0]);
      for (int col = 0; col < pic.width(); col++)
         for (int row = 0; row < pic.height(); row++)
         {
            Color color = pic.get(col, row);
            Color gray  = Luminance.toGray(color);
            pic.set(col, row, gray);
         }
      pic.show();
   }
}
```

create a new picture

fill in each pixel

```
% java Grayscale mandrill.jpg
```

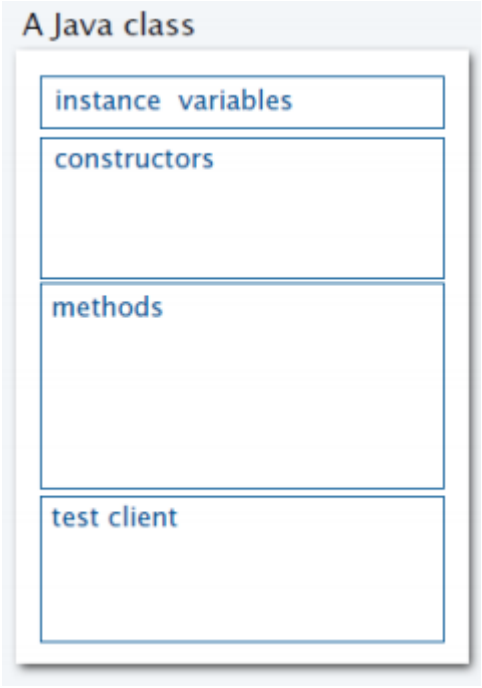


实现一个数据类型

要创建数据类型，您需要提供以下代码

- 定义一系列值（实例变量）
- 实现对于这些值的操作（方法）
- 创造和实例化新的对象（构造器）

在 Java 中，一个数据类型的实现也被称为类 class


A Java class
instance variables
constructors
methods
test client