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：

多态性的概念

多态性(polymorphism)的概念

- 多态性是面向对象程序设计的一个重要特征
- 利用多态性可以设计和实现一个易于扩展的系统
- 多态(polymorphism)
  - 多种-poly
  - 形态-morph
- 在C++中多态性的表述
  - 具有不同功能的函数可以用同一个函数名
- 效果
  - 用一个函数名调用不同内容的函数，完成不同的工作

void 打(... ...);

void 打(){打电话;}

void 打(){打屁股;}

多态性形式之一——静态多态性

- 函数重载和运算符重载实现的多态性属于静态多态性
- 在程序编译时系统就能决定调用的是哪个函数，因此，又称编译时的多态性
- 静态多态性是通过函数的重载实现的(运算符重载实质上也是函数重载)

```
int max(int a,int b)
{
   return (a>b)?a:b;
}
int max(int a,int b,int c)
{
   if(b>a) a=b;
   if(c>a) a=c;
   return a;
}
double max(double a,double b,double c)
{
   if(b>a) a=b;
   if(c>a) a=c;
   return a;
}
```

```
cout<<max(2,3);
cout<<max(2.5, 3.4, 2.7);
cout<<max(5,2,3);
```

```
Complex Complex::operator+(Complex &c2)
{
   return Complex(real+c2.real, imag+c2.imag);
}

CTime CTime::operator + (CTime &t)
{
   ...
}
```

```
Complex c1(...), c2(...),c3;
CTime t1(...), t2(...), t3;
int i1(10), i2(20), i3;
c3 = c1 + c2;
t3 = t1 + t2;
i3 = i1 + i2;
```

多态性形式之二——动态多态性

- 动态多态性是在程序运行过程中才动态地确定操作所针对的对象
- 动态多态性又称运行时的多态性

一种死板的机制

```
class Student{
public:
   Student(int, string,float);
   void display( );
protected:
   int num;
   string name;
   float score;
};
```

```
class Graduate:public Student
{
public:
   Graduate(int, string, float, float);
   void display( );
private:
   float pay;
};
```

```
int main(){
   Student stud1(1001,"Li",87.5);
   Graduate grad1(2001,"Wang",98.5,563.5);
   Student *pt=&stud1;
   pt->display( );
   pt=&grad1;
   pt->display( );
   return 0;
}
```

```
void Student::display( ) {
   cout<<"num:"<<num<<"\nname:"<<name<<"\nscore:"<<score<<"\n\n";
}
```

```
void Graduate::display( ) {
   cout<<"num:"<<num<<"\nname:"<<name<<"\nscore:"<<score<<"\npay="<<pay<<endl;
}
```

运行中的动态是这样的！

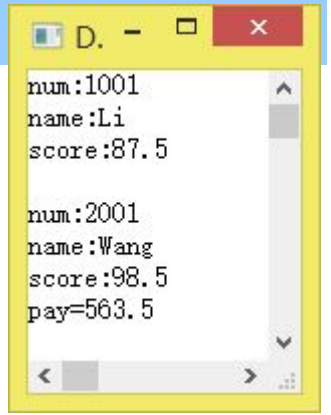
```
class Student{
public:
   Student(int, string,float);
   virtual void display( );
protected:
   int num;
   string name;
   float score;
};
```

```
class Graduate:public Student
{
public:
   Graduate(int, string, float, float);
   void display( );
private:
   float pay;
};
```

```
int main(){
   Student stud1(1001,"Li",87.5);
   Graduate grad1(2001,"Wang",98.5,563.5);
   Student *pt=&stud1;
   pt->display( );
   pt=&grad1;
   pt->display( );
   return 0;
}
```

```
void Student::display( ) {
   cout<<"num:"<<num<<"\nname:"<<name<<"\nscore:"<<score<<"\n\n";
}
```

```
void Graduate::display( ) {
   cout<<"num:"<<num<<"\nname:"<<name<<"\nscore:"<<score<<"\npay="<<pay<<endl;
}
```



CSDN 学院

THANKS

本课程由 迁者-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/