

Práctica 4 – Estructuras

Redictado Programación I

1 Coordenadas: la estructura `posn`

Recordemos de la clase de teoría que podíamos representar coordenadas en el plano a través de una estructura con dos campos. En *racket* viene definida la estructura `posn`, que podemos usar a tal fin. Aquí el diseño de datos de la misma:

```
(define-struct posn [x y])
; una posición es (make-posn Number Number)
; interpretación: un elemento en posn representa una
; posición en coordenadas cartesianas.
```

Recordemos también que esta definición crea automáticamente cuatro funciones para operar con la estructura `posn`:

- `make-posn`, llamado **constructor**. Es el que nos permite crear objetos de la clase `posn`.
- `posn-x` y `posn-y`, llamados **selectores**. Nos permiten acceder a cada uno de los campos de la estructura.
- `posn?`, llamado el **predicado** de la estructura. Dado un objeto cualquiera, devuelve `#true` si su argumento es un objeto de tipo `posn`.

Ejercicio 1. Escriba las firmas de cada una de estas cuatro funciones.

Ejercicio 2. Escriba cinco instancias de la estructura `posn`, asignando una constante a cada caso, dónde existan por lo menos un ejemplo para cada cuadrante del plano.

2 Evaluando Estructuras

A través de dos leyes fundamentales, podemos calcular con estructuras. Estas leyes son simples y relacionan cómo se comportan los selectores respecto del constructor:

<pre>(posn-x (make-posn a b)) == definición de posn-x (ley 1) a</pre>	<pre>(posn-y (make-posn a b)) == definición de posn-y (ley 2) b</pre>
---	---

A partir de estas dos leyes, podemos calcular con estructuras de tipo `posn`. Por ejemplo, si tenemos:

```
(define p (make-posn 3 4))
(define q (make-posn -2 0.5))
```

Podemos calcular la expresión `(posn-y (make-posn (posn-y q) (posn-x p)))`:

```

(posn-y (make-posn (posn-y q) (posn-x p)))

== definición de q

(posn-y (make-posn (posn-y (make-posn -2 0.5)) (posn-x p)))

== (ley 2)

(posn-y (make-posn 0.5 (posn-x p)))

== definición de p

(posn-y (make-posn 0.5 (posn-x (make-posn 3 4))))

== (ley 1)

(posn-y (make-posn 0.5 3))

== (ley 2)

3

```

Ejercicio 3. Calcule el valor de las siguientes expresiones:

1. `(posn-x p)`
2. `(- (posn-y p) (posn-y q))`
3. `(posn-y (make-posn (posn-x p) (posn-x q)))`

Recuerde que al decir **calcule** le estamos pidiendo resolver *con papel y lápiz*, y no utilizando *racket*.

Revise sus procedimientos (en especial el último) con el evaluador paso a paso de *racket*

Ejercicio 4. Defina una función `dist-origen` que calcule la distancia al origen de un punto, representado por una estructura `posn`. Calcule el valor de las siguientes expresiones:

1. `(dist-origen (make-posn (/ 6 2) 4))`
2. `(+ (dist-origen (make-posn 12 5)) 4)`

3 Retornando una Estructura

Ejercicio 5. Diseñe una función `simétrico`, que dada una posición, nos devuelva su simétrica respecto del origen.

Ejercicio 6. Diseñe una función `desplazamiento-x`, que dada una posición, y un valor de traslado `delta`, nos devuelva la posición trasladada en su eje `x` `delta` unidades.

Ejercicio 7. Diseñe una función `desplazamiento-y`, que dada una posición, y un valor de traslado `gamma`, nos devuelva la posición trasladada en su eje `y` `gamma` unidades.

Ejercicio 8. Diseñe una función `desplazamiento-xy`, que dada una posición, y un valor de traslado `omega`, nos devuelva la posición trasladada en su eje `x` y en el eje `y` `omega` unidades.

Ejercicio 9. Diseñe una función `distancia`, que dado dos puntos en el plano calcule la distancia entre ellos. Extienda la definición para que, en caso que alguno de los argumentos no sea de tipo `posn`, muestre el mensaje `"Tipos incorrectos para la función."`

4 Estructura Fracción

Ejercicio 10. En este ejercicio representaremos un `fraccion` con una estructura con los siguientes campos:

- 1er campo: numerador
- 2do campo: denominador

Teniendo en cuenta esto se pide:

1. Diseñe una estructura `fraccion` que contenga los campos descriptos más arriba.
2. Escriba cinco instancias de la estructura `fraccion` asignando una constante a cada caso.
3. Diseñe una función que valide una estructura de tipo `fraccion`. Recuerde que el denominador nunca puede ser cero.
4. Diseñe una función que construya una estructura de fracción válida.

Ejercicio 11. Diseñe una función `suma-fracciones` que tome como entrada dos elementos del tipo `fraccion` y retorne la `fraccion` que resulta de sumar las entradas. Recuerde que las fracciones pueden tener igual o distinto denominador. Actuar en consecuencia en el diseño.

Ejercicio 12. Diseñe una función `simplifica-fraccion` que tome como entrada una `fraccion` y simplifica la misma. Las funciones `gcd` y `min` pueden serle de utilidad.

5 Estructura Alumno

Ejercicio 13. En este ejercicio representaremos un `alumno` con una estructura con los siguientes campos:

- 1er campo: Nombre del alumno.
- 2do campo: Promedio de sus calificaciones (un valor entre 0 y 10).
- 3er campo: Porcentaje de asistencia a clases (un valor entre 0 y 100).

Teniendo en cuenta esto se pide:

1. Diseñe una estructura `alumno` que contenga los campos descriptos más arriba.
2. Escriba cinco instancias de la estructura `alumno` asignando una constante a cada caso.

Ejercicio 14. Diseñe una función `asistencia-perfecta?` que tome como entrada un valor de tipo `alumno` determine si su porcentaje de asistencia se encuentra entre el 90% y el 100%.

Ejercicio 15. Diseñe una función `baja-asistencia?` que tome como entrada un valor de tipo `alumno` determine si su porcentaje de asistencia es inferior al 60%.

Ejercicio 16. Diseñe una función `condicion` que tome como entrada un valor de tipo `alumno` y devuelva un string indicando su condición. Las condiciones posibles son: `"Libre"`, `"Regular"` y `"Promovido"`. Para calcular la condición del `alumno` deben tenerse en cuenta las siguientes reglas:

- Si el alumno tiene un porcentaje de inasistencia mayor al 40% queda automáticamente libre, sin importar el promedio de sus calificaciones.
- Si el alumno tiene una asistencia mayor o igual al 60%:
 - y tiene una nota inferior a 6, también se considera libre.
 - y tiene una nota mayor o igual a 6 y menor estricta que 8, se considera regular
 - y una nota mayor o igual a 8, se considera promovido.

En caso en que la función `condicion` reciba como entrada un dato que no corresponda a una estructura `alumno` deberá responder con un mensaje de error (como por ejemplo: `"Tipo de dato inválido"`).

6 Estructura Persona

Ejercicio 17. En este ejercicio representaremos una persona mediante una estructura con 5 campos:

- 1er campo: el nombre y apellido
- 2do campo: el valor numérico de su peso
- 3er campo: un string que representa la unidad en la cual está dado el peso (valores posibles: "G" o "Kg")
- 4to campo: el valor numérico de la estatura
- 5to campo: un string que representa la unidad en la cual está dada la estatura (valores posibles: "Mts" o "Cms")

Teniendo en cuenta esto se pide:

1. Diseñe una estructura persona que le permita representar a cualquier persona con su peso y estatura.
2. Genere por lo menos 5 instancias de la estructura persona, asignándole constantes en cada caso.

Ejercicio 18. Utilizando la estructura persona recientemente definida, diseñe una función `IMC` que tome como entrada un valor de tipo persona y calcule su índice de masa corporal. En caso que no reciba como entrada una estructura de tipo persona deberá mostrar el siguiente mensaje de error: `"Tipo de dato inválido"`.

Ayuda 1: Recuerde que cada estructura que define viene acompañada de un predicado que determina si un objeto es o no una estructura de ese tipo.

Ayuda 2: Por si no lo sabe, el índice de masa corporal (IMC) de una persona se calcula según la siguiente fórmula:

$$IMC = \frac{\text{Peso (kg)}}{\text{Estatura}^2 \text{ (Mts.)}}$$

Ayuda 3: Le puede servir conocer las siguientes equivalencias:

$$1 \text{ Kg} = 1000 \text{ G}$$

$$1 \text{ Mts} = 100 \text{ Cms}$$

Ejercicio 19. Diseñe una función peso-salud que tome como entrada un valor de tipo persona, y utilizando la función IMC, le indique en palabras a la persona cómo se encuentra según su peso. Para ello utilizaremos la siguiente clasificación:

- Si su IMC es inferior a 18.5, está dentro de los valores correspondientes a "bajo peso".
- Si su IMC es entre 18.5 y 24.9, está dentro de los valores "normales" o de peso saludable.
- Si su IMC es entre 25.0 y 29.9, está dentro de los valores correspondientes a "sobrepeso".
- Si su IMC es 30.0 o superior, está dentro de los valores de "obesidad".

En caso que no reciba como entrada una estructura de tipo persona deberá indicar que no es un tipo de dato válido, con el siguiente mensaje de error: "Tipo de dato inválido".

7 Estructura Auto

Ejercicio 20. En este ejercicio representaremos un auto con una estructura con los siguientes campos:

- 1er campo: Modelo.
- 2do campo: Año.
- 3to campo: Tipo de combustible (diesel o nafta).
- 4to campo: Rendimiento óptimo, expresado en kilómetros por litro.

Teniendo en cuenta esto se pide:

1. Diseñe una estructura auto que contenga los campos descriptos más arriba.
2. Diseñe una función costo-viaje que tome como entrada un valor de tipo auto y un número de kilómetros a recorrer y calcule el costo del viaje. Para esto debe tener en cuenta:
 - Cantidad de combustible: el número de litros necesarios para recorrer m kilómetros con un auto nuevo está determinado por su rendimiento óptimo. Sin embargo, con el correr de los años, el rendimiento disminuye. Se estima que si el auto tiene:
 - Entre 1 y 5 años, el rendimiento disminuye 2%
 - Entre 6 y 10 años, el rendimiento disminuye 6%
 - Entre 10 y 15 años, el rendimiento disminuye 10%
 - Más de 15 años, el rendimiento disminuye 15%

Es decir, si un auto 0km rinde 13km/litro, después de un año rendirá 12.74 km/litro, y después de 12 años 11,7 km/litro.

- Peajes: por cada 100 kilómetros recorridos, se debe pagar un peaje de \$50.
- Precio combustible: el precio actual del litro de nafta es \$19 y el litro de diesel \$17.

Ejemplo: Un gol naftero 2013 de rendimiento óptimo 13km/litro, debido a sus 4 años de antigüedad tendrá un rendimiento de 12,74 km/litros. Por lo tanto, recorrer 450 kms tendrá un costo de: $(450 / 12,74) * 19\$ + 200\$$

8 Estructura Casa

Ejercicio 21. Representaremos una casa con una estructura con los siguientes campos:

- 1er campo: *propietario*
- 2do campo: *dirección*
- 3er campo: *superficie en metros cuadrados*
- 4to campo: *zona*

Teniendo en cuenta esto se pide:

1. Diseñe una estructura casa que contenga los campos descriptos más arriba.
2. Diseñe una función venta que tome como entrada un valor de tipo casa y devuelva un mensaje sobre los datos de la venta de dicha propiedad. Dicho mensaje deberá dar los datos sobre el propietario que vende, en qué dirección se encuentra y el monto de dinero que recibe el propietario por dicha venta luego de realizados los descuentos correspondientes por el sellado de la escritura. Para realizar los cálculos se deben tener en cuenta las siguientes cuestiones:
 - Existen 4 zonas:
 - Zona A: el metro cuadrado en esta zona tiene un valor de 20000 pesos
 - Zona B: el metro cuadrado en esta zona tiene un valor de 15000 pesos
 - Zona C: el metro cuadrado en esta zona tiene un valor de 10000 pesos
 - Zona D: el metro cuadrado en esta zona tiene un valor de 5000 pesos
 - El propietario debe pagar un sellado que se descontará del precio de la venta. Si el precio de la venta supera el millón de pesos el sellado tiene un costo del 5 % del valor y si no lo supera se debe abonar el 3% del valor de la propiedad.

Ejemplo:

- Supongamos que el señor José Romero vende una propiedad que se encuentra en la calle Rueda 3456. Dicha propiedad posee una superficie de 120 metros cuadrados y, por la dirección en que se encuentra, pertenece a la zona C. El monto de dinero que José recibe por dicha venta es de 1140000 pesos. Si aplicáramos la función venta en este caso deseáramos que esta función nos devolviera el siguiente mensaje: *"El señor José Romero recibirá 1140000 pesos por la venta de su propiedad ubicada en la calle Rueda 3456."*

En caso que la función venta reciba como entrada un dato que no sea de tipo casa deberá mostrar el mensaje *"Tipo de dato incorrecto"*.

En caso que la función `venta` reciba como entrada un dato de tipo `casa` con una zona distinta a A, B, C o D deberá mostrar el mensaje "No se puede calcular el precio de venta por no disponer de los valores del metro cuadrado para la zona solicitada".

Ayuda: Recuerde que cada estructura que define viene acompañada de un predicado que determina si un objeto es o no una estructura de ese tipo.