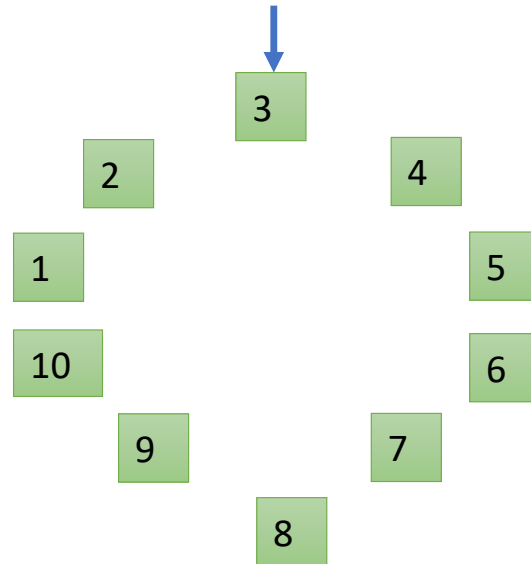


TAD Roulette

Especificación informal

- Una ruleta es una estructura circular de elementos donde hay uno destacado señalado con un puntero.



En este caso, la ruleta contiene los elementos del 1 al 10 en sentido horario y el elemento destacado es el 3.

Especificación informal de una Roulette

Las operaciones disponibles para una ruleta son:

- Saber si está vacía o no.
- Devolver el elemento destacado.
- Girar n posiciones la ruleta en sentido horario si n es positivo o antihorario si n es negativo.
- Borrar el elemento destacado. EL destacado pasa a ser el siguiente en sentido horario.
- Añadir un nuevo elemento delante del destacado que pasa a ser el destacado.
- Aplicar una función a cada elemento de la ruleta quedando los resultados como elementos de una nueva ruleta.
- Pasar de una lista a una ruleta.
- Pasar de una ruleta a una lista.
- Crea una ruleta vacía.

Especificación formal de una Roulette

```
empty      :: Roulette a
isEmpty    :: Roulette a -> Bool
sign       :: Roulette a -> a
turn       :: Integer -> Roulette a -> Roulette a
delete     :: Roulette a -> Roulette a
insert     :: a -> Roulette a -> Roulette a
mapRoulette :: (a -> b) -> Roulette a -> Roulette b
listToRoulette :: [a] -> Roulette a
rouletteToList :: Roulette a -> [a]
```

Implementación del TAD Roulette

- Vamos a implementar la Roulette con una cola (Queue) y un entero que guarda el tamaño de la ruleta:
 - Los elementos en sentido horario de la ruleta son los sucesivos elementos de la cola
 - En la cabeza de la cola estará el elemento destacado.
- INVARIANTES:
 - Siempre hay un elemento destacado (salvo en la ruleta vacía)
 - Siempre mantendremos el elemento destacado en la cabeza de la cola.

`data Roulette a = R (Q.Queue a) Integer deriving Eq`

Ejemplo

`data Roulette a = R (Q.Queue a) Integer deriving Eq`

- Ejemplo.
 - La ruleta anterior se representará por

`R LinearQueue(3,4,5,6,7,8,9,10,1,2) 10`

Su show será `QueueRoulette:10(3,4,5,6,7,8,9,10,1,2)`

- NOTA: Todas las operaciones con colas estarán cualificadas con Q:
`Q.Queue`, `Q.enqueue`, `Q.first`, etc.