

# [读书笔记] SICP 计算机程序的构造和解释

接下来会看MIT Lisp版本的SICP，以下是找到的一些资源：

1. [课程主页](#)：是MIT 6.001的课程主页，包含一些和该课程相关的资源
2. [习题答案](#)
3. 作业：包含5个[项目](#)，以及16个[编程作业](#)
4. [课程安排](#)：列出了该课程的上课进度，可通过[这里](#)获得课程和书本的对应关系
5. [翻译视频](#)：这应该是MIT 2005年这门课程的视频
6. 电子书：[官网电子书](#)、[HTML版本](#)
7. 一些大佬整理的资源：
8.
  1. [DeathKing/Learning-SICP](#)
  2. [Clement Li: 学习SICP（《计算机程序的构造和解释》）的一些准备工作](#)
  3. [Scheme入门中文教程](#)

已整理内容：

- **第一章：构造过程抽象**
- - [\[1B\]程序设计的基本元素](#)
  - [\[2B\]过程及其产生的计算过程](#)
  - [\[3B\]用高阶函数做抽象](#)

第一章主要介绍过程抽象，**主要想法**就是通过高阶过程来抽象出不同的“概念”，通过传递过程作为参数来得到针对特定问题的过程，或者将过程传入高阶过程返回封装好的过程，由此来对过程进行统一的修改。需要对现有方法抽象出抽象的概念，并通过高阶过程来产生具体过程。

- **第二章：构造数据抽象**
- - [\[4B\]数据抽象引导](#)
  - [\[5B\]层次性数据和闭包性质](#)
  - [\[6B\]符号数据](#)
  - [\[7B\]抽象数据的多重表示](#)
  - [\[8B\]带有通用型操作的系统](#)

第二章主要介绍数据抽象，主要介绍以下内容：

- **抽象屏障**：需要通过抽象屏障将数据对象的使用与具体实现分隔开来，只通过选择函数和构造函数来沟通。同样也需要通过抽象屏障将数据对象的不同表示方法分隔开来，可以构建通用选择函数来同时处理不同表示方式的数据对象。
- **分层设计的形式**：在最底层设计出基本元素以及对基本元素进行变换的高阶过程，在第二层中基于基本元素设计出不同的组合方式获得新的基本元素，在最后一层中抽象出描述组合模式的高阶过程。
- **处理不同数据对象的过程**：首先需要将过程划分为：枚举、过滤/映射和累积，然后确定好约定的界面，在枚举中将数据对象转化为约定界面，使得后续过程在所有数据对象中都能通用。**优点**：选择一种合适的约定界面来连接不同的标准部件，使得我们能对这些标准部件进行模块化设计而无需考虑具体的数据结构（因为连接部件的界面是约定好的），由此来控制程序复杂度，使得我们能通过级联标准部件的方式来构建复杂的系统。
- **数据不同表示方法**：可以采用数据导向的方式构建通用选择函数，使得数据具有不同的表示方法：

- **封装操作**：可以让数据增加标志，然后通用选择函数通过数据的标志判断是用什么表示方法，再采用特定的选择函数进行处理（构造函数要增加标志，通用选择函数只传递数据内容给特定的选择函数）
  - **封装表示方式**：可以将数据表示为过程，当传入选择函数名作为参数时，直接返回对应的数据。
  - **数据导向**：可以用一个二维表格来根据操作和表示方法保存特定的选择函数到特定的位置，然后根据传入的参数标志和需要的操作提取出对应的过程（特定选择函数直接写成处理数据内容的形式，因为通过数据标志确定要选取的操作后，会只传递数据内容给特定选择函数，但是构造函数要记得增加标志）。
- **通用型操作**：同样可以构建一个数据类型和操作名的二维表格，由此实现在不同数据类型下的通用操作。为了避免每次新增数据类型就要添加很多新的操作，而数据类型其实是有层次的，可以通过为每个数据类型都定义一个 `raise` 操作，当表格中找不到传入数据类型个对应的操作时，就能尝试调用该数据的 `raise` 操作提升到更高一层的，再尝试寻找想要的操作。而为每个数据类型定义一个 `drop` 操作，可以在适当的时机将数据类型下降一定层次，可以得到交优的结果。
- **抽象数据的设计方法**：
  - 首先假设存在满足程序员对数据对象预期效果的构造函数和选择函数，基于构造函数和选择函数来实现上层过程，由此能无需考虑数据对象的具体实现。
  - 随后可以通过对构造函数和选择函数的具体实现来影响程序性能，而无需考虑基于此的上层过程。如果数据对象具有不同表示方法，则需要构建通用选择函数。
  - 可以将数据对象的操作添加到通用型操作中，需要对产生数据的过程添加上类型标志。

- **第三章：模块化、对象和状态**

- [\[9B\]赋值和局部状态](#)
  - [\[10B\]求值的环境模型](#)

提高模块性、降低复杂性、抽象屏障。