

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

虚函数

□ 什么是虚函数？

- 在函数定义的头部加上virtual，该函数就是虚函数
- 事实上，在某基类中声明为virtual并在一个或多个派生类中被重新定义的同名成员函数，称为虚函数。

□ 虚函数的定义：

```
virtual    函数返回类型    函数名 (参数表)
{
    函数体
}
```

□ 虚函数的用途：

- 实现运行时的多态性，即通过指向派生类的基类指针，访问派生类中同名覆盖成员函数
-

手机类改成虚函数定义

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

测试手机类中的虚函数

```
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

```
C:\WINDOWS\system32\cmd.exe
The phone is mobile
The phone is mobile
The phone is mobile
The phone is mobile
The phone is mobile
The phone is mobile
The phone is mobile
请按任意键继续. . .
```

```
D:\2015数据\程序慕课录制\录制实验\test5_gu.exe
The phone is mobile
The phone is mobile
The phone is mobilegsm
The phone is mobilecdma
The phone is mobilegsm
The phone is mobilecdma
```

Process exited after 0.2052 seconds with return value 0
请按任意键继续. . .

微软拼音 半 :

宠物类虚函数实现多态性

```
#include <iostream>
using namespace std;
class Pet
{public: virtual void Speak() { cout<<"How does a pet speak ?"<<endl; } };
class Cat : public Pet
{public: virtual void Speak() { cout<<"miao!miao!"<<endl; } };
class Dog : public Pet
{public: virtual void Speak() { cout<<"wang!wang!"<<endl; } };
int main()
{
    Pet      *p1, *p2, *p3,obj; //基类对
    Dog dog1;
    Cat cat1;
    obj = dog1;                //用D
    obj.Speak();
    p1 = &cat1;                //用C
    p1->Speak();
    p1 = &dog1;                //用D
    p1->Speak();
    p2=new Cat;                //动态
    p2->Speak();
    p3=new Dog;                //动态
    p3->Speak();
    Pet      &p4 = cat1;      //以C
    p4.Speak();
    return 0;
}
```

//基类

//派生类

//派生类

C:\WINDOWS\system32\cmd.exe

How does a pet speak ?

miao!miao!

wang!wang!

miao!miao!

wang!wang!

miao!miao!

请按任意键继续. . .

虚函数的使用限制

- ❑ 应通过指针或引用调用虚函数，而不要以对象名调用虚函数

```
Pet obj;
```

```
Dog dog1;
```

```
obj = dog1;
```

```
obj.Speak();    //执行的是基类Speak()函数
```

```
Pet *p1 = &dog1;
```

```
p1->Speak();
```

- ❑ 在派生类中重定义的基础类虚函数仍为虚函数，同时可以省略virtual关键字
 - ❑ 不能定义虚构造函数，可以定义虚析构函数
-

感谢收看！