

**院 系：计 算 机 学 院**

**实验课程：程序设计基础实验课程**

**实验项目：继承与派生**

**指导老师：徐永广**

**开课时间：2022 ～ 2023年度第2学期**

**专 业：计算机科学与技术**

**班 级：22级 计科一 班**

**学 生：李志远**

**学 号：20222131091**

**华南师范大学教务处**

**一、 实验目的与要求**

1．熟练掌握类的继承，能够定义和使用类的继承关系。

2．掌握派生类的声明与定义方法。

3．熟悉公有派生和私有派生的访问特性。

4．理解虚基类在解决二义性问题中的作用。

**二、实验内容**

**1.设计一个圆类 Circle 和一个桌子类 Table，再设计一个圆桌类 Roundtable，它是由前两个类派生出来的，要求输出圆桌的高度、面积和颜色等数据。其中，类 Circle 包含数据成员 Radius 和求面积的成员函数，类 Table 包含数据成员 height 和返回高度的成员函数，类 Roundtable 再添加数据成员 color 和相应的成员函数。**

**Code:**

*#include* <iostream>

using namespace std;

class Shape {

   public:

     virtual void GetArea()

    {

    }

};

class Circle:public Shape{

  public:

   Circle()

  {

     r=2;

  }

   void GetArea()

  {

     cout<<3.14\*r\*r<<endl;

  }

  private:

    double r;

};

class Rectangle:public Shape{

  public:

   Rectangle()

  {

     x=2;

     y=1;

  }

   void GetArea()

  {

     cout<<x\*y<<endl;

  }

  private:

    double x;

    double y;

};

class Square:public Shape{

  public:

   Square()

  {

     x=2;

  }

   void GetArea()

  {

     cout<<x\*x<<endl;

  }

  private:

    double x;

};

   int main()

  {

     Circle c;

     c.GetArea();

     Rectangle r;

     r.GetArea();

     Square s;

     s.GetArea();

  }

**结果：12.56**

**2**

**4**

**2．编写程序声明一个基类 Shape，再派生出 Rectangle 类和 Circle 类，二者都有 GetArea()函数；计算对象的面积。使用程序 Rectangle类创建一个派生类 Square。**

**Code：**

*#include* <iostream>

*#include* <cstring>

using namespace std;

class Circle {

  public:

    Circle(double *r*)

   {

    Radius = *r*;

   }

    void GetArea()

  {

    cout<<Radius<<endl;

  }

  private:

      double Radius;

};

class Table {

  public:

    Table(double *h*)

   {

    height = *h*;

   }

    void Getheight()

  {

    cout<<height<<endl;

  }

    private:

     double height;

};

class Roundtable :public Circle,public Table {

  public:

     Roundtable(double *height*, double *Radius*, string *Color*):Circle(*Radius*),Table(*height*)

  {

    color = *Color*;

  }

     void Getcolor()

    {

       cout<<color<<endl;

    }

  private:

     string color;

};

 int main()

{

   Roundtable r(4,2,"red");

   r.Circle::GetArea();

   r.Table::Getheight();

   r.Getcolor();

}

**结果：**

**2**

**4**

**red**