

## 3.2 继承\_物联网 / 嵌入式工程师 - 慕课网

“ 慕课网慕课教程 3.2 继承涵盖海量编程基础技术教程，以图文图表的形式，把晦涩难懂的编程专业用语，以通俗易懂的方式呈现给用户。

继承允许我们依据另一个类来定义一个类，这两个类之间存在明显的 "is a" 关系。这使得创建和维护一个应用程序变得更容易。这样做，也达到了重用代码功能和提高执行时间的效果。

```
class 派生类名:继承方式 基类名1, 继承方式 基类名2, ...  
  
{  
  
    派生类类体  
  
};
```

- 新创建的类称为 \*\*\*\* 派生类 (子类)，被继承的类称为 \*\*\*\* 基类 (父类)
- 继承方式有:public , protected,private

```
class Luban:public Hero,public Shooter{  
  
public:  
  
    Luban(const string &name,const int distance,const int legLen);  
  
    ~Luban(void);  
  
    void show(void) const;  
  
private:  
  
    int legLen;  
  
};
```

此时 Luban 这个类拥有了 Hero 类和 Shooter 这个类的所有属性。

先按照继承时的顺序挨个调用基类的构造函数，然后调用派生类的构造函数。

```
Luban::Luban(const string &name,const int distance):  
  
    Hero(name),  
  
    Shooter(distance)  
  
{  
  
    this->legLen = legLen;  
  
}
```

基类构造函数的调用，必须在派生类的初始化列表中指定。如果没有指定，则 \*\*\*\* 默认调用基类无参数的构造函数。

析构函数调用顺序和构造函数调用顺序相反

实现一个后裔类，观察其构造函数和析构函数调用顺序

---

全文完

本文由 简悦 SimpRead 优化，用以提升阅读体验

使用了 全新的简悦词法分析引擎 beta，点击查看详细说明

