

《数据结构与算法实验》第 1 次实验

学院：

专业：

年级：

姓名：

学号：

日期： 2022 年 2 月 28 日

一、 实验目的

1. 复习模板、运算符重载、异常处理、多文件的编写和调试。
2. 以 Point 点类为例，复习有关的 C++ 编程知识。
3. 复习继承与派生、多态性与虚函数，先看《C++ 程序设计》（谭浩强）第 11 章，第 12 章。
4. 完成下面 4 个题目。

二、 实验内容

1. **类的定义与测试：**定义一个点类，并进行测试。
2. **类的继承：**分别声明 Teacher(教师)类和 Cadre(干部)类，采用多重继承方式由这两个类派生出新类 Teacher_Cadre(教师兼干部)类。要求：
 - 在两个基类中都包含姓名、年龄、性别、地址、电话等数据成员。
 - 在 Teacher 类中还包含数据成员 title(职称)，在 Cadre 类中还包含数据成员 post(职务)。在 Teacher_Cadre 类中还包含数据成员 wages(工资)。
 - 对两个基类中的姓名、年龄、性别、地址、电话等数据成员用相同的名字，在引用这些数据成员时，指定作用域。
 - 在类体中声明成员函数，在类外定义成员函数。
 - 在派生类 Teacher_Cadre 的成员函数 show 中调用 Teacher 类中的 display 函数，输出姓名、年龄、性别、职称、地址、电话，然后再用 cout 语句输出职务与工资。
3. **类的派生：**定义一个抽象类 Shape，由它派上生出 3 个派生类：Circle(圆形)、Rectangle(矩形)、Triangle(三角形)，用一个函数 printArea 分别输出以上三者的面积，3 个图形的数据在定义对象时给定。
4. **类的派生：**定义一个抽象类 Shape，由它派上生出 5 个派生类：Circle(圆形)、Square(正方形)、Rectangle(矩形)、Trapezoid(梯形)、Triangle(三角形)。用虚函数分别计算几种图形的面积，并求它们的和。要求用基类指针数组，使它的每一个元素指向一个派生类对象。

三、 设计与编码

1. 本实验用到的理论知识

- **运算符重载（静态多态性）**：对于用户自己定义的类型的数据，用重载实现它们的运算和输入输出。
- **动态多态性**：通过虚函数实现在程序运行时动态决定操作针对的对象，具体决定调用哪个函数。
- **继承和派生**：通过继承和派生在已有类的基础上建立新的类，具有原来类的一定特点，是原来类的具体化表现，用来解决更加具体的问题。

2. 算法设计

- (1) **类的定义与测试**：一个点类应该包含该点的横坐标 x 与纵坐标 y ，还由一些常见的函数，如点的加法，因此需要对简单四则运算的符号进行重载，从而实现点类之间的运算。至于点的比较则是通过比较两点到原点的距离决定，只有两点横坐标纵坐标相等时两点才相等。最后还要对输入、输出运算符进行重载，实现点的输入和输出。
- (2) **类的继承**：通过类的继承对基类中的信息进行提取。
- (3) **类的派生**：建立一个 Shape 基类，在它的基础之上派生出不同形状类，不同形状类里编码该形状的面积公式，在程序运行时进行不同派生类的调用即可。计算面积之和则是使用基类指针数组，建立新的变量，使用循环语句求和。

3. 编码

(1) 类的继承与测试-point.h（定义类）

```
1 //Point.h 点类的定义
2 #ifndef _POINT_H
3 #define _POINT_H
4 #include <iostream>
5 using namespace std;
6 class Point { //类的定义
7 public: //外部接口
8     Point() //无参构造函数
9     {
10         x=0;y=0;
11     }
12     Point(int X, int Y) //有参构造函数
13     {
14         x=X,y=Y;
15     }
16     ~Point() //析构造函数
17     {
18         // cout<<"删除"<<"("<<x<<","<<y<<")点."<<endl;
19     }
20     Point(const Point &p) //复制构造函数;
21     {
22         x=p.x;y=p.y;
23     }
24     void move(int newX,int newY); //点值变为新的值
25     int getX(); //获取x值
26     int getY(); //获取y值
```

```

27 Point operator = (const Point & p); //重载赋值运算符
28 Point operator + (const Point& p); //重载+运算符:当前对象+参数中对象p
29 Point operator * (const int i); //重载*运算符:当前对象*参数表中整数i
30 int operator >(const Point& p); //重载>运算符:判断当前对象>参数中对象p
31 int operator <(const Point& p); //重载<运算符:判断当前对象<参数中对象p否
32 int operator ==(const Point& p); //重载==运算符:判断当前对象=参数中对象p否
33 friend istream &operator >> (istream &input, Point &p); //运算符<<重载
34 friend ostream &operator << (ostream &output, Point &p); //运算符<<重载
35
36 private: //私有数据成员
37     int x, y;
38 };
39 #endif // _POINT_H

```

(1) 类的继承与测试-point.cpp (定义类函数)

```

1  #include<iostream>
2  #include"point.h"
3  using namespace std;
4  void Point::move(int newX,int newY)
5  {
6      x=newX;
7      y=newY;
8  }
9  int Point::getX()
10 {
11     return x;
12 }
13 int Point::getY()
14 {
15     return y;
16 }
17
18 Point Point::operator = (const Point& p)
19 {
20     x=p.x;
21     y=p.y;
22     return *this;;
23 }
24
25 Point Point::operator+(const Point& p)
26 {
27     Point tp;
28     tp.x=x+p.x;
29     tp.y=y+p.y;
30     return tp;
31 }
32 Point Point::operator*(const int i)
33 {
34     Point tp;

```

```

35     tp.x=x*i;
36     tp.y=y*i;
37     return tp;
38 }
39 int Point::operator >(const Point& p)//比较它们距离原点的距离
40 {
41     return (x*x+y*y)>(p.x*p.x+p.y*p.y)? 1:0 ;
42 }
43 int Point::operator <(const Point& p)//比较它们距离原点的距离
44 {
45     return (x*x+y*y)<(p.x*p.x+p.y*p.y)? 1:0 ;
46 }
47 int Point::operator ==(const Point& p)
48 {
49     return (x==p.x)&(y==p.y)? 1:0 ;
50 }
51 ostream & operator << (ostream &output, Point &p)
52 {
53     output<<"("<<p.x<<","<<p.y<<")";
54     return output;
55 }
56 istream & operator >> (istream&input, Point &p)
57 {
58     input>>p.x>>p.y;
59     return input;
60 }

```

(1)类的继承与测试-main.cpp（主函数）

```

1  //Point类的测试
2
3  #include "point.h"
4  #include <iostream>
5  using namespace std;
6
7  int main()
8  { Point p,p1,p2,p3;
9      cin>>p; //输入0 0 进行测试
10     cout<<"p="<<p<<'\n';
11     p.move(1,2); //1 2
12     cout<<"new p="<<p<<'\n';
13     p1.move(3,4);
14     p1=p;
15     cout<<"p="<<p<<","<<p1<<'\n';
16     p2=p1*3; //3 6
17     cout<<p2<<'\n';
18     cout<<"p1="<<p1<<" p2=p1*3="<<p2;
19     p3=p1+p2;

```

```
20     cout<<"p1="<<p1<<" "<<"p2="<<p2<<" p3=p1+p2="<<p3<<"\n";
21     if (p3>p2) cout<<"p3="<<p3<<" p2="<<p2<<" p3>p2"<<"\n";
22     if (p1<p2) cout<<"p1="<<p1<<" p2="<<p2<<" p1<p2"<<"\n";
23     if (p1==p2) cout<<"p1==p2"<<"\n";
24
25     return 0;
26 }
```

(2) 类的继承

```
1  #include <iostream>
2  #include <string>
3  using namespace std;
4  class teacher
5  {
6      public:
7          teacher(string nam,int a,char s,string add,string tel,string til)
8          {
9              name=nam;
10             age=a;
11             sex=s;
12             address=add;
13             telephone=tel;
14             title=til;
15         }
16         void display();
17     protected:
18         string name;
19         int age;
20         char sex;
21         string address;
22         string telephone;
23         string title;
24 };
25 void teacher::display()
26 {
27     cout<<"name: "<<name<<endl;
28     cout<<"sex:"<<sex<<endl;
29     cout<<"address:"<<address<<endl;
30     cout<<"telephone:"<<telephone<<endl;
31     cout<<"title:"<<title<<endl;
32 }
33
34 class cadre
35 {
36     public:
37     cadre(string nam1,int a1,char s1,string add1,string tel1,string pos)
38     {
39         name=nam1;
40     }
```



```

41         age=a1;
42         sex=s1;
43         address=add1;
44         telephone=tel1;
45         post=pos;
46     }
47     void display1();
48     cadre(string pos)
49     {
50         post=pos;
51     }
52
53
54     protected:
55         string post;
56         string name;
57         int age;
58         char sex;
59         string address;
60         string telephone;
61         int wages;
62
63
64 };
65 void cadre::display1()
66 {
67     cout<<"name: "<<name<<endl;
68     cout<<"age:"<<age<<endl;
69     cout<<"sex:"<<sex<<endl;
70     cout<<"address:"<<address<<endl;
71     cout<<"telephone:"<<telephone<<endl;
72
73     cout<<"post:"<<endl;
74 }
75
76 class teacher_cadre:public teacher, public cadre
77 {
78     public:
79     teacher_cadre
80     (string nam,int a,char s,string add,
81     string tel,string til,string pos,int wag):
82     teacher(nam,a,s,add,tel,til),
83     cadre(pos),
84     wages(wag)
85     { }
86
87     void show();
88     protected:
89         int wages;
90
91
92 };

```

```
93 void teacher_cadre::show()
94 {
95     display();
96     cout<<"post:"<<post<<endl;
97     cout<<"wages:"<<wages<<endl;
98 }
99
100 int main()
101 {
102     teacher_cadre
103     a1("ZHANGli",21,'m',"shanghai","1234567","teacher","headmaster",9999);
104     a1.show();
105     return 0;
106 }
```

(3) 类的派生

```
1  #include <iostream>
2  using namespace std;
3
4  class Shape
5  {public:
6      virtual double area() const=0;
7  };
8
9  //定义Circle类
10 class Circle:public Shape
11 {public:
12     Circle(double r):radius(r){}
13     virtual double area()const {return 3.14*radius*radius;}
14     protected:
15     double radius;
16 };
17
18 //定义Rectangle类
19 class Rectangle:public Shape
20 {public:
21     Rectangle(double a,double b):aside(a),bside(b){}
22     virtual double area()const {return aside*bside;}
23     protected:
24     double aside,bside;
25 };
26
27 //定义Triangle类
28 class Triangle:public Shape
29 {public:
```

```
30     Triangle(double w,double h):width(w),height(h){}
31     virtual double area()const {return 0.5*width*height;}
32     protected:
33     double width,height;
34 };
35
36 void printArea(const Shape&s)
37     {cout<<s.area()<<endl;}
38 int main()
39 {
40     Circle circle(2);
41     cout<<"area of circle    = ";
42     printArea(circle);
43
44     Rectangle rectangle(3,4);
45     cout<<"area of rectangle = ";
46     printArea(rectangle);
47
48     Triangle triangle(3,4);
49     cout<<"area of triangle  = ";
50     printArea(triangle);
51
52     return 0;
53 }
```

(4) 类的派生

```
1  #include <iostream>
2  using namespace std;
3  class Shape
4  {public:
5      virtual double area()const=0;
6  };
7
8  //定义Circle类
9  class Circle:public Shape
10 {public:
11     Circle(double r):radius(r){}
12     virtual double area()const {return 3.14*radius*radius;}
13     protected:
14     double radius;
15 };
16
```



```
17 //定义Square类
18 class Square:public Shape
19 {public:
20     Square(double a):side(a){}
21     virtual double area()const {return side*side;}
22     protected:
23     double side;
24 };
25
26 //定义Rectangle类
27 class Rectangle:public Shape
28 {public:
29     Rectangle(double w,double h):width(w),height(h){}
30     virtual double area()const {return width*height;}
31     protected:
32     double width,height;
33 };
34
35 //定义Trapezoid类
36 class Trapezoid:public Shape
37 {public:
38     Trapezoid(double a,double b,double h):aside(a),bside(b),height(h){}
39     virtual double area()const {return 0.5*height*(aside+bside);}
40     protected:
41     double aside,bside,height;
42 };
43
44 //定义Triangle类
45 class Triangle:public Shape
46 {public:
47     Triangle(double w,double h):width(w),height(h){}
48     virtual double area()const {return 0.5*width*height;}
49     protected:
50     double width,height;
51 };
52
53 int main()
54 {
55     Circle circle(2);
56     Square square(4);
57     Rectangle rectangle(3,4);
58     Trapezoid trapezoid(2,4,3);
59     Triangle triangle(3,4);
60     Shape *pt[5]={&circle,&square,&rectangle,&trapezoid,&triangle};
61     double areas=0.0;
62     for(int i=0;i<5;i++)
63     {
64         areas=areas+pt[i]->area();
```

```
65 | }  
66 | cout<<"sum of all shapes = "<<areas<<endl;  
67 | return 0;  
68 | }
```

四、 运行与测试

(1) 类的继承与测试

```
3 4  
p=(3, 4)  
new p=(1, 2)  
p=(1, 2), p1=(1, 2)  
(3, 6)  
p1=(1, 2) p2=p1*3=(3, 6) p1=(1, 2) p2=(3, 6) p3=p1+p2=(4, 8)  
p3=(4, 8) p2=(3, 6) p3>p2  
p1=(1, 2) p2=(3, 6) p1<p2  
p1==p1  
  
-----  
Process exited after 3.975 seconds with return value 0  
请按任意键继续. . .
```

(2) 类的继承

```
name: ZHANGli  
age:21  
sex:m  
address:shanghai  
telephone:1234567  
title:teacher  
post:headmaster  
wages:9999  
  
-----  
Process exited after 0.153 seconds with return value 0  
请按任意键继续. . .
```

(3) 类的派生

```
area of circle    = 12.56  
area of rectangle = 12  
area of triangle  = 6  
  
-----  
Process exited after 0.1555 seconds with return value 0  
请按任意键继续. . .
```

(4) 类的派生

```
sum of all shapes = 55.56
-----
Process exited after 0.1817 seconds with return value 0
请按任意键继续. . .
```

五、 总结与心得

通过这次实验，对大一上学期的基本 C++ 知识进行了全面详细的复习。在编程实验的过程中我不仅锻炼了编程能力，更学会了许多新的知识。其中，最重要的部分是如何新建项目——将类定义、类函数、主函数分开写，使得更加合理与规范。除此之外还对类的继承和派生进行了复习，以及对于指针的应用，为今后的学习提供基础。