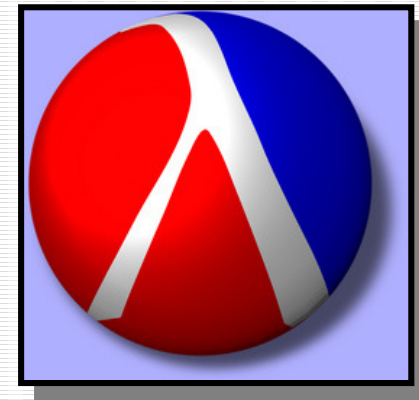


Fundamentos de Programación

Tipos de Datos: Símbolos y Estructuras

Profesor: Daniel Wilches Maradei
Diapositivas Originales: Jesús A. Aranda

Universidad del Valle



Qué es un símbolo ?

- Un símbolo es una pieza atómica de información representada por una cadena de caracteres
- Los símbolos pueden ser usados como nombre de variables, o para denotar Strings
- Al usar un símbolo le estamos diciendo a Scheme que acepte la palabra tal como es, sin tratar de interpretarlo más que como un cadena de caracteres
- La diferencia entre símbolos y Strings es que los primeros, aunque se forman de múltiples letras, son atómicos

Creando símbolos

- Para crear un símbolo debe anteponerse una comilla sencilla a la cadena de caracteres:

`'scheme`

- Los símbolos no pueden tener espacios
- Para comparar símbolos se usa el predicado `symbol=?`

Ejemplo symbol=?

- (symbol=? 'uno 'dos)
false
- (symbol=? 'tres 'tres)
true

Ejemplo de uso de los símbolos

```
(cond
  ((= dia 1) 'domingo)
  ((= dia 2) 'lunes)
  ((= dia 3) 'martes)
  ....
  ((= dia 7) 'sabado)
  (else 'DiaInvalido)
)
```

Strings != Símbolos

- Los Strings, a diferencia de los símbolos, no son atómicos, es decir, Scheme entiende que son un conjunto de caracteres
- Podemos preguntar por el tamaño de un String, más no de un símbolo. Igualmente, solo en los Strings se puede preguntar por el carácter que está en la posición `N`.
- Los Strings se declaran encerrando entre `"` una cadena de caracteres, mientras los símbolos se declaran precediendo con ``` una cadena

Imágenes

- En Scheme también se le puede asociar un símbolo a una imagen, de este modo podemos mostrarla cuando deseemos:

(define celular )

- Esto lo logramos mediante la opción Editar+Insertar Imagen
- En el momento en el que hagamos esto, el código fuente deja de ser texto plano !!

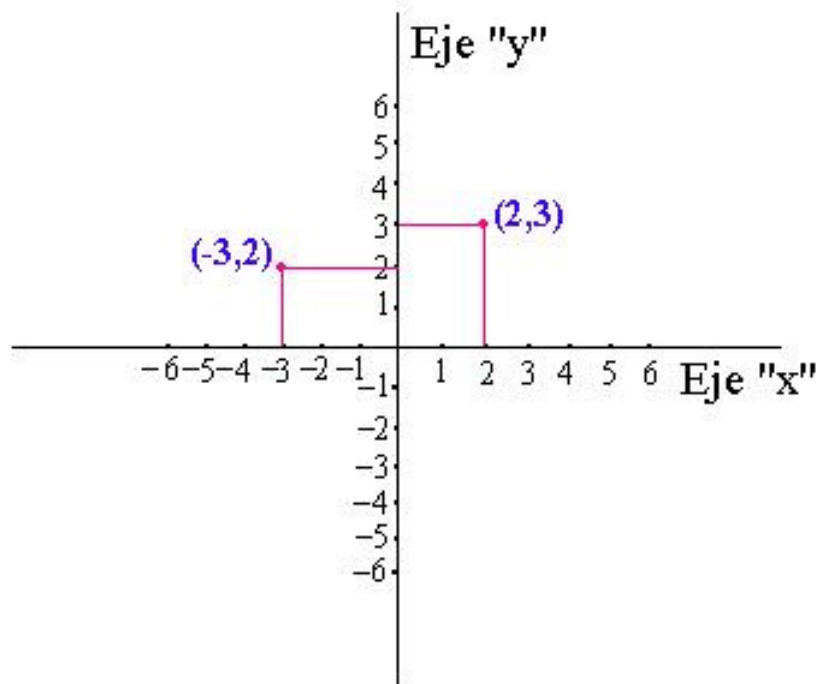
Estructuras

- Son tipos de datos más complejos, no atómicos, que se conforman de varias partes.
- A cada una de sus partes se le da un nombre
- Sus partes pueden ser accedidas de manera independiente
- Para qué pueden servir las estructuras ?

Uso de las estructuras

1. Definir tipos de datos complejos:
 - Cómo representar un punto ?
 - Cómo representar una persona ?
2. Cuando necesitemos devolver varios valores desde una función
 - Si necesitamos devolver en una función un par de números por ejemplo

Ejemplo



- Qué deberíamos hacer para representar un **PUNTO 2D** en Scheme ?
- Noten que se compone de 2 partes: **coordenada X** y **coordenada Y**

Representación

Symbol 'correcto'

correcto

String "Hola Mundo"

H	o	l	a		M	u	n	d	o
---	---	---	---	--	---	---	---	---	---

Struct Punto2D

x	y
23	45

Definición de la estructura Punto2D

(define-struct Punto2D (X Y))

Struct Punto2D

x	y

- **define-struct** es la manera de definir una estructura llamada **Punto2D** que contiene 2 componentes: **X** y **Y**

Creación de una variable de tipo estructura

(make-Punto2D 3 4)

Struct Punto2D	
x	y
3	4

- `make-<nombreestructura>` es una función que crea una estructura del tipo especificado.
- Esta función se define automáticamente cuando usamos `define-struct`

Acceso a los componentes de una estructura

(Punto2D-X variable)

(Punto2D-Y variable)

- Cuando usamos el `define-struct`, además de crearse la función `make-<estructura>`, se crean las funciones accesoras a sus campos. Estas funciones se llaman:
`nombreestructura-nombrecampo`

Ejemplo de estructuras

```
(define-struct Punto2D (X Y))  
(define puntoInicial (make-Punto2D 1 1))  
(define puntoFinal (make-Punto2D 5 12))
```

"La suma de las coord.X de ambos puntos es:"

```
(+ (Punto2D-X puntoInicial) (Punto2D-X puntoFinal))
```

Ejercicio

- Cree una función que calcule la distancia entre 2 puntos.
- Recuerde que la fórmula para calcular la distancia entre 2 puntos es:

$$\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

Estructura posn

- La estructura `Punto2D` que acabamos de crear ya está predefinida en Scheme pero con otro nombre: `posn`
- `posn` es el equivalente a un punto en 2 dimensiones, sus componentes son `x` y `y`

Creación de estructura Punto3D

- Crear una estructura **Punto3D** que permita representar un punto en sus sistema de 3 dimensiones

Creación de estructura EstudianteUV

- Crear una estructura **EstudianteUV** que permita representar a un estudiante de la **Universidad del Valle**. Entre los componentes pueden estar nombre, código, año de nacimiento, créditos, etc.

struct?

- Hay un predicado que es declarado también cuando usamos una estructura con `define-struct`:

`nombreestructura?`

- Este predicado nos permite determinar si una variable contiene un valor del tipo de estructura deseado:

`Punto2D?`

`Punto3D?`

`EstudianteUV?`

`Posn?`

Guías de diseño

- Cuando creemos estructuras es importante documentar bien qué tipo de datos va a contener, cuál es la función de cada uno de sus componentes, y cuál es la función final de la estructura.

- ; Esta estructura representa un punto en un sistema
 - ; de 2 dimensiones.

- (define-struct Punto2D (X Y))

- ; Las componentes del punto X y Y deben ser valores
 - ; numéricos, y corresponden a su distancia en los eje
 - ; X y Y del punto de origen (0,0)