

南京航空航天大学

《面向对象程序设计语言》课程设计报告

空间数据分析系统

学号：

姓名： _

日期：

```

class Shape{

    int obj_id; //对象标识符

    string des; //描述对象特征

public:
    virtual float Area() = 0;    //纯虚函数
    virtual void WriteToFile(ofstream& o) = 0; //纯虚函数

};

class Point: public Shape{ //派生类
    float x, y;
    static int count;
public:
    virtual float Area(){return 0;}
    virtual void WriteToFile(ofstream& o);    //需要实现
};

class Rect: public Shape{ //派生类
    Point lp; //左下角
    Point rp; //右上角
    static int count;
public:
    virtual float Area(); //需要实现
    virtual void WriteToFile(ofstream& o);    //需要实现
};

class Line: public Shape{ //派生类
    Point p1;
    Point p2;
    double len;
    static int count;
public:
    virtual float Area(); //需要实现，求解长度
    virtual void WriteToFile(ofstream& o);    //需要实现
};
    
```

任务 1: 实现类 Point、类 Rectangle、类 Line 的成员函数，包括构造函数、拷贝构造函数、赋值函数、Area()函数、operator<<重载函数（采用 cout 能输出 Point、Rect、Line）、静态成员函数 static int GetCount(){return count;}

要求: 采用多文件技术 Point.h, Rectangle.h, Line.h, main.cpp --- 10 分

任务 2:

①从三个文件分别读入所有数据（Point 对象, Rect 对象, Line 对象），保存在 vector 中(利用文件流)，通过调用 GetCount()输出 Point、Rect、Line 对象个数

1. point.txt 2. rect.txt 3. line.txt ---10 分

或者②建立模板类 template<class T> ---20 分

```
class List{
    T* l;
    int size;
public:
    //实现成员函数
};
```

③任意给一个 Point 对象，找出所有和该对象成员变量 des(小写形式)相同的 Rect 对象(即 Point 的变量 des 和 Rect 的变量 des 相同)，输出所有的 Rect 对象到屏幕；任意给一个 Rect 对象，找出所有和该对象的 des 相同的 Point 对象，输出所有的 Point 对象到屏幕（检测 cout<<功能） ----10 分

任务 3: ① 遍历存储 Rect 的结构(vector 或 List)，找出面积最小和最大的 Rect。将最小和最大的 Rect 写入到文件 Rect_data.txt。

要求: 如果使用 vector，必须使用迭代器(iterator) ----5 分

② 遍历存储 Line 的结构(vector 或 List)，找出长度最小和最大的 Line。(类 Line 用 Area()函数返回长度)。将最小和最大的 Rect 写入到文件 Line_data.txt。

要求: 如果使用 vector，必须使用迭代器(iterator) ----5 分

任务 4: ①对类 Point 实现运算符重载 operator<，并演示

p1 < p2: p1.x < p2.x or p1.x = p2.x and p1.y < p2.y ----5 分

②对类 Point、类 Rect 实现运算符重载 operator+

Point: x1+x2, y1+y2

Rect: lp = min(lp1, lp2), rp = max(rp1, rp2)

任意给出两个 Point 和 Rect 对象，演示 operator+的功能 ----5 分

任务 5: ① 类 Point 实现成员函数 Distance()，求解两个点之间的距离 ---2 分

② 类 Rect 实现成员函数 Distance()，求解距离(最近的两点之间的距离) ---2 分

- ③ 类 Line 实现成员函数 Distance(), 求解距离(最近的两点之间的距离)
---2 分
- ④ 类 Point 实现成员函数 Distance_R(), 求解 Point 对象和 Rect 对象之间的距离(最近的两点之间的距离); 类 Point 实现成员函数 Distance_L(), 求解 Point 对象和 Line 对象之间的距离(最近的两点之间的距离)
---4 分
- ⑤ 编写程序演示上述功能

任务 6: ①体现多态性, 分别在 Point、Rect 和 Line 中实现 WriteToFile()虚函数将数据写入到文件, 格式为

Point: ID (x, y) des

Rect: ID (x1, y1) (x2, y2) des

Line: ID (x1, y1) (x2, y2) len

要求: 在 main 函数中调用 WriteData(Shape* s, ofstream& out_file)

void WriteData(Shape* s, ofstream& out_file)

{

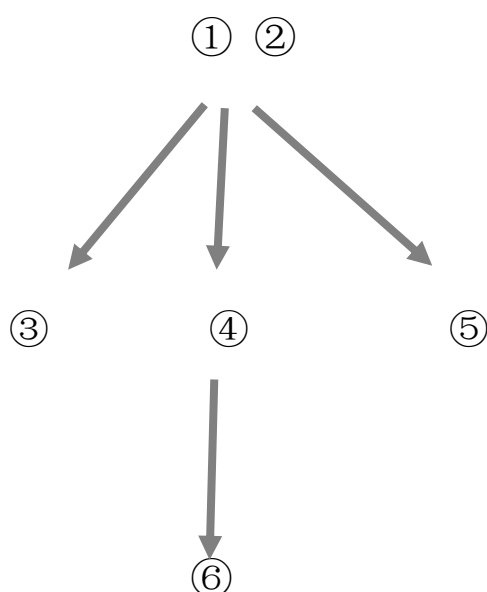
}

1.将所有 Point 对象的字符串变量 des 变成小写, 按坐标排序后(任务 4 中的①, 利用 sort 排序, $p_1 < p_2 < p_3 \dots < p_n$) 写入到 Point2.txt; --10 分

2.将 Rect 对象按照 des(字符串排序), 写入到 Rect2.txt; --10 分

3.将 Line 对象按照 len 递增数序排序, 写入到文件 Line2.txt --10 分

提示: 重载运算符 operator<, Rect 按照字符串排序, Line 按照 len 排序, 运用 sort 排序



文件格式:

① Point.txt

ID x y Des

ID: int

x, y: float

Des: string

②Line.txt

ID x1 y1 x2 y2 Len

ID: int

x1,y1,x2,y2: float

len: float

③Rect.txt

ID x1 y1 x2 y2 Area

ID:int

x1,y1,x2,y2: float

Area:float

如果选择任务 3，成绩 = 实际分数×110%