

第9章 统一接口 不同实现—多态性

1、多态性的基本概念

2、**派生类对象替换基类对象**

3、虚函数的定义

4、抽象类的定义

5、宠物类的设计

6、运算符重载

7、日期类对象判断大小

8、分数类对象运算符重载

派生类对象替换基类对象

□ 一个替换原则：

- 凡是基类对象出现的场合都可以用公有派生类对象取代

□ 三个替换形式：

1. 派生类对象给基类对象赋值
2. 派生类对象可以初始化基类对象的引用
3. 可以令基类对象的指针指向派生类对象，即将派生类对象的地址传递给基类指针

派生类对象替换基类对象举例

- 设计一个手机类mobile，使其能派生出两种发射制式（GSM或者CDMA）的手机类mobilegsm和mobilecdma
 - 派生类对象能够依据各自特征实现显示发射制式的功能，并实现运行时的动态绑定。
 - 通过将基类对象或者指针在运行时指向不同的派生类对象实现动态绑定。
-

一个基类与两个派生类的定义

```
#include <iostream>
using namespace std;
class mobile
{public:
    mobile(){}
    char mynumber[11];           //机主的电话号码
    void showinfo()              //显示制式
    {cout<<"The phone is mobile"<<endl;}
};
class mobilegsm:public mobile
{public:
    mobilegsm(){}
    void showinfo()              //显示制式
    {cout<<"The phone is mobilegsm"<<endl;}
};
class mobilecdma:public mobile
{public:
    mobilecdma(){}
    void showinfo()              //显示制式
    {cout<<"The phone is mobilecdma"<<endl;}
};
```

测试派生类对象替换基类对象

```
int main()
{
    mobile m,*p1;
    mobilegsm gsm;
    mobilecdma cdma;
    m=gsm;
    m.showinfo();
    m=cdma;
    m.showinfo();
    p1=&gsm;
    p1->showinfo();
    p1=&cdma;
    p1->showinfo();
    mobile &p4=gsm;
    p4.showinfo();
    mobile &p5=cdma;
    p5.showinfo();
    return 0;
}
```

//基类对象指针**p1**，基类对象**m**

//用**gsm**类对象给**mobile**类对象赋值

//用**cdma**类对象给**mobile**类对象赋值

//用**gsm**类对象地址给**mobile**类对象赋值

//用**cdma**类对象地址给**mobile**类对象赋值

//以**gsm**类对象地址初始化**mobile**类引用

//以**cdma**类对象地址初始化**mobile**类引用



再举例：派生类与基类定义

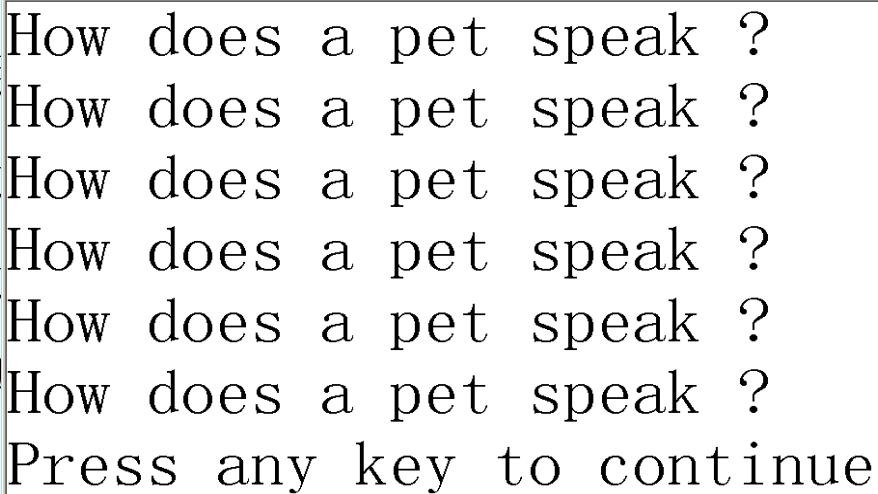
```
#include <iostream>
using namespace std;
class Pet          //基类：宠物类
{
public:
    void Speak() { cout<<"How does a pet speak ?"<<endl; }
};
class Cat : public Pet          //派生类：猫类
{
public:
    void Speak() { cout<<"miao!miao!"<<endl; }
};
class Dog : public Pet          //派生类：狗类
{
public:
    void Speak() { cout<<"wang!wang!"<<endl; }
};
```

派生类对象替换基类对象测试

```
int main()
{
    Pet      *p1, *p2, *p3,obj;    //基类对象指针p1, 基类对象obj
    Dog dog1;
    Cat cat1;
    obj = dog1;
    obj.Speak();
    p1 = &cat1;
    p1->Speak();
    p1 = &dog1;
    p1->Speak();
    p2=new Cat;
    p2->Speak();
    p3=new Dog;
    p3->Speak();
    Pet      &p4 = cat1;
    p4 .Speak();
    return 0;
}
```

//用**Dog**类对象给**Pet**类对象赋值

//用**Cat**类对象地址给基类指针赋值



How does a pet speak ?
How does a pet speak ?
How does a pet speak ?
How does a pet speak ?
How does a pet speak ?
Press any key to continue

两个程序运行共性

□ 手机类或者宠物类

- 尽管派生类对象赋值给了基类对象（指针或引用）

- 但基类对象都不调用派生类同名成员函数

- 只调用基类的同名成员函数

□ **问题**：如何让基类对象（指针或引用）能够调用派生类的函数呢？

- 也就是说如何实现基类对象指针与派生类对象动态绑定？

感谢收看！