# **笔记**

2017/5/23

# 继承和派生

1、“继承”就是在一个已存在的类的基础上建立一个新的类。已存在的类称为“基类(base class)”或“父类(father class)”，新建的类称为“派生类(derived class)”或“子类(son class )”。

2、一个新类从已有的类那里获得其已有特性，这种现象称为类的继承。通过继承，一个新建子类从已有的父类那里获得父类的特性。从另一角度说，从已有的类(父类)产生一个新的子类，称为类的派生。

3、一个派生类不仅可以从一个基类派生，也可以从多个基类派生。也就是说，一个派生类可以有一个或者多个基类。一个派生类有两个或多个基类的称为多重继承(multiple inheritance)。

# 虚函数

1、作用：它虚就虚在所谓“推迟联编”或者“动态联编”上，一个类函数的调用并不是在编译时刻被确定的，而是在运行时刻被确定的。由于编写代码的时候并不能确定被调用的是基类的函数还是哪个派生类的函数，所以被成为“虚”函数。

2、虚函数只能借助于指针或者引用来达到多态的效果。必须实现虚函数。

3、声明：：virtual ReturnType FunctionName(Parameter)；

4、对于虚函数来说，父类和子类都有各自的版本。由多态方式调用的时候动态绑定。

# 纯虚函数

1、定义：纯虚函数是在基类中声明的虚函数，它在基类中没有定义，但要求任何派生类都要定义自己的实现方法。在基类中实现纯虚函数的方法是在函数原型后加“=0”

　virtual void funtion1()=0

编译器要求在派生类中必须予以重写以实现多态性。同时含有纯虚拟函数的类称为抽象类，它不能生成对象。

2、特征：它们必须在继承类中重新声明函数（不要后面的＝0，否则该派生类也不能实例化），而且它们在抽象类中往往没有定义。

3、目的：使派生类仅仅只是继承函数的接口。纯虚函数一定没有定义，只用来规范派生类的行为，即接口。包含纯虚函数的类是抽象类，抽象类不能定义实例，但可以声明指向实现该抽象类的具体类的指针或引用。

4、虚函数与纯虚函数：

（1）定义一个函数为虚函数，不代表函数是不被实现的函数。

（2）定义为虚函数是为了允许用基类的指针来调用子类的这个函数。

（3）定义一个函数为纯虚函数，才代表函数没有被实现。

（4）定义纯虚函数是为了实现一个接口，起到一个规范的作用，规范继承这个类的程序员必须实现这个函数。

# 抽象类

1. 定义：声明了纯虚函数的类是一个抽象类。所以，用户不能创建类的实例，只能创建它的派生类的实例。
2. 作用：将有关的操作作为结果接口组织在一个继承层次结构中，由它来为派生类提供一个公共的根，派生类将具体实现在其基类中作为接口的操作。所以派生类实际上刻画了一组子类的操作接口的通用语义，这些语义也传给子类，子类可以具体实现这些语义，也可以再将这些语义传给自己的子类。
3. 使用：抽象类只能作为基类来使用，其纯虚函数的实现由派生类给出。如果派生类中没有重新定义纯虚函数，而只是继承基类的纯虚函数，则这个派生类仍然还是一个抽象类。如果派生类中给出了基类纯虚函数的实现，则该派生类就不再是抽象类了，它是一个可以建立对象的具体的类。
4. 抽象类是不能定义对象的。

# 虚析构函数

1、在有动态分配堆上内存的时候，析构函数必须是虚函数，但没有必要是纯虚的。这样做是为了当用一个基类的指针删除一个派生类的对象时，派生类的析构函数会被调用。

2、并不是要把所有类的析构函数都写成虚函数。因为当类里面有虚函数的时候，编译器会给类添加一个虚函数表，里面来存放虚函数指针，这样就会增加类的存储空间。所以，只有当一个类被用来作为基类的时候，才把析构函数写成虚函数。

3、构造函数不能是虚函数。而且，在构造函数中调用虚函数，实际执行的是父类的对应函数，因为自己还没有构造好, 多态是被disable的。 析构函数可以是虚函数，而且，在一个复杂类结构中，这往往是必须的。析构函数可以是纯虚的，但纯虚析构函数必须有定义体，因为析构函数的调用是在子类中隐含的。

4、派生类的override虚函数定义必须和父类完全一致。除了一个特例，如果父类中返回值是一个指针或引用，子类override时可以返回这个指针（或引用）的派生。（没懂）