



链滴

1.8单根继承结构

作者: [someone756](#)

原文链接: <https://ld246.com/article/1441520657429>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

 在OOP中，自从C++面世以来就已变得非常瞩目的一个问题就是，所有的类最终都继承自单一的基类。在Java中（事实上包括除C++以外的所有OOP语言），但答案是es，这个最终基类的名字就是Object。事实证明，单根继承带来了许多好处。

 在单根继承结构中所有的对象都具有一个共用接口，所以他们归根到都是相同的基本类型。另一种（C++所提供的）结构无法确保所有所有的对象都属于同一个基本类型从向后兼容的角度看，这么做能够更好地适应C模型，而且受限较少，但是当要进行完全的面向对象程序设计时，必须要构建自己的继承体系，才能够使得它可以提供其他OOP语言内置的便利。并且在获的任何新类库中，总会用到一些不兼容的接口，需要花力气（可能要通过多重继承）来使新街口融入的设计之中。这么做来换取C++额外的灵活性是否值得呢？如果在C上面投资巨大，这么做就很有价。如果是刚刚从头开始，那么像Java这样的选择通常会有更高的生产率。

 单根继承结构保证所有对象都具备某些功能。因此你知道，在你的系中你可以在每个对象上执行某些基本操作。所有对象都可以很容易地在堆上创建，而参数传递也得到极大地简化。

 单根继承结构使垃圾回收器的实现变得容易的多，而垃圾回收器正是Ja a相对C++的重要改进之一。由于所有对象都保证具有其类型信息，因此不会因无法确定对象的类型陷入僵局。这对于系统级操作（如异常处理）显得尤其重要，并给变成带来了更大的灵活性。

总结：

- 单根继承结构中所有对象具有一个共同接口，都是相同的基本类型
- 在每个对象上执行某些基本操作
- 所有对象很容易在堆上创建，简化参数传递
- 使垃圾回收器的实现变得容易

-----END-----