

Design Principles

Easy Construction 让对象的构建方法变得简洁.

`lnk = Link(1, Link(2, Link(3, None)))`

`lnk = Link([1, 2, 3])` # 我们显然更愿意采取这种构建方法.

Set Boundaries 设置合理的抽象边界

(理论上) 没有对实例属性设置下划线意味着我们可以随意访问它.

作为程序员, 如果想告诉别人此处存在抽象边界, 就可以通过加下划线实现.

但我们仍无法完全避免这种事发生, 例如刻意修改 `--str--`, `--iter--` 等方法的时候

加双下划线的属性不能从类外的方法(例如继承的子类)中调用和修改.

强行调用会导致系统告诉你 "no attribute"

仍有强行调用的方法, 小心你的同事们!

Check your assumptions 检查你的假设

大部分时候我们创建的类都是为了对现实世界建模.

自然, 我们要保证其中的数据符合实际.

比如人的名字一定是字符串, 而非列表、布尔值等等乱七八糟的东西.

有些假设不一定适用于所有情况, 比如假定人一定有姓、中间名、名.

我们需要做一些拓展应对这种例外.

Design For Reuse 让程序有一定的泛用性

项目就像积木城堡, 我们希望代码作为一种易用的积木应用于各种城堡中