以下都是使用using的作用

```
class B : public A
{
public:
    //B(int i, int j, int k) :A(i, j, k){}
    using A::A;  //继承A的构造函数。    using:就让某个名字在当前作用域内可见。
                 //遇到这条代码的时候，会把基类的每个构造函数，都生成一个与之对应的派生类构造函数。
                  //B(构造函数形参列表...):A(照抄的构造函数形参列表){} ：函数为空
                      //B(int i,int j,int k):A(i,j,k){};
    //如果基类A的构造函数有默认参数的话，那么编译器遇到这种using A::A的时候，就会帮咱们在派生类B中构造出多个构造函数来
    //a)第一个构造函数是带有所有参数的构造函数；
    //b)其余的构造函数，每个分别省略掉 一个默认参数。
    //B(int i,int j,int k):A(i,j,k){}
    //B(int i,in j):A(i,j){};
    //如果类B，只含有using A::A从A类继承来的构造函数的话，那么编译器是会给它合成默认的构造函数的。
};
```

```
//一：继承的构造函数
//一个类只继承其直接基类（父类）的构造函数。默认、拷贝、移动构造函数是不能被继承的。                    的定义覆盖掉(相当于你只继承了一部分基类中的构造函数)。
//  B ad(3, 4, 5);
//如果基类含多个构造函数，则多数情况下，派生类会继承所有这些构造函数，但如下例外情况：
//1)如果你在派生类中定义的构造函数与基类构造函数有相同的参数列表，那么从基类中继承来的构造函数会被你在派生类中的定
//2)默认、拷贝、移动构造函数 不会被继承
B ad(3, 4);
B ef();
```