

Lenguajes de Programación

Práctica 2

Semestre 2022-1

Facultad de Ciencias, UNAM

Profesora Karla Ramírez Pulido
Ayud. Lab Silvia Díaz Gómez
Ayud. Lab Fernanda Montserrat Romo Olea

Fecha de inicio: 14 de Octubre de 2021
Fecha de entrega: 27 de Octubre de 2021

Descripción

La práctica definir algunos tipos de datos y algunas funciones dentro del archivo `practica2.rkt`. No está permitido usar primitivas de Racket que resuelvan directamente los ejercicios.

Ejercicios

1. (2 pts.) Definir un tipo de dato abstracto **Figura** que sea utilizado para trabajar con figuras geométricas.

El tipo de dato abstracto debe contener:

- Un constructor (`triangulo a b c`) donde `a`, `b` y `c` son números reales y representan los lados del triángulo.
- Un constructor (`cuadrado a`) donde `a` es un número real y representa el lado del cuadrado.
- Un constructor (`rectangulo a b`) donde `a` y `b` son números reales y representan la altura y base del rectángulo.
- Un constructor (`rombo a D d`) donde `a`, `D` y `d` son números reales y representan el lado, diagonal mayor y diagonal menor del rombo respectivamente.
- Un constructor (`paralelogramo a b h`) donde `a`, `b` y `h` son números reales y representan los lados y altura del paralelogramo respectivamente.
- Un constructor (`circulo D`) donde `D` es un número real y representa el diámetro del círculo.
- Un constructor (`elipse a b`) donde `a` y `b` son números reales y representan el semieje mayor y el semieje menor de la elipse respectivamente.

Una vez definido el tipo de dato, se deben de definir las siguientes funciones:

- (a) (1.5 pts.) Una función (`perimetro f`) que dada una figura regrese el perímetro de ésta.

```
(perimetro (triangulo 2 3 4)) => 9  
(perimetro (circulo 3))      => 9.42477796076938
```

```
;; perimetro: Figura → number  
(define (perimetro f) ...)
```

(b) (1.5 pts.) Una función (**area f**) que dada una figura calcule el área de ésta.

```
(area (rectangulo 2 3)) => 6
(area (rombo 3 4 5))    => 10
```

```
;; area: Figura → number
(define (area f) ...)
```

2. Para un tren, existen distintos tipos de vagones:

- Vagón simple, del cual se conoce la capacidad total de pasajeros y la capacidad utilizada.
- Vagón locomotora, del cual se conoce el peso total de carbón utilizado.
- Vagón restaurante, del cual se conoce el número de mesas y meseros disponible.
- Vagón dormitorio, del cual se conoce el número de camas disponibles.
- Vagón de carga, del cual se conoce el peso total utilizado por el equipaje.

Un tren consiste de una locomotora en un extremo, seguida por al menos un vagón (que no sea de tipo locomotora) y finalmente una locomotora en el otro extremo del tren.

Definir los tipos de datos **Vagon** (2 pts.) y **Tren**(1 pto.) que cumplan con la definición anterior. Estas definiciones, deben permitir sólo configuraciones válidas de trenes. Una vez definidos los tipos de datos, se deben definir las siguientes funciones.

(a) (0.5 pts.) Una función (**pasajeros t**) que obtiene el número total de pasajeros.

```
;; pasajeros: Tren → number
(define (pasajeros t) ...)
```

(b) (0.5 pts.) Una función (**peso-carbon t**) que obtiene el peso total del carbón.

```
;; peso-carbon: Tren → number
(define (peso-carbon t) ...)
```

(c) (0.5 pts.) Una función (**sin-cama t**) que determina, de acuerdo al número de pasajeros y el número de camas disponibles, cuántos pasajeros se quedarán sin cama durante la noche, considerando que en cada cama cabe una persona.

```
;; sin-cama: Tren → number
(define (sin-cama t) ...)
```

(d) (0.5 pts.) Una función (**peso-aproximado t**) que determina, de acuerdo al peso del carbón, del equipaje y el de los pasajeros, un aproximado del peso total del tren. Considerar como peso promedio de un pasajero 73kg.

```
;; peso-aproximado: Tren → number
(define (peso-aproximado t) ...)
```

Requerimientos

Deberás entregar tu práctica a través de la plataforma **Google classroom** antes de las 23:59 del día de entrega indicado, tal y como lo indican los lineamientos de entrega. Deberás adjuntar tu código en un archivo llamado **practica2.rkt**. Revisa bien mayúsculas, minúsculas y espacios para el nombre del archivo que se pide.

El orden en el que aparezcan las funciones en el archivo solicitado, debe ser el orden especificado en este PDF, de lo contrario podrán penalizarse algunos puntos. Puedes utilizar funciones auxiliares, se recomienda definirla después de la función que la va a utilizar. No dudes en aclarar tus dudas en la sesión de laboratorio y vía correo electrónico.