

面向对象程序设计作业（5）

姓名：王浩宇	学号：X01814031	成绩：
--------	--------------	-----

1. 一个平面点由两个坐标(x,y)确定，请编写一个表示平面点的类 Point。要求：

- 1) 在默认构造函数中初始化类对象为原点 (0, 0);
- 2) 添加一个带参数的构造函数，传入两个坐标值。
- 3) 添加拷贝构造函数。
- 4) 添加赋值操作函数 operator=。
- 5) 为类的两个成员添加 getter 函数和 setter 函数，getter 函数为 const 函数；

2. 平面直线的方程为 $ax + by + c = 0$ ，请设计一个类 Line，表示一条平面直线。要求：

- 1) 在默认构造函数中初始化直线为 x 轴。
- 2) 添加一个带参数的构造函数，传入直线方程的三个系数。
- 3) 添加拷贝构造函数。
- 4) 添加赋值操作函数 operator=。
- 5) 添加一个函数，获取直线的方向 $\theta = \arctan\left(-\frac{b}{a}\right)$ 。
- 6) 添加一个函数，获取该直线的三个参数。
- 7) 添加一个函数，计算点到直线的距离，函数的参数是第一题的 Point 类对象。
- 8) 添加一个函数，判断一个点是否在这条直线上，函数的参数是

第一题的 Point 类对象。提示：由于存在计算误差，当点到直线的距离小于一个很小的阈值时，可以认为点在直线上。

3. 编写一个函数，用于生成一组随机点 $P_i = (x_i, y_i), i = 1 \dots m$ ，这些随机点都是某一条直线 $l: ax + by + c = 0$ 上的点，但是受到很小的噪声污染，即， $\forall i: y_i = -\frac{b}{a}x_i - \frac{c}{a} + \eta_i$ ，其中 η_i 是一个很小的噪声，服从 0 均值高斯分布（使用 `std::normal_distribution` 生成满足高斯分布的随机数来模拟这个噪声，参考：
http://www.cplusplus.com/reference/random/normal_distribution/）。函数的输出为一个 Line 对象，样本点数目，输出为随机点数组。提示：使用 `vector<Point>` 传回生成的随机点对象数组。
4. 编写一个函数 `FitLine`，该函数传入一组随机点，返回一条直线。该直线是采用最小二乘法，用传入的随机点拟合的直线。
5. 编写一个函数 `CalcDistance`，计算一组点到一条直线的平均距离。
6. 在 `main` 函数中，生成一个 `Line` 对象 `L0`，并调用上述函数生成随机点集，然后调用你的 `FitLine` 函数拟合出直线 `L1`。并用 `CalcDistance` 函数计算你所得到的直线 `L1` 与样本点集之间的拟合误差（即平均距离）。在 `main` 函数中，输出真实直线 `L0` 的方程，拟合直线 `L1` 的方程，以及拟合误差。

本作业需要了解的算法知识如下：

假设有 m 个点 $P_i = (x_i, y_i), i = 1 \dots m$ ，那么直线 $l: ax + by + c = 0$ 的参数可以由以下公式计算：

$$\bar{x} = \frac{1}{n} \sum_{i=1}^m x_i, \bar{y} = \frac{1}{n} \sum_{i=1}^m y_i$$
$$A = \sum_{i=1}^m (x_i - \bar{x})^2, B = \sum_{i=1}^m (x_i - \bar{x})(y_i - \bar{y}), \beta = \frac{1}{\sqrt{A^2 + B^2}}$$
$$a = -\frac{B}{\beta}, b = \frac{A}{\beta}, c = -a \times \bar{x} - b \times \bar{y}$$

要求提交以下内容：

- 1) 你设计的多文件结构中，每个文件的文件名与该文件负责的主要逻辑功能。
- 2) 把你所设计的多个文件依次插入在本文档后面的表格中，每一个源文件都要指明文件名。
- 3) 把每一个测试用例的运行结果截图粘贴在本文档后面。
- 4) 列出你在开发本程序过程中碰到的一个 bug，把包含 bug 的函数代码插入到本文档，并指明 bug 的具体位置，并给出如何修正该 bug。

在下面说明多文档结构：

PointAndLine.h point 类和 line 类和两者的成员方法

DataAndCalculate.h 数据的生成和拟合误差的计算

Main.cpp 主函数，用于输入参数，输出结果

在下面插入代码，列出每个源文件的文件名，在文件名之后换行插入代码：

PointAndLine.h

```
#pragma once

#include <iostream>
#include <algorithm>
#include <unordered_map>
#include <iomanip>
#include <math.h>
#include <vector>
#include <random>
using namespace std;
#define pi acos(-1)

class point
{
private:
    double x;
    double y;
public:
    point():x(0),y(0){}
    point(const double _x,const double _y)
    {
        x=_x;
        y=_y;
    }
    point(const point &v)
    {
        x=v.x;
        y=v.y;
    }
    point &operator=(const point &v)
```

```

{
    if(this!=&v)
        this->x=v. x;
    this->y=v. y;
    return *this;
}
public:
    double getter(int v) const
    {
        if(v==1) return x;
        else return y;
    }
    point setter(const point v)
    {
        this->x=v. x;
        this->y=v. y;
        return *this;
    }
};

class Line
{
private:
    double a;
    double b;
    double c;
public:
    Line():a(0),b(1),c(0){}
    Line(const double _a,const double _b,const double _c)
    {
        a=_a;
        b=_b;
        c=_c;
    }
    Line(const Line &v)
    {
        a=v. a;
        b=v. b;
        c=v. c;
    }
    Line &operator=(const Line &v)
    {
        this->a=v. a;
        this->b=v. b;
    }
}

```

```

        this->c=v. c;
        return *this;
    }

public:
    double GetLineTheta()
    {
        if(a==0&&b>0) return pi/2;
        if(a==0&&b<0) return pi/2;
        return atan(-b/a);
    }

    double GetParameter(const int v)
    {
        if(v==1) return a;
        else if(v==2) return b;
        else return c;
    }

    double CalcDistance(point v)
    {
        return fabs(a*v. getter(2)+b*v. getter(1)+c)/sqrt(a*a+b*b);
    }

    bool IsOnLine(point v)
    {
        if(a*v. getter(1)+b*v. getter(2)+c<1e-4)
            return true;
        else
            return false;
    }
};

vector<point> CreateData(Line v, int cnt);

Line FitLine(vector<point>data);

double CalcDistance(Line v, vector<point>data);

```

DataAndCalculate.h

#include "PointAndLine.h"

```

vector<point> CreateData(Line v, int cnt)
{
    vector<point>data;
    default_random_engine generator;
    normal_distribution<double>dist;
    for(int i=1;i<=cnt;i++)
    {
        double x0=rand()%150;
        double a=v.GetParameter(1);
        double b=v.GetParameter(2);
        double c=v.GetParameter(3);
        double y0=-b*x0/a-c/a+dist(generator);
        point temp(x0,y0);
        data.push_back(temp);
    }
    return data;
}

Line FitLine(vector<point>data)
{
    double sigma_xy=0,sigma_x=0,sigma_y=0,sigma_x2=0,average_x=0,average_y=0;
    for(int i=0;i<(int)data.size();i++)
    {
        sigma_xy+=data[i].getter(1)*data[i].getter(2);
        sigma_x+=data[i].getter(1);
        sigma_y+=data[i].getter(2);
        sigma_x2+=data[i].getter(1)*data[i].getter(1);
    }
    int N=data.size();
    average_x=sigma_x/N;
    average_y=sigma_y/N;
    double a=(N*sigma_xy-sigma_x*sigma_y)/(N*sigma_x2-sigma_x*sigma_x);
    double b=average_y-a*average_x;
    Line ans(1,-a,-b);
    return ans;
}

double CalcDistance(Line v, vector<point>data)
{
    double dis=0;
    for(int i=0;i<(int)data.size();i++)
    {
        dis+=v.CalcDistance(data[i]);
    }
}

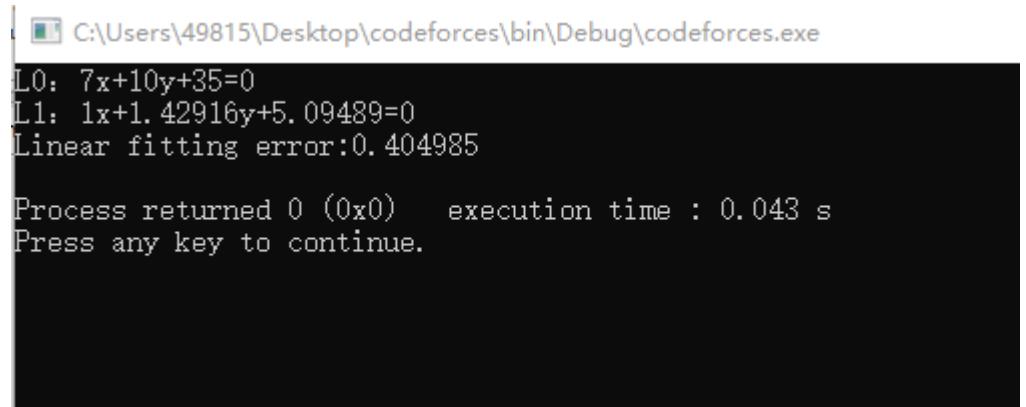
```

```
    dis=dis/(int)data.size();
    return dis;
}
```

Main.cpp

```
#include "PointAndLine.h"
#include "DataAndCalculate.h"
int main()
{
    Line L0(10, 35, 45);
    vector<point>data = CreateData(L0, 25);
    Line L1 = FitLine(data);
    double dis = CalcDistance(L1, data);
    cout << "L0:" << L0.GetParameter(1) << "x+" << L0.GetParameter(2) << "y+"
    << L0.GetParameter(3) << "=0" << endl;
    cout << "L1:" << L1.GetParameter(1) << "x+" << L1.GetParameter(2) << "y+"
    << L1.GetParameter(3) << "=0" << endl;
    cout << "Linear fitting error:" << dis << endl;
    return 0;
}
```

在下面粘贴程序运行截图：



```
C:\Users\49815\Desktop\codeforces\bin\Debug\codeforces.exe
L0: 7x+10y+35=0
L1: 1x+1.42916y+5.09489=0
Linear fitting error:0.404985

Process returned 0 (0x0)  execution time : 0.043 s
Press any key to continue.
```

```
C:\Users\49815\Desktop\codeforces\bin\Debug\codeforces.exe
L0: 3x+6y+35=0
L1: 1x+2.00059y+11.7616=0
Linear fitting error:0.31584
```

```
Process returned 0 (0x0) execution time : 0.038 s
Press any key to continue.
```

```
C:\Users\49815\Desktop\codeforces\bin\Debug\codeforces.exe
L0: 7x+8y+35=0
L1: 1x+1.14345y+5.09489=0
Linear fitting error:0.465036
```

```
Process returned 0 (0x0) execution time : 0.059 s
Press any key to continue.
```

在下面列出你遇到的 bug 以及对 bug 的修正：

没有遇到 bug，但是遇到了一个易错点，在计算直线的 θ 角度时，由于传入的参数是 $-b/a$ ，这里要对 a 进行特判一下，不然可能会出现 b 除以 0 的错误。