

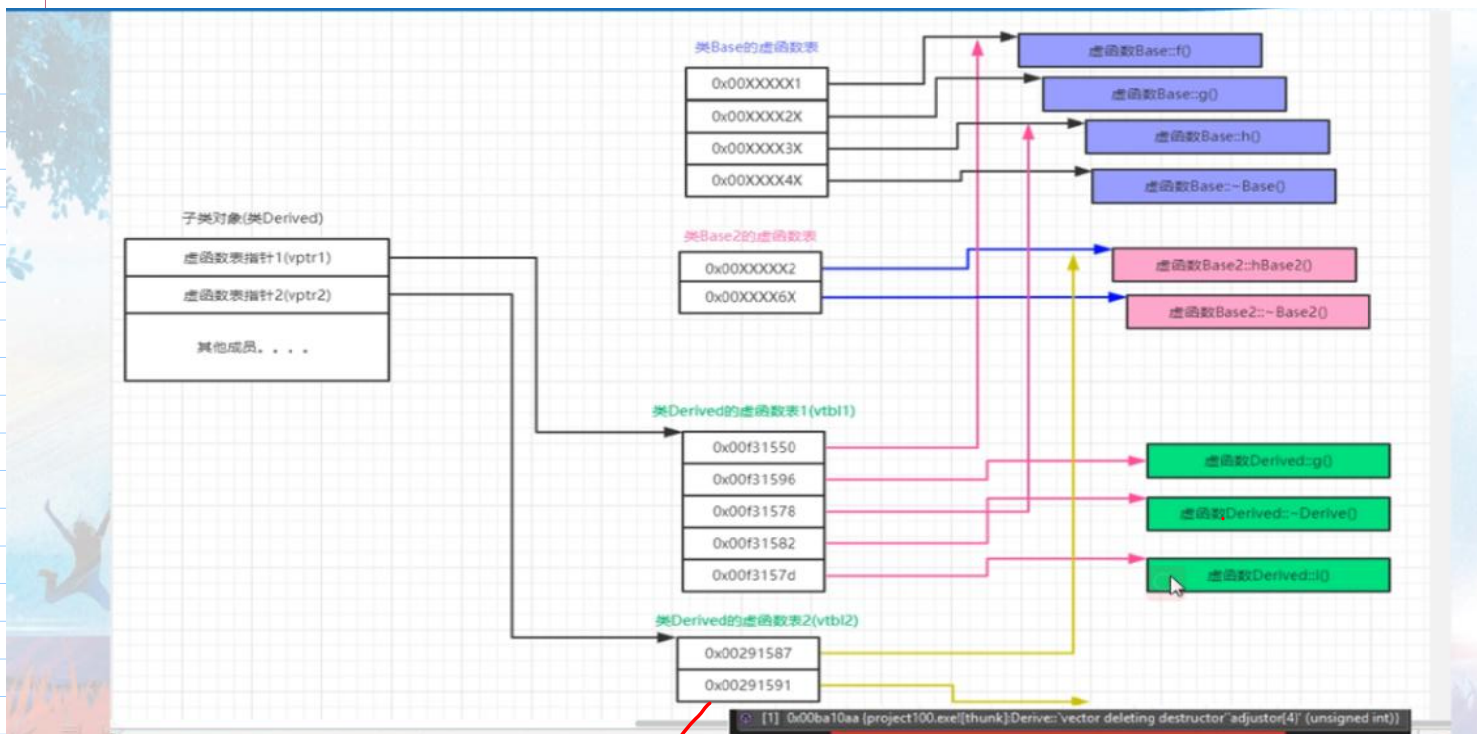
```
//Derive 类的第二个虚函数表中发现了thunk字样:
//一般这玩意用在多重继承中(从第二个虚函数表开始可能就会有);用于this指针调整。这玩意其实是一段汇编代码,这段代码干两个事情
// (1)调整this指针
// (2)调用Derive析构函数
```

先Base_this通过vp_ptr在vtable中找到thunk.将this指针偏移至子类this调用delete该this,并且在子类的析构函数中插入代码调用每个父类析构函数

```
//二: 如何成功删除用第二基类指针new出来的继承类对象
//a) 我们要删除的实际上是整个Derive()对象
//b) 要能够保证Derive()对象的析构函数被正常调用
//c) 编译器会调用Base2的析构函数,还是调用Derive的析构函数呢?
//d) 执行delete pb2时,系统的动作会是?
//d. 1)如果Base2里没有析构函数,编译器会直接删除以pb2开头的这段内存,一定报异常,因为这段内存压根就不是new起始的内存;
//d. 2)如果Base2里有一个析构函数,但整个析构函数是个普通析构函数(非虚析构函数),那么当delete pb2,
//这个析构函数就会被系统调用,但是delete的仍旧是pb2开头这段内存,所以一定报异常。因为这段内存压根就不是new起始的内存;
//析构函数如果不是虚函数,编译器会实施静态绑定,静态绑定意味着你delete Base2指针时,删除的内存开始地址就是pb2的当前位置;
```

//d.4)Derive里就算没有虚析构函数,因为Base2里有虚析构函数,编译器也会为此给Derive生成虚析构函数,为了调用Base2()和Base()生成虚析构函数,为了调用Base2()和Base()虚析构函数;

//e)老师要求大家,凡是涉及到继承的,所有类都要求大家写虚析构函数;



thunk