

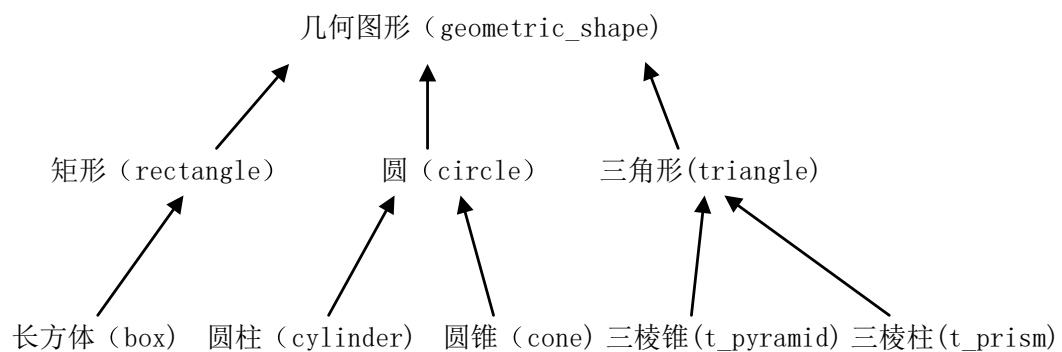
实验3 多态性实现

【实验目的】

- (1) 掌握多态的概念。
- (2) 理解静态多态性和动态多态性的含义。
- (3) 掌握使用虚函数和继承实现动态多态性的方法。
- (4) 掌握运算符重载的方法。

【实验内容】

1. 设有几何图形的派生关系如下图所示。



对平面图形可求周长和面积，对立体图形可以求体积以及底面图形的周长和底面积。

设有主函数如下：

```
int main() {  
    Geometric_shape * gs[]={new Circle(10),new Rectangle(6,8), new  
    Triangle(3,4,5), new Box(6,8,3), new Cylinder(10,3), new Cone(10,3), new  
    T_pyramid(3,4,5,3), new T_prism(3,4,5,3)};  
    for (int i=0;i<8;i++){  
        gs[i]->Show();    // 输出图形的类型，如“圆形”  
        cout<<endl;  
    }  
    cout<<"平面图形: "<<endl;
```

```

        for (int i=0;i<3;i++) {
            cout<<"图形周长: "<<gs[i]->perimeter()<<"\t";
            cout<<"图形面积: "<<gs[i]->area()<<"\t";
            cout<<"图形体积: "<<gs[i]->volume()<<endl;
        }
        cout<<"立体图形: "<<endl;
        for (int i=3;i<8;i++) {
            cout<<"图形底周长: "<<gs[i]->perimeter()<<"\t";
            cout<<"图形底面积: "<<gs[i]->area()<<"\t";
            cout<<"图形体积   : "<<gs[i]->volume()<<endl;
        }
        return 0;
    }
}

```

请编写各类的定义和实现代码，使给定的主函数 main 可以正确运行。

2. 定义一个复数类 **Complex**，重载运算符“+”，使之能执行下列运算。

Complex a(2,5), b(7,8), c(0,0);

c = a + b;

c = 4.1 + a;

3. 定义一个有理数类 **Rational** 并重载输出运算符<<，使得下面程序的输出结果

是： $1/3 + 1/6 = 1/2$

部分程序如下：

```

int main() {
    Rational x(1, 3), y(1, 6), z;
    z = x + y;
    cout << x << " + " << y << " = " << z << endl;
    return 0;
}

```

具体要求如下：

（1）在 `Rational` 类中定义私有数据成员 `nr` 和 `dm` 分别存放分子和分母。同时分子和分母要以最简形式存放。例如：分数 $3/9$ 应该以 $1/3$ 形式存放。

（2）在 `Rational` 类中定义带默认参数的构造函数，默认有理数为 0（分子为 0，分母为 1）。

（3）重载加法运算符 “+”，完成两个有理数的加法运算，运算结果仍然以最简形式存放。

（4）重载输出运算符 “<<”，以分数形式输出有理数。

【实验指导】

1. 用虚函数来实现题（1）主程序中的动态联编。即在 `Geometric_shape` 类中分别将计算面积、周长、体积等函数声明为虚函数后，就可以在该类的（直接或间接）派生类中定义与其基类虚函数原型相同的函数。这时，当用基类指针指向这些派生类的对象时，系统会自动用派生类中的同名函数来代替基类中的虚函数，从而实现运行时的多态。

2. 题（2）重载 “+” 运算符时要求分别用成员函数和友元函数两种方式实现。用友元函数实现时，注意友元函数的位置，测试友元函数的位置在类体内和在类体外是否有不同。

3. 题（3）重载 “+” 运算符时，可以将其重载为类的成员函数；重载 “<<” 运算符的重载时，不能将其重载为类的成员函数，只能重载为类的友元函数。