

13 Class Inheritance

如何设计类

在一个系统中，存在多种类型的对象。如果某些对象存在一些共性，如何设计类？

1. 复用。基于已有类派生新类。
2. 层级类结构。
3. 现在的接口调用，适用于未来定义的行为【多态机制】
 - 简洁的，一致的，一般性的表达
 - 易于扩展
 - 类似插件系统的机制

多态

- 类组织成层次结构
- 基类中具有多态机制的方法，申明成虚函数
 - 派生类重写该方法
- 基类指针/引用调用虚函数
 - 指针指向什么类型的对象，就引发该类型的函数调用

```
1 class Base
2 {
3 private:
4     int property;
5 protected:
6     int propertyB;
7 public:
8     virtual void Fun() {}
9     void FunCommon() {}
10 };
11
```

```
12 class DeriveA : public Base
13 {
14 private:
15     double value;
16 public:
17     virtual void Fun(){}
18     void FunCommonA(){}
19 };
20
21 class DeriveB : public Base
22 {
23 private:
24     double value;
25 public:
26     virtual void Fun(){}
27     void FunCommonB(){}
28 };
29
30 void Fun(Base *base)
31 {
32     base->Fun();
33 }
34
35 void Test()
36 {
37     DeriveA a;
38     DeriveB b;
39     Base *base;
40
41     base = &a;
42     Fun(base);
43
44     base = &b;
45     base->Fun();
46 }
47
```

