

## 面向对象编程

# Object-Oriented Programming

### 封装

Encapsulation 把数据与相关的行为绑定在一起，成为一种类或一个对象。

### 组合

Composition 对象可以包含其他对象

### 继承

Inheritance 对象的固有行为来自它的基类。

### 多态

## Polyorphism

- 1) 指一个函数可以在不同类的对象上运行，一般是继承自同一基类的对象
- 2) 即前面提到过的泛型函数 (Generics, duck typing)

经典的例子就是在任何 iterables 上运行的函数。

只要这些对象具有 `--iter--` 方法 (实际上，如果它有，那么它就是 iterable)

在 Python 中函数并不限定传入变量的类型，而 Java 和 C/C++ 中则有。

这实际上为在 Python 中实现 Generics 提供了很大便利。

- 3) 类型强制转换 (type coercion)，将参数转换为必要的类型。

例如 `int()`, `str()` 等耳熟能详的函数。

`int("2.1")` 是非法的，但 `int(2.1)` 和 `int("2")` 都是合法的。

- 4) 类型分派 (type dispatching)，函数检查参数类型以选择正确的行为。

在 Scheme project 中的 `scheme-eval` 函数就是如此。

常使用的判断方法为 `isinstance` 函数。