实验5 类和对象

# 实验目的

掌握类和对象的创建

掌握构造函数、构造函数的重载，拷贝构造函数、析构函数的设计和使用

掌握成员函数的设计和使用

# 实验内容

下面的代码已经创建了图像类的框架，请完善该图像类。在该类中，实现图像的读入、保存、显示，并实现图像的翻转、缩放、裁剪等操作。在主程序中，读入某个图像文件（比如“fruits.jpg”） ，对其进行缩小，上下翻转，左右翻转，指定区域裁剪等操作。

**[使用多文件结构设计该类，即类的声明代码在.h文件中，类的实现代码在.cpp文件中，main函数的代码在另一个.cpp文件中。]**

请编程实现：

图像类：

1. 创建图像类Image，实现各个重载的构造函数，拷贝构造函数（深拷贝），析构函数。
2. 实现对图像的读入，保存，显示。即实现函数Read，Show，Write。
3. 获取图像中某像素点的值。即实现函数At()。
4. 将图像中给定像素点的像素值设置为某值，即实现函数Set。将图像所有像素的像素值设定为某值，即实现函数SetAll。
5. 同一个函数实现图像的上下翻转、左右翻转。即实现函数Flip。
6. 根据指定区域裁剪图像。
7. 求图像的均值，方差。
8. 图像的旋转，缩放。
9. 定义友元函数交换两个Image对象的数据。

在函数中实现：

1. 创建Image类对象img。
2. 读入文件中的图像“fruits.jpg”,并显示。
3. 利用Image类的成员函数，对图像进行翻转、旋转，并显示。
4. 利用Image类的成员函数，将图像长宽缩小到1/2大小，并显示；将图像长宽放大2倍，并显示。
5. 利用拷贝构造函数，创建新的对象new\_img。
6. 给定的两个点（Point）：左上点（top\_left）和右下点（bottom\_right），将此指定区域内的new\_img对象图像进行裁剪操作，并显示结果。
7. 求图像的所有像素点的均值和方差，并输出。
8. 交换两个Image对象的数据。

**Image.h**

#ifndef IMAGE\_H

#define IMAGE\_H

class Image

{

public:

Image(); //无参数的构造函数，创建行列都为零的Image对象

Image(int h, int w); //构造函数重载，创建h行，w列的Image对象

Image(int h, int w, unsigned char val); //构造函数重载，创建的图像像素值都为val;

Image(char\* ImageName); //构造函数重载，利用文件名从硬盘加载图像文件成为Image对象;

Image(unsigned char \*m, int rows, int cols); //构造函数重载，从一维静态数组创建Image对象，图像的行数和列数由后面两个参数给出;

Image(unsigned char m[][100], int rows); //构造函数重载，从静态二维数组创建Image对象，图像的行数（二维数组的第一个维度）由第二个参数rows给出;

Image(unsigned char \*\*m, int h, int w); //构造函数重载，从动态数组（二级指针）创建Image对象，图像的行数和列数由后面两个参数给出;

Image(const Image &im); //拷贝构造函数;

~Image(); //析构函数;

void Read(char\* ImageName); //从硬盘文件中读入图像数据;

void Write(char\* filename); //将图像数据保存为图像文件;

void Show(char\* winname); //显示图像;

unsigned char& At(int row, int col); //获取第row行第col列的像素点的值;

void Set(int row, int col, unsigned char value); //设置像素(row,col)为某值;

void SetAll(unsigned char value); //设置图像所有像素为同一值;

void Flip(int code); //图像的翻转; 根据code的值：0:左右翻转，1:上下翻转;

void Resize(int code); //图像的缩放;根据code的值：0:缩小一倍，1:放大一倍;

void Cut(int x1,int y1,int x2,int y2);//裁剪点(x1,y1)到点(x2,y2)的图像

void Rotate(int degree);//图像旋转的函数（简单起见，旋转角度为90度的整数倍）

void Mean\_Variance(float &m, float &var);//求图像的均值和方差，利用参数输出

friend void Swap(Image &a, Image &b);//使用友元函数交换两个Image对象的数据

private:

unsigned char \*\*data;

int height;

int width;

};

#endif

**Image.cpp**

#include "cv.h"

#include "highgui.h"

#include "Image.h"

*//构造函数*

Image::Image()

{

//write your code here

}

*//构造函数重载*

Image::Image(int h, int w)

{

//write your code here

}

// 其他重载构造函数的实现

// ......

*//拷贝构造函数*

Image::Image(const Image &im)

{

//write your code here

}

*//析构函数*

Image::~Image()

{

//write your code here

}

*//从硬盘读入图像文件;*

void Image::Read(char\* ImageName)

{

IplImage\* img = cvLoadImage(ImageName, CV\_LOAD\_IMAGE\_GRAYSCALE);

unsigned char \*img\_data = (unsigned char \*)(img->imageData);

int widthstep = img->widthStep;

//将一维指针img\_data指向的内存中的值写入成员变量二维指针data所指的内存中

//write your code here

cvReleaseImage(&img);

}

//保存图像;

void Image::Write(char \*filename)

{

IplImage\* img = cvCreateImage(cvSize(width, height), IPL\_DEPTH\_8U, 1);

unsigned char \*img\_data = (unsigned char \*)(img->imageData);

int widthstep = img->widthStep;

//将成员变量二维指针data所指内存中的值写入一维指针img\_data所指的内存

//write your code here

cvSaveImage(filename, img);

cvReleaseImage(&img);

}

*//显示图像;*

void Image::Show(char \*winname)

{

IplImage \*img = cvCreateImage(cvSize(width, height), IPL\_DEPTH\_8U, 1);

unsigned char \*img\_data = (unsigned char \*)(img->imageData);

int widthstep = img->widthStep;

//将data所指内存中的值写入img\_data所指的内存

//write your code here

cvNamedWindow(winname, CV\_WINDOW\_AUTOSIZE);//创建窗口

cvShowImage(winname, img);

cvWaitKey(0);

cvReleaseImage(&img); //释放图像;

}

*//获取图像中指定点的值*

unsigned char& Image::At(int row, int col)

{

//write your code here

}

*//设置图像为同一值*

void Image::Set(unsigned char value)

{

//write your code here

}

*//false 左右，true 上下;*

void Image::Flip(int code)

{

//write your code here

}

*//图像缩小，放大*

void Image::Resize(int code)

{

//write your code here

}

//图像裁剪的函数

//图像旋转的函数

//write your code here

//实现友元函数，交换两个Image对象的数据

void Swap(Image &a, Image &b)

{

}

**CppExp.cpp**

#include “Image.h”

int main(int argc, char\* argv[])

{

Image img; *//创建对象*

img.Read("Fruits.jpg");

*//img.Write("FruitsCopy.jpg");*

cvNamedWindow("Image", CV\_WINDOW\_AUTOSIZE);

img.Show("Image");

cvWaitKey(0); *//等待按键*

//write your code here

//实现图像的左右翻转,如img.Flip(true);并显示

//实现图像的上下翻转,显示

//实现图像的缩放,显示

//获取图像的某点的像素值,并修改

//使用拷贝构造函数创建新的对象

Image new\_img(img);

//截取指定区域内的图像,并显示

//旋转图像并显示（简单起见，旋转角度为90度的整数倍）

//求图像的均值和方差，并在命令行输出

//交换两个图像的数据

Image img1("Baboon.jpg");

Image img2("Lena.jpg");

img1.Show("Image1");

img2.Show("Image2");

cvWaitKey(0); *//等待按键*

Swap（img1, img2）;

img1.Show("Image1");

img2.Show("Image2");

cvWaitKey(0); *//等待按键*

return 0;

}

# 实验要求

完成上述代码，并能显示正确的结果图像。