CSDN 学院

C++语言基础

迂者 - 贺利坚

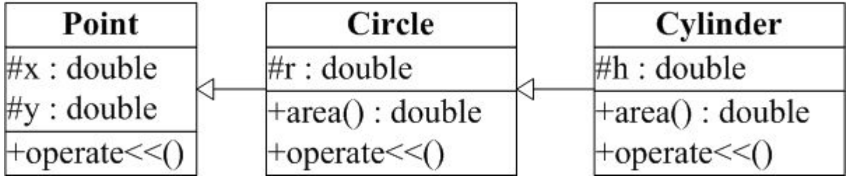
http://blog.csdn.net/sxhelijian/

http://edu.csdn.net

本节主题：

案例：一个接口，多种方法

多态性：一个接口，多种方法



```
class Point{
public:
   Point(double x=0,double y=0);
   friend ostream & operator<<(ostream &,const Point &);
protected:
   double x,y;
};
```

```
class Circle:public Point{
public:
   Circle(double x=0,double y=0,double r=0);
   double area ( ) const;
   friend ostream &operator<<(ostream &,const Circle &);
protected:
   double radius;
};
```

```
class Cylinder:public Circle {
public:
   Cylinder (double x=0,double y=0,double r=0,double h=0);
   double area( ) const;
   friend ostream& operator<<(ostream&,const Cylinder&);
protected:
   double height;
};
```

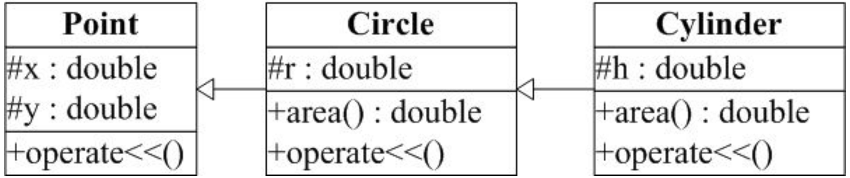
```
Point::Point(double a,double b):x(a), y(b){}
ostream & operator<<(ostream &output,const Point &p){
   output<<"  点：["<<p.x<<","<<p.y<<"]"<<endl;
   return output;
}
```

```
Circle::Circle(double a,double b,double r):Point(a,b),radius(r) { }
double Circle::area( ) const { return 3.14159*radius*radius; }
ostream &operator<<(ostream &output,const Circle &c)
{
   output<<"  圆：Center=["<<c.x<<","<<c.y<<"],r="<<c.radius<<
       ",area="<<c.area( )<<endl;
   return output;
}
```

```
Cylinder::Cylinder(double a,double b,double r,double h):Circle(a,b,r),height(h) {}
double Cylinder::area( ) const
     { return 2*Circle::area( )+2*3.14159*Circle::radius*height;}
ostream &operator<<(ostream &output,const Cylinder& cy){
   output<<"  圆柱：Center=["<<cy.x<<","<<cy.y<<"],r="<<cy.radius<<
        ", h="<<cy.height<<",area="<<cy.area( )<<endl;
   return output;
}
```

用指针输出几何体

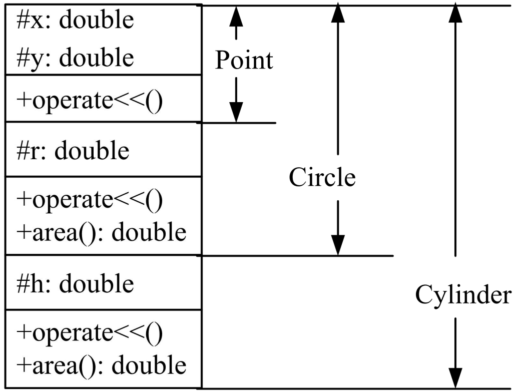
```
int main( )
{
   Cylinder cy1(3.5, 6.4, 5.2, 10);
   cout<<cy1<<endl;

   Point *p=&cy1;
   cout<<"用指向点的指针输出："<<endl;
   cout<<(*p);

   Circle *c= &cy1;
   cout<<"用指向圆的指针输出："<<endl;
   cout<<(*c);

   Cylinder *cy= &cy1;
   cout<<"用指向圆柱的指针输出："<<endl;
   cout<<(*cy)<<endl;
   return 0;
}
```



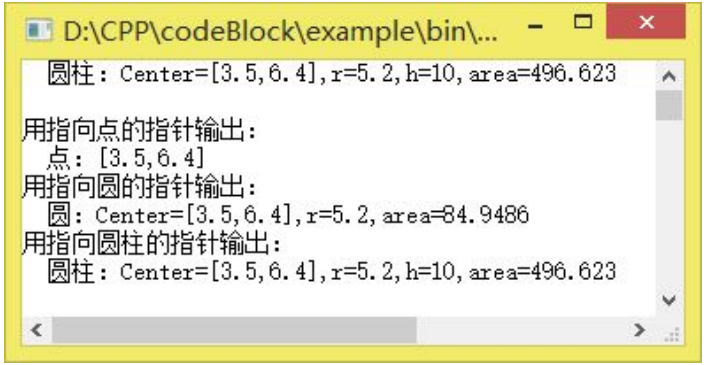
Point
#x : double
#y : double
+operate<<()

Circle
#r : double
+area() : double
+operate<<()

Cylinder
#h : double
+area() : double
+operate<<()



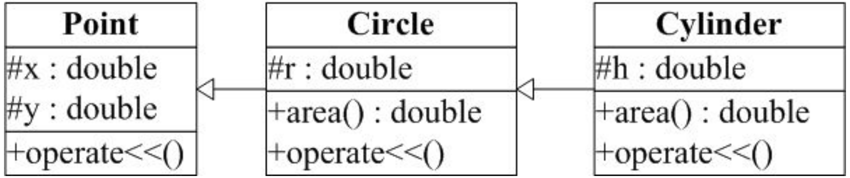
#x: double
#y: double
+operate<<()
#r: double
+operate<<()
+area(): double
#h: double
+operate<<()
+area(): double



```
 圆柱： Center=[3.5,6.4],r=5.2,h=10,area=496.623

用指向点的指针输出:
 点： [3.5,6.4]
用指向圆的指针输出:
 圆： Center=[3.5,6.4],r=5.2,area=84.9486
用指向圆柱的指针输出:
 圆柱： Center=[3.5,6.4],r=5.2,h=10,area=496.623
```

用引用输出几何体

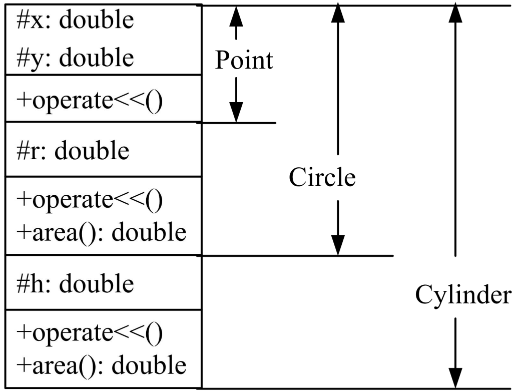
```
int main( )
{
   Cylinder cy1(3.5, 6.4, 5.2, 10);
   cout<<cy1<<endl;

   cout<<"使用引用："<<endl;
   Point &pRef=cy1;
   cout<<"用点的引用输出："<<endl;
   cout<<pRef<<endl;

   Circle &cRef=cy1;
   cout<<"用圆的引用输出："<<endl;
   cout<<cRef<<endl;

   Cylinder &cyRef=cy1;
   cout<<"用圆柱的引用输出："<<endl;
   cout<<cyRef<<endl;
   return 0;
}
```



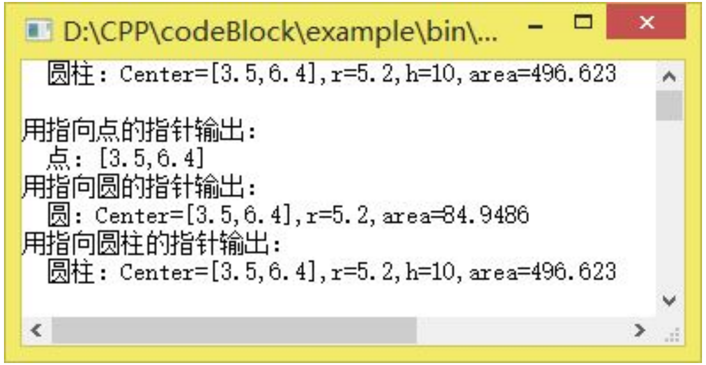
Point
#x : double
#y : double
+operate<<()

Circle
#r : double
+area() : double
+operate<<()

Cylinder
#h : double
+area() : double
+operate<<()



#x: double
#y: double
+operate<<()
#r: double
+operate<<()
+area(): double
#h: double
+operate<<()
+area(): double

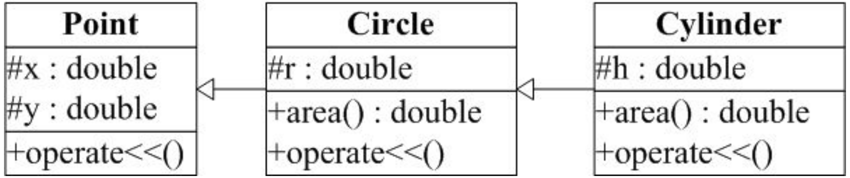


```
 圆柱: Center=[3.5,6.4],r=5.2,h=10,area=496.623

用指向点的指针输出:
 点: [3.5,6.4]
用指向圆的指针输出:
 圆: Center=[3.5,6.4],r=5.2,area=84.9486
用指向圆柱的指针输出:
 圆柱: Center=[3.5,6.4],r=5.2,h=10,area=496.623
```

用输出几何体面积

```
int main( )
{
   Cylinder cy1(3.5, 6.4, 5.2, 10);

   Circle *c= &cy1;
   cout<<"关于面积："<<endl;
   cout<<"用指向圆的指针输出："<<endl;
   cout<<c->area()<<endl;

   Cylinder *cy= &cy1;
   cout<<"用指向圆柱的指针输出："<<endl;
   cout<<cy->area()<<endl<<endl;
   return 0;
}
```

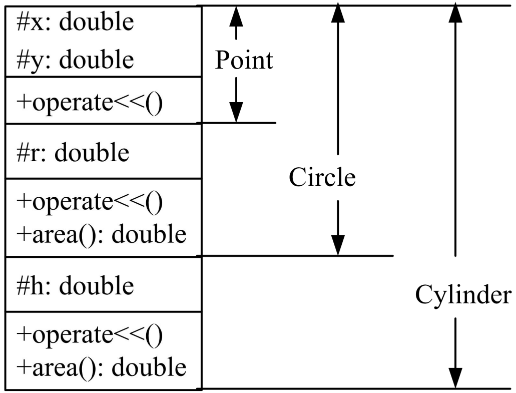
这三例，均是静态多态！



Point
#x : double
#y : double
+operate<<()

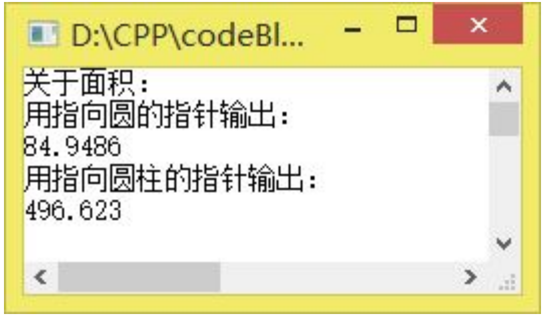
Circle
#r : double
+area() : double
+operate<<()

Cylinder
#h : double
+area() : double
+operate<<()



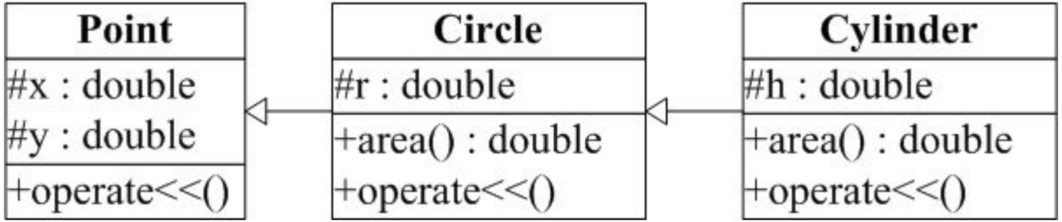
#x: double
#y: double
+operate<<()
#r: double
+operate<<()
+area(): double
#h: double
+operate<<()
+area(): double

Point
Circle
Cylinder



```
关于面积:
用指向圆的指针输出:
84.9486
用指向圆柱的指针输出:
496.623
```

函数重载 VS. 函数覆盖



- operator<<()是函数重载——静态多态

```
ostream &operator<<(ostream &output,Point &p)
ostream &operator<<(ostream &output,Circle &c)
ostream &operator<<(ostream &output,Cylinder& cy)
```

函数参数不同

- area()是函数覆盖——可以做到动态多态

```
double Circle::area( ) const {......}
double Cylinder::area( ) const {......}
```

函数名、参数相同，但在类族的不同层次

一个接口
多种方法

CSDN 学院

THANKS

本课程由 迂者-贺利坚 提供

CSDN网站：www.csdn.net
企业服务：http://ems.csdn.net/
人才服务：http://job.csdn.net/
CTO俱乐部：http://cto.csdn.net/
高校俱乐部：http://student.csdn.net/
程序员杂志：http://programmer.csdn.net/

CODE平台：https://code.csdn.net/
项目外包：http://www.csto.com/
CSDN博客：http://blog.csdn.net/
CSDN论坛：http://bbs.csdn.net/
CSDN下载：http://download.csdn.net/