

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

多态性含义

👉 **多态**指相同语法结构，代表多种功能或操作

👉 例如： $X+Y$

👉 X 和 Y 都为整数类型：相加操作先以个位对齐，然后对应位相加

👉 X 和 Y 都为浮点类型：相加操作先以小数点对齐，然后对应位相加

👉 当然+两边可以为不同类型也可以相加

👉 能否给+赋予新的加法操作呢？虚数相加

👉 又例如：`datatype max(datatype a, datatype b)`

👉 a 和 b 都为整数，可以**直接**用关系运算符判断比较得出最大值

👉 a 和 b 都为日期对象，判断比较操作较为复杂

👉 能否将`max()`函数重载，求两个新类的对象的大小呢？

多态性含义

👉 无论 $X+Y$ 还是`datatype max (datatype a, datatype b)`

👉 都实现了一种语法格式，隐含了多种操作，这就是多态

👉 多态实现了“一种接口，多种方法”

👉 具体地讲就是将运算符重载，将函数重载，实现不同功能

多态性的两种形式

👉 编译时多态性

👉 编译器对源程序进行编译时就可以确定所调用的是哪一个函数

👉 编译时多态性通过**重载**来实现

👉 函数重载

👉 运算符重载

👉 运行时多态性

👉 在程序运行过程中根据具体情况来确定调用的是哪一个函数

👉 运行时多态通过**虚函数**来实现

编译时多态

```
class 兔子
{
public:
    .....
    void 逃生（老鹰 a）{“兔子蹬鹰”；}
    void 逃生（老狼 b）{“动如脱兔”；}
    .....
};
```

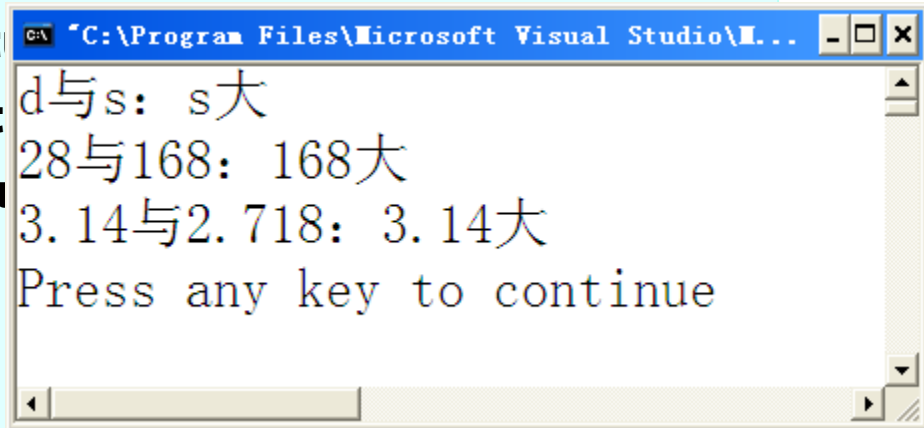
```
void main()
{
```

```
    兔子 rabbit;
    老鹰 eagle;
    老狼 wolf;
    rabbit.逃生(eagle);
    rabbit.逃生(wolf);
}
```

👉 逃生函数是重载函数，相同函数名
👉 在调用函数时，传入的参数类型在
编译时就确定了

编译时多态再举例

```
#include<iostream>
using namespace std;
char    max_( char x, char y ) { ret
int     max_( int x, int y )    { ret
float   max_( float x, float y ) {ret
void main()
{
    float a=3.14,b=2.718;
    cout<<"d与s: "<<max_('d','s')<<"大"<<endl;
    cout<<"28与168: "<<max_(28,168)<<"大"<<endl;
    cout<<"3.14与2.718: "<<max_(a,b)<<"大"<<endl;
}
```



```
C:\Program Files\Microsoft Visual Studio\W...
d与s: s大
28与168: 168大
3.14与2.718: 3.14大
Press any key to continue
```

运行时多态

```
#include<iostream>
using namespace std;
class pet
{public:
    virtual void speak() {cout<<"zzz"<<endl;};
};
class cat : public pet
{
public:
    void speak() { cout<< "miao!miao!"<<endl; }
};
class dog : public pet
{
public:
    void speak() { cout<< "wang!wang!"<<endl; }
};
void main()
{
    pet pet1,*p=&pet1;
    cat cat1;
    dog dog1;
    int x;
    cin>>x;
    if(x==1) p = &cat1;
    if(x==2) p = &dog1;
    p->speak();
}
```

程序中唯一的陌生词

//p为宠物类指针
//定义猫类对象
//定义狗类对象
//根据用户输入将猫或狗对象地址
//石头扔的远近, 只有扔了才知
//例如用户输入1, 则执行猫对象
//例如用户输入2, 则执行狗对象
//究竟运行哪个函数, 只有运行时才知:

```
C:\WINDOWS\system32\cmd.exe
1
miao!miao!
请按任意键继续. . .
```

```
C:\WINDOWS\system32\cmd.exe
2
wang!wang!
请按任意键继续. . .
```

```
C:\WINDOWS\system32\cmd.exe
0
zzz
请按任意键继续. . .
```

感谢收看！