14. 向量和结构体

1. 简介

本章中，我将讲解向量和结构体。向量是一组通过整数索引的数据。同一个向量可以储存不同类型的数据，这有点与C语言的数组不同。与表相比，向量更加紧凑且存取时间更短。但从另外一方面来说，向量是通过副作用来操作的，这样会造成负担。Scheme中的结构体与C语言中的结构体类似。但Scheme中的结构体比C语言中的更容易使用，这是因为Scheme为结构体自动创建了读取函数和写入函数，这受益于Lisp/Scheme程序设计语言中的宏。

1. 向量
   1. 字面

向量通过闭合的‘#(’和‘)’表示，例如#(1 2 3)。作为字面值时，它们应该被引用，例如：

'#(1 2 3) ; 整数向量

'#(a 0 #\a) ; 由符号、整数和字符构成的向量

* 1. 定义在向量上的函数

下面的函数都是R5RS规定的函数：

(vector? obj)

如果obj是一个向量则返回#t。

(make-vector k)

(make-vector k fill)

放回一个有k个元素的向量。如果指定了第二个参数fill，那么所有的元素都会被初始化为fill。

(vector obj …)

返回由参数列表构成的向量。

(vector-length vector)

返回向量vector的长度。

(vector-ref vector k)

返回向量vector的索引为k的元素。（译注：和C语言类似，向量从0开始索引。）

(vector-set! vector k obj)

将向量vector的索引为k的元素修改为obj。

(vector->list vector)

将vector转换为表。

(list->vector list)

将list转换为向量。

(vector-fill! vector fill)

将向量vector的所有元素设置为fill。

例：一个对向量中元素求和的函数。

01: (define (vector-add v1 v2)

02: (let ((lenv1 (vector-length v1))

03: (lenv2 (vector-length v2)))

04: (if (= lenv1 lenv2)

05: (let ((v (make-vector lenv1)))

06: (let loop ((i 0))

07: (if (= i lenv1)

08: v

09: (begin

10: (vector-set! v i (+ (vector-ref v1 i) (vector-ref v2 i)))

11: (loop (1+ i))))))

12: (error "different dimensions."))))

练习1

编写一个用于计算两向量内积的函数。