1. 定义一个职工类**Employee**

**数据成员 (private)**：

职工姓名 (char name[20])、性别 (bool sex)、年龄、工龄、工资

**成员函数 (public)**：

* 设置职工的基本信息：姓名、性别、年龄、工龄、工资
* 修改职工姓名
* 修改工资
* 年薪计算（12×工资+年终奖）
* 工龄增加
* 显示职工信息

**要求**：将类的定义与类成员函数的实现分开。

2. 定义一个灰度图像类**Image**

**数据成员 (private)：**

图像的宽度、图像的高度、图像的像素值 (int value[200][200])

**成员函数 (public)：**

* 图像初始化（图像中每个像素的灰度值为一个随机产生的0~255之间的值）
* 读取某个指定位置的像素值
* 设置某个指定位置的像素值
* 计算整个图像中所有像素值的平均值
* 显示整个图像的像素值。

**要求**：将类的定义与类成员函数的实现分开。

[参考答案]

#include <iostream>

#include <cstring>

using namespace std;

class Employee

{

public:

// 设置职工的基本信息

void set(char \*\_name, bool \_sex, int \_age, int \_wage, double \_salary);

// 修改职工姓名

void changeName(char \*\_name);

// 修改工资

void changeSalary(double \_salary);

// 年薪计算

double computSalary(double award);

// 工龄增加

void update();

// 显示职工信息

void print();

private:

char name[20]; // 职工姓名

bool sex; // 职工性别

int age; // 职工年龄

int wage; // 职工工龄

double salary; // 职工工资

};

// 设置职工的基本信息

void Employee::set(char \*\_name, bool \_sex, int \_age, int \_wage, double \_salary)

{

strcpy(name,\_name);

sex = \_sex;

age = \_age;

wage = \_wage;

salary = \_salary;

}

// 修改职工姓名

void Employee::changeName(char \*\_name)

{

strcpy(name,\_name);

}

// 修改工资

void Employee::changeSalary(double \_salary)

{

salary = \_salary;

}

// 年薪计算

double Employee::computSalary(double award)

{

return 12\*salary+award;

}

// 工龄增加

void Employee::update()

{

wage++;

age++;

}

// 显示职工信息

void Employee::print()

{

cout<<"姓名: "<<name<<endl;

cout<<"性别: "<<(sex?"男":"女")<<endl;

cout<<"年龄: "<<age<<endl;

cout<<"工龄: "<<wage<<endl;

cout<<"工资: "<<salary<<endl;

}

int main()

{

Employee em; // 类对象定义

em.set("张三",true,30,0,3000); // 类成员访问

em.print();

cout<<"年薪: "<<em.computSalary(10000)<<endl;

em.update();

em.changeSalary(4000);

em.print();

cout<<"年薪: "<<em.computSalary(10000)<<endl;

return 0;

}

#include <iostream>

#include <iomanip>

#include <cstdlib>

#include <ctime>

using namespace std;

class Image

{

public:

// 图像初始化

void set(int \_width, int \_height);

// 读取某个位置的像素值

int getPixel(int x, int y);

// 设置某个位置的像素值

void setPixel(int x, int y, int val);

// 计算整个图像中所有像素值的平均值

double computeAverage();

// 显示整个图像的像素值

void print();

private:

int width; // 图像的宽度

int height; // 图像的高度

int value[200][200]; // 图像的像素值

};

// 图像初始化

void Image::set(int \_width, int \_height)

{

srand(time(0));

width = \_width;

height = \_height;

for(int i=0; i<height; ++i)

for(int j=0; j<width; ++j)

value[i][j] = rand()%256;

}

// 读取某个位置的像素值

int Image::getPixel(int x, int y)

{

return value[x][y];

}

// 设置某个位置的像素值

void Image::setPixel(int x, int y, int val)

{

value[x][y] = val;

}

// 计算整个图像中所有像素值的平均值

double Image::computeAverage()

{

double sum = 0.0;

for(int i=0; i<height; ++i)

for(int j=0; j<width; ++j)

sum += value[i][j];

return sum/(width\*height);

}

// 显示整个图像的像素值

void Image::print()

{

for(int i=0; i<height; ++i)

{

for(int j=0; j<width; ++j)

cout<<setw(6)<<value[i][j];

cout<<endl;

}

}

int main()

{

Image image; // 类对象定义

image.set(10, 10); // 类成员访问

image.setPixel(5,5,200);

image.print();

return 0;

}