

问题

虚函数可以声明为inline吗？

内联函数 (inline)

内联函数的作用是**提高函数执行的效率**，在程序中的每个调用点将函数体展开，而不是按照通常的函数调用机制取调用，从而减少调用函数花费的额外开销。内联函数有一些特点：

- 定义在class内的成员函数默认是inline函数（虚函数除外）
- 通常只有函数非常短小的时候（如10行代码内）才适合定义成inline函数，否则会导致程序变慢
- 头文件中不仅要包含inline函数的声明，还要包含其定义，方便编译器查找。
- （缺点）inline函数会增加执行文件的大小。

虚函数可以声明为inline吗？

可以，但是只在编译器知道调用的对象是哪个类型的时候才可以。

虚函数一般不能声明为inline的，因为inline函数是在编译期将函数内容替换到函数调用处的，是静态编译的。而当基类指针或引用调用虚函数时，不能声明为inline，因为虚函数是在运行时动态调用的，编译器并不知道它绑定的是哪个对象。

借助下面的代码来理解上面的两种情况：

```
1  class Base {
2  public:
3      inline virtual void who() { cout << "I am Base\n"; }
4  };
5  class Derived :public Base {
6  public:
7      //从语法上讲，这里可以写成inline，只是当基类指针调用派生类时，不能内联，编译器会自动
      忽略掉inline
8      inline void who() { cout << "I am Derived\n"; }
9  };
10 int main() {
11     Base b;
12     b.who(); //这里的who()是通过基类对象直接调用的，在编译期间就确定了，因此它可以是
      内联的
13
14     Base *p = new Derived();
15     p->who(); //通过基类的指针调用，在运行时才能确定，所以不能内联
16     return 0;
17 }
```

参考资料

[C++内联函数能否是虚函数？](#)

[C++中虚函数不能是inline函数的原因](#)

[C++虚函数\(10\)-虚函数能否为inline?](#)