

纯虚函数

上一篇文章整理了什么是虚函数及其工作机制（请参考14_虚函数详解），接下来就顺便也看看什么是纯虚函数吧。

初识纯虚函数

- 纯虚函数只在基类中声明，但没有定义，因此没有函数体。
- 纯虚函数的声明只需在虚函数形参列表后面添加 `=0` 即可。
- 含有纯虚函数的类都是抽象类。
- 只含有纯虚函数的类称为接口类。

函数声明

纯虚函数的声明很简单，就是在虚函数的形参列表后面添加一个 `=0` 即可，如：

```
1 class Cat {
2     public:
3         virtual void eat()=0;
4 };
```

纯虚函数与抽象类

含有纯虚函数的类称为抽象类（注意！！只要含有就是）。什么是抽象类？它有以下几个特点：

- 抽象类不能实例化对象。
- 抽象类的派生类也可以是抽象类（会继承），也可以通过实现全部纯虚函数使其变成非抽象类，从而可以实例化对象。
- 抽象类的指针可以指向其派生类对象，并调用派生类对象的成员函数。

举个例子，在基类Cat中有两个纯虚函数eat()和sleep()，基类不能直接实例化一个对象来调用这两个函数，但在其派生类CatA和CatB中，可以通过实现这两个函数，当派生类不是抽象类时，便可以实例化对象了。具体请看下面代码示例：

```
1 class Cat{
2     public:
3         //含有纯虚函数，因此Cat为抽象类
4         virtual void eat() = 0;
5         virtual void sleep() = 0;
6 };
7 class CatA : public Cat {
8     public:
9         virtual void eat() { cout << "eat fish." << endl; };    //实现了eat()函数
10        virtual void sleep() = 0;    //仍为纯虚函数，因此CatA也是抽象类
11 };
12 class CatB : public CatA {
13     public:
```

```

14 //两个纯虚函数都被实现，都变成一般的虚函数，因此CatB不是抽象类
15 virtual void eat() { cout << "eat fish." << endl; };
16 virtual void sleap() { cout << "sleap for a long time." << endl; };
17 };
18
19 int main() {
20     Cat a;           //报错，Cat是抽象类，不能实例化对象
21     CatA A;          //报错，CatA也是抽象类，不能实例化对象
22     CatB B;          //正确，CatB不是抽象类
23     CatB *p1 = &B;
24     CatA *p2 = &B;    //抽象类虽然不能实例化对象，但是可以声明其指针或引用
25     p1->eat();         //打印出 eat fish.
26     p1->sleap();       //打印出  sleap for a long time.
27     p2->eat();         //打印出 eat fish.
28     p2->sleap();       //打印出  sleap for a long time.
29     return 0;
30 }

```

补充一个小疑惑：

刚了解抽象类概念的时候，有个疑惑：既然抽象类不能直接实例化对象，为何不直接使用带有虚函数的一般类就好，而是要定义一个不能实例化对象的抽象类？原因我们可以这样理解，比如我们定义一个类的时候，我们希望它应该具有某种功能（如Teachers类有一个teaching功能，但具体教什么科目还不知道），因此我们可以在定义了它的派生类之后（如MathTeacher），再具体实现teaching这个函数。

纯虚函数与接口类

- 只含有纯虚函数的类称为接口类。（注意！！是只含有）
- 接口类没有任何数据成员，也没有构造函数和析构函数。
- 接口类的指针也可以指向其派生类对象